

Solving Multi-Objective Constrained Trajectory Optimization Problem by an Extended Evolutionary Algorithm

Runqi Chai, *Member, IEEE*, Al Savvaris, Antonios Tsourdos, *Member, IEEE*, Yuanqing Xia, *Senior Member, IEEE*, and Senchun Chai

Abstract—Highly constrained trajectory optimization problems are usually difficult to solve. Due to some real-world requirements, a typical trajectory optimization model may need to be formulated containing several objectives. Because of the discontinuity or nonlinearity in the vehicle dynamics and mission objectives, it is challenging to generate a compromised trajectory that can satisfy constraints and optimize objectives. To address the multi-objective trajectory planning problem, this study applies a specific multiple-shooting discretization technique with the newest NSGA-III optimization algorithm and constructs a new evolutionary optimal control solver. In addition, three constraint handling algorithms are incorporated in this evolutionary optimal control framework. The performance of using different constraint handling strategies is detailed and analyzed. The proposed approach is compared with other well-developed multi-objective techniques. Experimental studies demonstrate that the present method can outperform other evolutionary-based solvers investigated in this paper with respect to convergence ability and distribution of the Pareto-optimal solutions. Therefore, the present evolutionary optimal control solver is more attractive and can offer an alternative for optimizing multi-objective continuous-time trajectory optimization problems.

Index Terms—Trajectory optimization, Multi-objective optimal control, multiple-shooting, NSGA-III optimization, Pareto-optimal.

I. INTRODUCTION

OVER the past couple of decades, trajectory optimization problems have attracted a large amount of attentions due to their increasing significance in industry and military fields [1], [2]. Generally, this type of problem aims to find the optimal state and control sequences so as to optimize the predefined performance index. Relative works on this topic can be found in various scientific and engineering applications such as agent/robot trajectory planning [3], [4], autonomous vehicle optimal path design [5], and spacecraft optimal control systems [6]–[8]. More precisely, in [2] the author proposed a time-optimal trajectory generation strategy for a multi-body car model. Pritesh et al. [1] solved a fixed-wing unmanned aerial vehicle trajectory planning problem by embedding human expert cognition. In addition, the trajectory generation

problem for a class of wheeled inverted pendulum vehicles was studied and reported in [9]. Besides, an optimal spacecraft Sun-Earth orbital transfer trajectory was designed by applying a hybrid invariant manifold method [6]. Similarly, the low computational cost orbital transfer path was generated by Peng and Wang in [7], wherein an adaptive surrogate optimization technique was constructed. In their follow-up research [8], an emergency transfer trajectory design mission was considered and solved via a fast surrogate-based optimization method. Although many optimization strategies have been designed for trajectory planning problems, it is still challenging to generate the optimal or near optimal state and control trajectories under a highly constrained environment.

Since it is difficult and unrealistic to obtain an analytical solution from a nonlinear continuous-time system, discretization methods are usually employed to solve trajectory planning problems with constraints. For discretization techniques, one effective technique which has been applied in practical problems is the single shooting method. In a shooting scheme, only the control variable is parameterized. Then numerical integration techniques (e.g. the Euler method) are applied to satisfy the differential constraints [10]. Another well-known methodology is the collocation method (e.g. the pseudospectral techniques). Three well-developed collocation methods are the direct collocation method [11], the Chebyshev pseudospectral method [12] and the Gauss pseudospectral method [13]. Unlike shooting techniques, collocation methods parameterize both the control and state variables. Subsequently, the continuous-time optimal control problem is discretized to a nonlinear programming problem (NLP) which can be addressed via effective nonlinear optimization techniques.

In the recent ten years, evolutionary-based optimization techniques have become popular and have been widely implemented to address optimal trajectory design problems. Compared with traditional gradient-based algorithms, evolutionary algorithms do not require initial guess values since the initial population is chosen randomly, and it is more likely than gradient approaches to find the global optimal solution [14]. Furthermore, there is no derivative information required in a heuristic approach, which means it will not suffer from the difficulty of constructing Jacobian and Hessian matrix. Contributions made to apply these approaches in optimal control can be found in literatures [15]–[20]. For example, in [15], a constrained space plane reentry trajectory design problem was addressed by using a genetic algorithm (GA), whereas

R. Chai, A. Savvaris and A. Tsourdos are with the School of Aerospace, Transport and Manufacturing, Cranfield University, UK, e-mail: (r.chai@cranfield.ac.uk), (a.savvaris@cranfield.ac.uk), and (a.tsourdos@cranfield.ac.uk).

Y. Xia and S. Chai are with the school of Automation, Beijing Institute of Technology, Beijing, China, e-mail: (xia_yuanqing@bit.edu.cn), (chais-c97@163.com).

Kamesh et al. [16] combined a hybrid genetic algorithm with collocation method in order to analyze an Earth-Mars orbit transfer problem. In [17], the authors generated the optimal trajectory for robotic manipulators based on particle swarm optimization (PSO). Conway et al. [18] combined evolutionary algorithms with direct collocation to create a bi-level structural optimal control solver. An enhanced differential evolution method was constructed in our previous work to approximate the optimal flight trajectory [19], where a simplex-based direct search mechanism was embedded in the algorithm framework. Besides, a detailed analysis and comparative study between different optimization techniques for solving an aeroassisted vehicle orbital hopping trajectory design issue can be referred to [20].

The problem address in this paper is a multi-objective optimal trajectory design for the spacecraft in the reentry phrase. For general engineering multi-objective problems, the evolutionary multi-objective optimization (EMO) methodology has been illustrated as a promising tool to analyze the relationships between objectives and calculate the pareto-front [21]. New techniques based on EMO have been widely studied during the past decades [22]–[26]. For example, Roy et al. [22] developed an optimal path control strategy for solving general multi-objective optimization problems. Ji et al. [23] designed a modified NSGA-II algorithm to address a multi-objective continuous berth allocation problem. In [24], the author proposed a decomposition-based EMO technique, along with a novel diversity factor, for handling many-objective problems. Moreover, an enhanced many-objective PSO method was proposed in [25], wherein a two-stage strategy was designed so as to better balance the convergence and diversity of the approximated pareto solutions. Furthermore, in [26] a constraint consensus-based mechanism, together with a new mutation operator, was studied for solving multi-objective benchmark problems. However, most of these EMO techniques cannot be directly applied to solve the multi-objective spacecraft trajectory design problem. This is because most of these works only targeted at unconstrained problems or problems with algebraic equality and inequality constraints. If an EMO is employed to calculate the multi-objective optimal spacecraft trajectory, a proper treatment of the continuous-time differential constraints is also required. To do this, in this paper, a novel NSGA-III-based optimal control solver is designed and applied to solve a multi-objective spacecraft trajectory optimization problem. So far to the best of the author's knowledge, there is no adequate work that has been reported to investigate the multi-objective reentry trajectory design problem, and the NSGA-III-based algorithm has not been applied to this kind of problem before. Hence, the present study is an attempt to address these concerns.

The main contributions of the presented work include the following aspects:

- 1) The spacecraft trajectory optimization problem established in [27] is reformulated and extended to a new multi-objective reentry optimal control model. Various path constraints including the control rate and obstacle avoidance are considered in the optimization process.
- 2) The original NSGA-III algorithm is extended by embed-

ding a discretization scheme such that this integrated computational framework can have the capability of handling system dynamics and producing optimal trajectories for the multi-objective spacecraft optimal control problem.

- 3) Different constraint handling algorithms are embedded in the proposed framework to deal with various mission constraints. The performance of these strategies is analyzed in detail. The constraint-handling strategy, with the best found performance, is suggested to solve the investigated application problem.
- 4) The multi-objective reentry optimization model, together with the extended NSGA-III algorithm, is tested for different mission scenarios. Experimental results are verified and compared with other EMOs based on the performance indicators established in the multi-objective domain.

This paper is constructed as follows. In Section II, a newly-established trajectory optimization model is extended to a multi-objective optimization formulation. Following that, Section III presents the discretization technique and the main framework of the extended NSGA-III for solving the discrete-time nonlinear programming model. Section IV constructs several constraint handling strategies. Case studies and comparative simulation results are presented in Section V. In Section VI, the influences of some important parameters with respect to the optimal result are analyzed. Finally, the conclusions are drawn in Section VII.

II. PRELIMINARIES AND PROBLEM FORMULATION

A typical multi-objective optimal control problem with the consideration of state boundary and path constraints can be formulated as follows.

It is desired to obtain a control function $u(t) \in \mathbb{R}^u$ such that the multiple objective functions/performance indices can be minimized:

$$\min(J(x(t), u(t))) = [J_1(x(t), u(t)), J_2(x(t), u(t)), \dots, J_M(x(t), u(t))] \quad (1)$$

in which M denotes the number of mission objectives; $x(t) \in \mathbb{R}^x$ represents the state variable which is required to satisfy the dynamic constraints:

$$\dot{x}(t) = f(x(t), u(t), t) \quad (2)$$

the state boundary constraints:

$$\zeta(x(0), x(t_f), t_0, t_f) = 0 \quad (3)$$

and the state and control path constraints:

$$P(x(t), u(t), t) \leq 0 \quad (4)$$

A. Multi-objective spacecraft trajectory optimization problem

The overall objective of the spacecraft trajectory optimization problem studied in this work is to generate a constrained optimal flight trajectory, for a given vehicle, to strike the predetermined terminal condition in maximum or minimum performance indices. The performance index defines

optimality, which is mission-dependent and typically designed by the mission planners. In most cases, equality and inequality constraints are adhered to the equations of motion such as variable box constraints, no-fly zones and mission specified path restrictions. These limitations are usually modeled into equations and employed to search the optimal control sequence. Therefore, the first step is to formulate the continuous-time spacecraft optimal control optimization problem used throughout this research.

B. Dynamics

The dynamics of the spacecraft, together with the initial boundary conditions, are given by the following system of equations [19], [28]:

$$\begin{aligned} \dot{r} &= V \sin \gamma \\ \dot{\theta} &= \frac{V \cos \gamma \sin \psi}{r \cos \phi} \\ \dot{\phi} &= \frac{V \cos \gamma \cos \psi}{r} \\ \dot{V} &= \frac{-D}{m} - g \sin \gamma \\ \dot{\gamma} &= \frac{L \cos \sigma}{mV} + \left(\frac{V^2 - gr}{rV} \right) \cos \gamma \\ \dot{\psi} &= \frac{L \sin \sigma}{mV \cos \gamma} + \frac{V}{r} \cos \gamma \sin \psi \tan \phi \\ (r(0), \phi(0), \theta(0), V(0), \gamma(0), \psi(0)) &= (r_0, \phi_0, \theta_0, V_0, \gamma_0, \psi_0) \end{aligned} \quad (5)$$

where r stands for the distance from the center of the Earth. θ and ϕ stand for the longitude and latitude, respectively. V represents the speed, while γ denotes the flight path angle. ψ represents the azimuth angle, whereas m is the mass of the spacecraft. The angle of attack α and bank angle σ are the two control variables. g is the gravity acceleration. For simplicity reasons, the dynamic equations described in Eq.(5) is abbreviated as $\dot{x} = f(x, u)$, $x(0) = x_0$, where $x \in \mathbb{R}^6$ and $u \in \mathbb{R}^2$. $L(\alpha, \rho)$ and $D(\alpha, \rho)$ are the lift and drag forces and they are functions of α and the density of the atmosphere ρ . The simplification of the atmosphere model is given by $\rho = \rho_0 \exp((r - r_0)/h_s)$, in which ρ_0 is the density of atmosphere at sea-level, whereas h_s denotes the density scale height.

C. Motivation and mission objective

It is worth mentioning that most existing spacecraft trajectory design works are targeted at single-objective problems. However, in reality, for optimal trajectory design, a lot of missions may have multiple mission requirements. These objectives must be frequently considered during the path planning phase, and this brings the development of multiple objective trajectory optimization techniques.

For the reentry trajectory optimization problem investigated in the present study, four competitive objective functions are considered so as to take more real-world requirements into account (e.g. $J = [J_1, J_2, J_3, J_4]$). The first objective function to be optimized is the cross range value, i.e. the terminal longitude $\phi(t_f)$. A larger cross range value is usually desired and this can be an important indicator to measure the entry flying ability of the space vehicles. In addition, minimizing the total aerodynamic heating and the heat flux are also chosen as the objectives since the vehicle structure integrity is largely influenced by these two criteria. Moreover,

since it is not desirable to have many oscillations during the flight, a smoothness indicator is also chosen as the objectives. Therefore, the mission objectives selected for the analysis are established as follow:

$$\begin{cases} \max J_1 = \phi(t_f) \\ \min J_2 = \int_{t_0}^{t_f} \dot{Q}(t) dt \\ \min J_3 = \int_{t_0}^{t_f} \dot{\gamma}(t) dt \\ \min J_4 = \max \dot{Q}(t) \end{cases} \quad (6)$$

D. Constraints

For safety reasons, the spacecraft entry manoeuvre should satisfy different types of constraints. During the entire flight mission, all the variables should vary in their allowable range known as box constraints and it can be written as:

$$x_{min} \leq x \leq x_{max}, \quad u_{min} \leq u \leq u_{max} \quad (7)$$

Early studies on spacecraft trajectory design problems only used box constraints to limit the controls [28]. However, in practice, certain requirements for the rate of control variables should be considered such that the control sequence and its derivative cannot vary significantly. These requirements are achieved using the lag equations shown as follows:

$$\begin{cases} \dot{\alpha} = k_\alpha(\alpha_c - \alpha) \\ \alpha_c^{min} \leq \alpha_c \leq \alpha_c^{max} \end{cases} \quad \begin{cases} \dot{\sigma} = k_\sigma(\sigma_c - \sigma) \\ \sigma_c^{min} \leq \sigma_c \leq \sigma_c^{max} \end{cases} \quad (8)$$

Eq.(8) is adhered to Eq.(5) in the evolutionary phase (will be discussed in the next section), thereby increasing the state-space order by two. For this case, α_c and σ_c are now treated as the control variable.

In order to protect the vehicle structure integrity, three path constraints, namely aerodynamic heating, dynamic pressure and load factor, are also taken into account.

$$\begin{aligned} \dot{Q} &= K_Q \rho^{0.5} V^{3.07} (c_0 + c_1 \alpha + c_2 \alpha^2 + c_3 \alpha^3) < \dot{Q}^{max} \\ P_d &= \frac{1}{2} \rho V^2 < P_d^{max} \\ n_L &= \frac{\sqrt{L^2 + D^2}}{mg} < n_L^{max} \end{aligned} \quad (9)$$

where $\dot{Q}^{max}$, P_d^{max} and n_L^{max} represent the maximum acceptable heating rate, dynamic pressure and acceleration, respectively.

To ensure the terminal guidance, the state variable at the final time instant must satisfy the terminal constraints. Specifically, the error of altitude, speed and flight path angle at the final time instant should be less than a certain limit. This can be expressed as:

$$e_x = \begin{bmatrix} e_r \\ e_v \\ e_\gamma \end{bmatrix} = \begin{bmatrix} |r_f - r(t_f)| \\ |V_f - V(t_f)| \\ |\gamma_f - \gamma(t_f)| \end{bmatrix} \leq \begin{bmatrix} e_r^{max} \\ e_v^{max} \\ e_\gamma^{max} \end{bmatrix} \quad (10)$$

Remark 1. It should be noted that the terminal error constraints (described in Eq.(10)) can be removed if collocation schemes are used to discretize the dynamic model. Since for direct collocation methods, the terminal state of the trajectory can be set to the target state, whereas for other discretization methods like multiple shooting, it does not collocate the terminal state variables.

In general, for a dynamic mission profile, the no-fly zone constraints should be included for geopolitical consideration or threat/obstacle avoidance. For the case investigated in this paper, the no-fly zone constraints are described as circular exclusion zones with infinite altitude [4], [28]. Suppose the center of the j th no-fly zone is (θ_j, ϕ_j) and its radius is R_j , then the no-fly zone constraints can be modeled as:

$$NF = \arg \min \sqrt{(\theta - \theta_j)^2 + (\phi - \phi_j)^2} \geq R_j \quad (11)$$

$$j = 1, 2, \dots, N_{zone}$$

where N_{zone} is the number of no-fly zones.

III. AN EXTENDED NSGA-III ALGORITHM WITH CONSTRAINT HANDLING

A. Discrete-time optimal control model

This section presents the multi-objective optimization method used to solve the problem. It is worth noting that in [29], the authors designed and applied a fuzzy physical programming algorithm to transform multiple objectives into a single objective formulation. Subsequently, an interactive strategy was incorporated in the algorithm framework so as to better control the optimization process [30]. It was shown that this transformation technique is able to produce a compromised solution that can satisfy the designer's specific preference requirements. However, one main disadvantage of this approach is that its effectiveness is largely depended on the designer's experience and physical knowledge of the problem. If the preference regions defined by the mission planners are not accurate, then the results might not be credible.

The motivation of using MOEAs mainly relies on their ability in producing a set of approximated pareto-optimal solution. Moreover, it is easy to implement and requires less physical knowledge of the problem. In addition, it should be mentioned that for most existing gradient-based optimal control solvers, it is assumed in the implementation that all the objectives and constraints have continuous first and second order derivatives. In some practical control problems, however, the nonlinearity and nonsmoothness of the objectives or path constraints can be high. That indicates it is hard to obtain the gradient information for constructing the Jacobian and Hessian matrix. This problem becomes more difficult when the dimension of the objective function increases. Therefore, in order to solve the multi-objective optimal control problem constructed in Section II, an extended NSGA-III algorithm is proposed. Since the original NSGA-III algorithm has no capability in dealing with dynamic constraints (e.g. the vehicle dynamics), discrete techniques should be implemented such that the continuous-time problem can be transcribed to static NLPs. The discretization scheme applied in this paper is based on the multiple shooting technique. That is, the control variables are parameterized at temporal nodes $[t_0, t_1, \dots, t_f]$. Then, the equations of motion are integrated with a numerical integration method, e.g. forth-order Runge-Kutta method. For convenience, let $x_k = [r_k, \theta_k, \phi_k, V_k, \gamma_k, \psi_k]^T$ denotes the approximation of states at t_k time instant, and ξ_k stands for the step length for the k th time interval $[t_k, t_{k+1}]$. The discretized

version of the continuous-time problem with constraints is then formulated as:

$$\begin{aligned} & \text{minimize} && J = [J_1, J_2, J_3, J_4] \\ & \text{subject to} && x_{k+1} = x_k + \xi_k \sum_{i=1}^s b_i f(x_{ki}, u_{ki}) \\ & && x_{ki} = x_k + \xi_k \sum_{j=1}^s a_{ij} f(x_{kj}, u_{kj}) \\ & && g(x_{ki}, u_{ki}) \geq 0 \\ & && x(0) = x_0 \\ & && i, j = 1, \dots, s, \quad k = 0, \dots, N_k - 1 \end{aligned} \quad (12)$$

in which N_k denotes the number of discretized time nodes. $g(x, u)$ represents the inequality constraints given by Eq.(7)-(11). a_{ij} and b_i are discretization coefficients and $1 \leq i, j \leq s$. In Eq.(12), x_{kj} and u_{kj} are the intermediate state and control variables on the current time interval $[t_k, t_{k+1}]$. In addition, the intermediate time point t_{ki} holds $t_{ki} = t_k + c_i \xi_k$, $0 \leq c_1 \leq \dots \leq c_s \leq 1$. Eq.(12) is obtained by using an s -stage multiple shooting scheme. The order conditions with respect to multiple shooting discretizations for differential equations are detailed in Table I.

TABLE I: Order conditions of multiple shooting schemes

Order/Stage s	Conditions: $c_i = \sum_{j=1}^s a_{ij}$, $d_j = \sum_{i=1}^s b_i a_{ij}$
$s = 1$	$\sum b_i = 1$
$s = 2$	$\sum d_i = \frac{1}{2}$
$s = 3$	$\sum b_i c_i^2 = \frac{1}{3}, \sum d_i c_i = \frac{1}{6}$
$s = 4$	$\sum b_i c_i^3 = \frac{1}{4}, \sum d_i c_i^2 = \frac{1}{12},$ $\sum b_i c_i a_{ij} c_j = \frac{1}{8}, \sum a_{ij} c_j d_i = \frac{1}{24}$

Following the use of the direct multiple shooting technique, the resulting multi-objective NLP problem is solved by the NSGA-III algorithm, which is detailed in the next subsection of this paper.

B. Reference points generation

The framework of the classic NSGA-III algorithm is constructed based on the original NSGA-II approach with modifications in its selection mechanism. Both of these two techniques implement crossover and mutation operator to create the offspring generation. In addition, these two techniques use fast non-dominated sorting technique to assign the non-dominant rank for each individual. However, unlike NSGA-II, the maintenance of diversity among population members in the newly proposed method is preserved by creating and adaptively updating a number of well-distributed reference points. For completeness, a brief description of the way used to select the reference points is recalled.

It should be noted that the way of choosing reference points can either be predetermined in a structured manner or specified preferentially by the decision makers. In this study, as suggested in [31], a systematic approach that places points on a normalized hyperplane (an $(M-1)$ -dimensional unit simplex) is conducted such that the reference points can be equally distributed to all objective axes and have an intercept of one on each axis. The total number of reference points (H) can be determined by $H = C_{M+p-1}^p$, where M denotes the index of objectives, and p represents the number of divisions along each normalized objective. Take the entry trajectory optimization problem discussed in Section II as an example, if seven divisions ($p = 7$) is selected for each normalized performance

index axis, then for an $M = 4$ problem, $H = C_{4+7-1}^7$ or 120 reference points will be chosen. In NSGA-III, the main aim for designing a set of reference points is to associate each population members with each of these reference points in some sense. Since the created reference points are widely distributed on the entire normalized objective space, it can be expected that the obtained individuals are widely distributed on or near the true pareto-optimal front. The algorithm used to associate each individual can be found in [31].

Remark 2. It should be noted that in [32] and [33], two multi-objective evolutionary methods were tested to solve a multi-objective spacecraft orbital hopping problem. Both of these two algorithms used the crowding distance mechanism to maintain the diversity of population. That is, the last front members having the largest crowding distance values are chosen. However, based on the results obtained in [32] and other literature [34], this domination principle might lack the ability to assign sufficient selection pressures and emphasize feasible candidates, thus resulting in poor-distributed pareto front. Since the NSGA-III algorithm uses a niching process and a set of reference points distributed widely on the entire normalized objective-plane, it is likely to find near pareto-optimal solutions corresponding to the predefined reference points. Hence this algorithm is selected to generate the pareto-optimal solution for the problem considered in this paper.

C. Combination of improved NSGA-III with Runge-Kutta discretization scheme

By applying the multiple shooting discretization technique, a series of static NLP problem shown in Eq.(12) can be obtained. For simplicity reasons, this equation is further rewritten as:

$$\begin{aligned} & \text{minimize} \quad J = (J_1, J_2, J_3, J_4) \\ & \text{subject to} \quad h_k(u) = 0, \quad k = 1, 2, \dots, E, \\ & \quad \quad g_j(u) \leq g^*, \quad j = 1, 2, \dots, I, \\ & \quad \quad u_{\min} \leq u \leq u_{\max}, \quad u \in \mathbb{R}^{n_c \times N_k}. \end{aligned} \quad (13)$$

where I and E represent the total number of inequality and equality constraints, respectively. g^* is the maximum allowable value for the inequality constraints and n_c is the dimension of control variables. Since for multiple shooting method, only the control variables are parameterized, the optimization parameters of the corresponding NLP are the control sequences shown in Eq.(13). One advantage of using the combination of the multiple shooting and NSGA-III algorithm for solving the multi-objective optimal control problem is that the control box constraints (described in Eq.(8)) and equations of motion (Eq.(5)) can be satisfied automatically by initializing all population members within the specified lower and upper bounds and by integrating the dynamic model forward through numerical integration (e.g. RK-4). Specifically, if the initial population contains NP individuals, then all the decision variables can be generated randomly according to the limits of demanded angle of attack and bank angle (see Eq.(14)). This indicates that every decision variable can be in the feasible zone.

$$\begin{aligned} \alpha_c &= \alpha_c^{\min} + \text{rand}(\cdot) \times (\alpha_c^{\max} - \alpha_c^{\min}) \\ \sigma_c &= \sigma_c^{\min} + \text{rand}(\cdot) \times (\sigma_c^{\max} - \sigma_c^{\min}) \end{aligned} \quad (14)$$

Then combining all the optimization parameters, the structure of the individual can be defined by a matrix of decision variables as shown in Eq.(15), where $i = 1, 2, \dots, N_p$ and $j = 1, 2, \dots, N_k$.

$$u_i = \begin{pmatrix} \alpha_c^{1,1}, & \dots & \alpha_c^{1,j}, & \dots & \alpha_c^{1,N_k}, \\ \vdots & & \vdots & & \vdots \\ \alpha_c^{i,1}, & \dots & \alpha_c^{i,j}, & \dots & \alpha_c^{i,N_k}, \\ \vdots & & \vdots & & \vdots \\ \alpha_c^{N_p,1}, & \dots & \alpha_c^{N_p,j}, & \dots & \alpha_c^{N_p,N_k}, \\ \sigma_c^{1,1}, & \dots & \sigma_c^{1,j}, & \dots & \sigma_c^{1,N_k}, \\ \vdots & & \vdots & & \vdots \\ \sigma_c^{i,1}, & \dots & \sigma_c^{i,j}, & \dots & \sigma_c^{i,N_k}, \\ \vdots & & \vdots & & \vdots \\ \sigma_c^{N_p,1}, & \dots & \sigma_c^{N_p,j}, & \dots & \sigma_c^{N_p,N_k} \end{pmatrix} \quad (15)$$

Moreover, another advantage of using multiple shooting discretization scheme is that a good accuracy can also be achieved and controlled by the user if the step length of the temporal nodes is small enough [35]. In terms of the discretization scheme used in this paper, the error order can be approximated as $\mathcal{O}(\|\frac{T}{N_k}\|_\infty^s)$, in which $T = t_f - t_0$ is the time duration of the mission.

IV. DIFFERENT CONSTRAINT HANDLING STRATEGIES

Normally, most of the spacecraft trajectory optimization problems are highly constrained. However, there is not sufficient literature on handling strict equality and inequality path constraints in a multi-criteria optimal control problem, since most of the existing multi-objective evolutionary solvers were developed for solving unconstrained problems. In [36], the author extended the MODE/D to a constrained MOEA/D-DE algorithm and suggested to use a penalty function P to handle constraints. However, it is usually difficult to find a proper balance between the objective term and the path constraint violation term for the studied problem.

To effectively deal with the infeasibility of the control sequence, this study applies the definition of constraint violation degree V and this will be the primary metric used for the constraint handling strategies illustrated in the following subsections [19]. In terms of the inequality constraints described in the optimization model (Eq.(13)), the violation degree for relation " $\leq$ " (i.e. $g_j(u) \leq g_j^*$) can be defined as:

$$\mu_{g_j}(u) = \begin{cases} 0, & g_j(u) \leq g_j^*; \\ \frac{g_j(u) - g_j^*}{g_j^{\max} - g_j^*}, & g_j^* \leq g_j(u) \leq g_j^{\max}; \\ 1, & g_j(u) \geq 2g_j^*. \end{cases} \quad (16)$$

where $g_j(u)$ represents the j th constraint value for each individual. Based on Eq.(16), the tolerance range is $(g_j^*, g_j^{\max})$, where $g_j^{\max}$ stands for the maximum value of the constraint for the individual. If an individual can satisfy the constraint g_j , then according to Eq.(16), its violation degree value should equal to 0.

For equality constraints (i.e. $h_k(u) = h_k^*$), the tolerance region $(h_k^{\min}, h_k^{\max})$ can be chosen based on the minimum

and maximum constraint value for the individual. Therefore, the relation “=” is defined as:

$$\mu_{h_k}(u) = \begin{cases} 1, & h_k(u) \geq h_k^{max}; \\ \frac{h_k(u) - h_k^*}{h_k^{max} - h_k^*}, & h_k^* \leq h_k(u) \leq h_k^{max}; \\ 0, & h_k(u) = h_k^*; \\ \frac{h_k^* - h_k(u)}{h_k^* - h_k^{min}}, & h_k^{min} \leq h_k(u) \leq h_k^*; \\ 1, & h_k(u) \leq h_k^{min}. \end{cases} \quad (17)$$

Then the constraint violation for each individual in the current generation can be calculated as:

$$V_i = \sum_{j=1}^I \mu_{g_j}(u) + \sum_{k=1}^E \mu_{h_k}(u), \quad i = 1, 2, \dots, NP \quad (18)$$

Based on the violation degree function established above, each individual can be associated with all the constraints. The magnitude of the solution infeasibility can be directly reflected by the value of the violation function.

A. Superiority of feasible solution method

The superiority of feasible solution (SF) constraint handling strategy is introduced in this subsection. From Eq.(18), the violation degree of the i th individual is the sum violation of all the constraints and based on this definition, the V-based dominant rule “ $\succ$ ” can be given by:

Definition 1. [37] (V-based dominant rule “ $\succ$ ”) For two individuals u_1 and u_2 in the current population, u_2 is said to be dominated by u_1 if and only if one of the following relationships is satisfied:

- 1) $V(u_2) > V(u_1) > 0$.
- 2) $V(u_1) = 0, V(u_2) > 0$.
- 3) $V(u_1) = V(u_2) = 0$, and for each objective function i , $J_i(u_1) < J_i(u_2)$ is satisfied (Classic dominance definition).

As shown in Definition.1, the feasible individual can always dominate the infeasible one, while the individual with smaller violation degree always dominates the one with higher violation value. After the V-based dominance relationships are determined, each candidate among the current population can be divided into different ranks. It should be noted that for highly constrained spacecraft trajectory optimization problems, it is likely that all of the individual among the population are infeasible solution in the first several generations. Then, three problem types can be introduced. If all the individuals in the current population are infeasible solutions, the problem type is set to 0, whereas if some of the individuals are feasible solutions, the problem type is set to 0.5. Correspondingly, *Problem type* = 1 means all the candidates among the population are feasible solutions. To improve the algorithm efficiency, for the first several generations (*Problem type* = 0), the non-dominant rank can be simply assigned by sorting the violation degree of the individuals such that the computational complexity can be reduced. Supposing the NSGA-III algorithm, together with the SF strategy, is adopt to solve a typical high index spacecraft trajectory optimization problem defined in Section II, the overall optimization procedure is summarised in the Pseudocode (see Algorithm 1).

Algorithm 1 Overall procedures of the SF-based algorithm

```

1: Initialize the first population  $P_1 = (u_1, \dots, u_{N_p})$  and other
   control parameters of the proposed algorithm;
2: /*Main Loop*/
3: for generation  $G := 1, 2, \dots, G_{max}$  do;
4:   (a). Calculate the objective function values  $J$  and the
5:     violation degree according to the Eq.(16)-(17) for
6:     each individual;
7:   (b). Generate offspring population  $Q_G$  by using the
8:     recombination and mutation processes [31];
9:   (c). Combine  $Q_G$  with  $P_G$  to obtain  $R_G$  (e.g.  $R_G =$ 
10:     $Q_G \cup P_G$ );
11:   (d). Specify the problem type by checking the violation
12:     degree of individuals;
13:   if Problem type = 0 then
14:     Get all non-dominated ranks by sorting the
15:     violation degree.
16:   end if
17:   if Problem type = 0.5 or 1 then
18:     assign all non-dominated ranks using the
19:     V-based dominant rule.
20:   end if
21:   (f). Select the best  $N_p$  individuals as the candidates of
22:     the new generation  $S_G$  via reference point-
23:     based selection operation [38];
24:   (g). Set  $G = G + 1$ ;
25: end for
26: Output the optimal control pareto solution;

```

B. Penalty function based method

Another constraint handling strategy that can be applied to deal with infeasible solutions is the penalty function (PF) based method. By applying the violation degree information, this approach transforms a constrained problem into an unconstrained version. To penalize the infeasible solution among the population, a constraint violation term is introduced to form an augmented fitness function. This process can be given by:

$$J(u) = \begin{cases} J(u), & \text{if all constraints can be satisfied;} \\ J(u) + P, & \text{otherwise.} \end{cases}$$

where the penalty term $P = c(\sum_{j=1}^I \mu_{g_j}(u) + \sum_{k=1}^E \mu_{h_k}(u))$, where c stands for a positive constant named penalty factor. After calculating the augmented fitness function for each individual, the classic nondominant sorting process will be processed to rank all the candidates. Compared with the SF strategy, PF method tends to have better capability to maintain the population diversity. This is because in SF, feasible solutions can always dominate the infeasible one. As a result, some “good” infeasible solutions will be removed from the population. If a problem contains disconnected feasible regions, the SF method might not perform properly [26].

C. Multi-objective constraint handling technique

The multi-objective constraint handling strategy (MOCH) is based on the concept of multi-objective optimization. It transcribes the constrained multi-objective problem into an uncon-

strained version by defining the total constraint violation value as an additional objective, thereby increasing the objective space by one. More precisely, for the problem considered in this paper, we can define $J_5 = \min \sum_{j=1}^I \mu_{g_j}(u) + \sum_{k=1}^E \mu_{h_k}(u)$.

Different from the PF technique, the tuning process of the penalty factor is no longer necessary for MOCH. However, one of the main disadvantages with respect to this strategy is that it may result in a significant increase in terms of the processing time. Besides, the extra number of objective has negative influences in terms of the searching procedures [23].

The aforementioned three constraint handling methods are all established and embedded in the proposed evolutionary optimal control solver. In this study, we are interested in finding the most suitable constraint handling method to help the infeasible trajectories to quickly move toward the feasible path region.

D. Computational complexity analysis

The computational complexity of one generation of the proposed algorithm for solving trajectory optimization problems is presented in this subsection. Since each individual in the population represents a trajectory, the initialization of all the trajectories using multiple-shooting scheme requires $\mathcal{O}(sN_pN_k)$ computations, where s is the number of inner steps in the Runge-Kutta scheme (e.g. if forth-order Runge-Kutta scheme is used, then $s = 4$). According to [31], the calculation and normalization of objectives require $\mathcal{O}(M^2N_p)$ computations, whereas $\mathcal{O}(MN_p^2)$ computations are needed for the association of population members with reference points.

The computational complexity of the three constraint handling strategies is then analyzed. $\mathcal{O}(N_pE)$ and $\mathcal{O}(N_pI)$ computations are required to calculate the equality and inequality constraints of a $2N_p$ population members, respectively. If the PF method is chosen as the constraint handling method, the calculation of the augmented fitness function requires $\mathcal{O}(N_p)$ operations. Based on the augmented fitness value, the classic nondominant sorting process will be implemented to rank all the candidates, and this requires $\mathcal{O}(N_p \log^{M-2} N_p)$ operations [39]. For the SF method, as indicated in Algorithm.1, the identification of the problem type requires $\mathcal{O}(N_p)$ operations. Once the problem type is determined, the operation times for *Problem type* = 1 are $\mathcal{O}(N_p \log^{M-2} N_p)$, which is the same with the classic non-dominant sorting. For *Problem type* = 0.5, the worst-case computational complexity can be reduced to $\mathcal{O}((N_p - N_{inf}) \log^{M-2}(N_p - N_{inf})) + \mathcal{O}(N_{inf} \log N_{inf})$, where N_{inf} stands for the number of infeasible solutions among the current population. As for *Problem type* = 0, the worst-case computational operation is $\mathcal{O}(N_p \log N_p)$. From the complexity analysis, it is obvious that for *Problem type* = 0.5 and 1, the operations required for the SF strategy are less than that of PF.

Essentially, the SF method divides the solution-finding process into two steps: finding feasible candidates and optimizing the objectives. In the first step, the only information used is the constraint violation value. It considers that any feasible solution can be better than the infeasible one. This allows the SF algorithm can get quickly rid of infeasible solutions.

As for the MOCH approach, since the total constraint violation value is treated as an additional objective function, the objective space is increased by one and the problem becomes an unconstrained version. Therefore, the computational complexity of the sorting process is increased to $\mathcal{O}(N_p \log^{M+1-2} N_p)$, which is generally higher than the SF and PF methods.

V. EXPERIMENTAL RESULTS

In this section, the simulation results of the proposed algorithm on the four-objective spacecraft entry trajectory optimization problem described in Section II are presented. The vehicle's mass is given by $m = 6309.43\text{slug}$, while the final state error constraints are set as $e_r^{max} = 3000\text{ft}$, $e_v^{max} = 350\text{ft/s}$ and $e_\gamma^{max} = 1\text{deg}$. The following parameters of three no-fly zone constraints are used to construct the corresponding inequality constraints:

- 1). Center: $\theta_1 = 30^\circ$, $\phi_1 = 7.5^\circ$; Radius: $R_1 = 5^\circ$;
- 2). Center: $\theta_2 = 60^\circ$, $\phi_2 = 15^\circ$; Radius: $R_2 = 5^\circ$;
- 3). Center: $\theta_3 = 70^\circ$, $\phi_3 = 20^\circ$; Radius: $R_3 = 5^\circ$;

The parameters of the extended NSGA-III are given as follows: the crossover probability $p_c = 1.0$, crossover index $eta_c = 20$ and mutation index $eta_m = 20$. The maximum number of generations is given as 500. In addition, the whole entry flight time is set as 2000s and all the results are displayed with a fixed grid of $N_k = 200$ points. In this way, the optimization parameter for the resulting NLP problem becomes $n_c \times N_k = 2 \times 200 = 400$. The discretization parameters b_i and c_i used for the simulation are assigned as $b_i = (\frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{1}{6})$ and $c_i = (0, \frac{1}{2}, \frac{1}{2}, 1)$. The non-zero elements of a_{ij} are $a_{21} = 0.5$, $a_{32} = 0.5$ and $a_{43} = 1$, respectively. It is obvious that a_{ij} , b_i and c_i follow the conditions ($s = 4$) given by Table.I

Remark 3. For most optimal control problems, since the problem may have discontinuous points or nonsmooth segments, the quality of solution tends to be affected by the mesh refinement process. This process is designed to analyze if the current temporal set is reasonable and adaptively update the temporal set. However, it is difficult and unrealistic to incorporate this process in an evolutionary-based NLP solver. Therefore, to decrease the effect of nonsmoothness segments and discontinuity points, a large number of time nodes, which is uniformly distributed along the whole time history are chosen and fixed as the discrete temporal points.

A. Performance evaluation indicators

To evaluate the performance of the constructed method, certain evaluation metrics established in the multi-objective domain should be introduced and adopt. Two performance metrics that have been widely applied are the inverted generational distance (IGD) and the hypervolume value (HV). For completeness, these two performance metrics are introduced.

1). IGD [40]: Inverted generational distance is an indicator that has been favored by researchers. This value reflects both the convergence and diversity of the obtained pareto-optimal solution. To calculate this performance metric, the information of the true pareto front should be utilized.

However, this information is usually unavailable for most practical multi-objective engineering optimization problems.

2). HV [40]: The hypervolume value of the calculated pareto front is the size of the objective space dominated by those solutions. This indicator can be used to reflect the spread of the approximated solution on the objective space. To compute the HV metric, the following equation can be applied:

$$HV(P) = Leb(\bigcup_{u \in P} [J_1(u), R_1] \times \cdots \times [J_M(u), R_M]) \quad (19)$$

in which $Leb(\cdot)$ denotes the Lebesgue measure. $R = [R_1, R_2, \dots, R_M]^T$ represents a reference point which is dominated by all points on the pareto front. Compared with the IGD indicator, one important advantage of using the HV metric is that it does not rely on the true pareto-optimal solution. Therefore, in this paper, the HV value is applied as the primary performance metric for the experimental study.

B. Performance of different constraint handling methods

As discussed in Section IV, a comparative study in terms of the performance of different constraint handling strategies is firstly carried out. Three mission cases are tested in order to illustrate the performance of constraint handling methods under different path constraint settings. These mission cases are summarised as follows:

- Case 1: $[\dot{Q}^{max}, P_d^{max}, n_L^{max}] = [200, 280, 2.5]$.
- Case 2: $[\dot{Q}^{max}, P_d^{max}, n_L^{max}] = [150, 250, 1.25]$.
- Case 3: Case 1 with no-fly zone constraints.

TABLE II: Performance of constraint handling methods

Factor	Case 1			Case 2			Case 3		
	SF	PF	MOCH	SF	PF	MOCH	SF	PF	MOCH
I_{fst}	19	45	97	65	88	201	52	60	164
I_{all}	52	83	228	124	167	484	97	129	426
N_f	50	42	17	37	26	3	45	31	7

By limiting the maximum number of generations as 500, 20 trials were carried out independently for Case 1 to 3 using different constraint handling methods. The corresponding statistical results are tabulated in Table II. This table contains three indicators. The first indicator I_{fst} is the average iteration for finding the first feasible solution, while the second indicator I_{all} represents the average iteration for making all the population member feasible. The third indicator N_f stands for the average number of pareto-optimal solutions obtained.

As can be seen from the statistical results, the SF method can generally perform better than its counterparts in terms of finding feasible solutions for different mission cases. Based on the N_f results, it can be observed that getting rid of infeasible solution quickly can have positive influences in terms of obtaining more pareto-optimal solutions for the multi-objective spacecraft entry trajectory design problem. Moreover, when the constraint condition becomes tighter, the MOCH method might fail to drive all candidate solutions into the feasible region, thereby resulting in limited final pareto set. Therefore, although the use of SF might lose population diversity compared with PF and MOCH techniques, it tends to be more competitive for handling the investigated problem with various mission constraints.

C. Flight trajectory problem without no-fly zone constraint

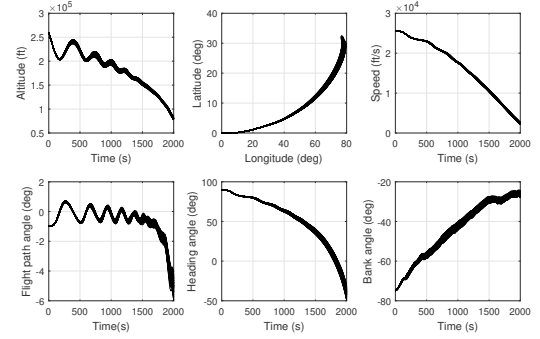


Fig. 1: Time history of states and controls (case 1)

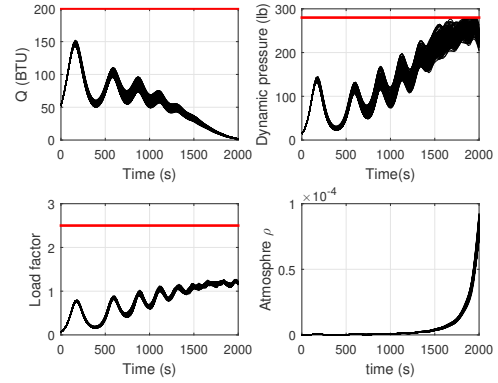


Fig. 2: Time history of path constraints (case 1)

This subsection presents the optimal results obtained using the extended NSGA-III algorithm. As analyzed in Section V.A, the SF technique is implemented as the constraint handling strategy. Prior to displaying the optimal trajectory in detail, the relationship between different objectives is analyzed. From the objective and system equations (e.g. Eq.(5) and Eq.(6)), it can be analyzed that an increase in the cross range ϕ will result in a slower deceleration, thereby generating more heat load. Hence maximizing the final latitude (cross range) and minimizing the aerodynamic heating are conflicting objectives. In addition, it is worth noting that the maximum heat flux value is usually achieved at the point where the vehicle completes the first descent phase. Minimizing the maximum heat flux value tends to enlarge the magnitude of the continuing skip hops so as to achieve the final velocity value, which means the path smoothness will be sacrificed. Thus minimizing the heat flux and minimizing the path smoothness indicator are also conflicting objectives.

The optimal trajectories obtained without considering the no-fly zone constraint are firstly presented. Based on the vehicle dynamics, mission objectives and constraints illustrated in Section II, the results in the last population are shown in Fig.1 and Fig.2.

As can be observed from Fig.1 and Fig.2, the proposed multi-objective optimal control solver manages to generate state trajectories between the predetermined initial and final

conditions without violating path constraints. Therefore, the thermal and structural safety of the entry vehicle can be guaranteed, which is the prerequisite for the validity of an optimal control solver to spacecraft trajectory optimization problems. To effectively capture the true behaviour of the flight vehicle, a high density of mesh grids is required. However, the high density of nodes tends to have negative influences in terms of the evolution and convergence processes. One way to handle this problem is to introduce the lag equation as described in Eq.(8). The function of the lag equation is similar to a first-order filter, that can filter the control input signal and make the trend in the actual control history smoother. This is apparent in Fig.1 (see the bank angle profile).

Since all the trajectory in the final population are feasible solutions, the next step is to verify the pareto-optimality and improvement of the proposed algorithm over other multi-objective solvers. An attempt is made to compare the solutions obtained using the proposed method, MOPSO [41] and MOEA/D [21] approaches on maximizing final latitude versus minimizing heating pane. Non-optimal results may produce higher heat loads than necessary for a desired cross range. Fig.3 shows the Pareto fronts obtained by different evolutionary approaches investigated in this paper. As illustrated in Fig.3, the improved NSGA-III algorithm generally preforms better than other multi-objective heuristic solvers for solving the multi-criteria trajectory planning problem, since the pareto set calculated by applying the proposed technique can cover the pareto front calculated using other heuristic algorithms. More precisely, the first front set (rank 1) obtained by the proposed algorithm contains 49 pareto-optimal solutions, whereas there are only 39 and 23 pareto-optimal solutions calculated using MOEA/D and MOPSO lying on the first front, respectively. In addition, the distribution of pareto fronts obtained using the extended NSGA-III is more uniform in the objective space. This is because the supply of a set of well-distributed reference points and the niching methodology in searching pareto-optimal solutions associated with every reference point have made the diversity preservation of obtained solutions more efficient and reliable.

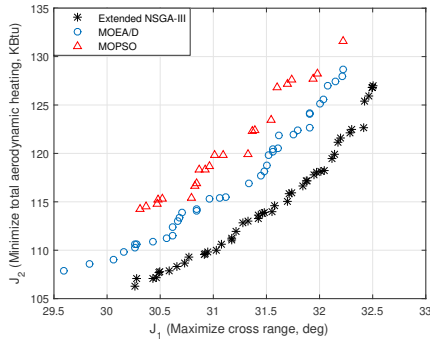


Fig. 3: Pareto front obtained via three solvers (Case 1)

Due to the randomness of evolutionary-based optimization methods in initialization, it is not enough to analyze the simulation results in only one trial. In order to eliminate the influences of randomness, this study has implemented three

TABLE III: Statistical boundary results (mission case 1)

Extended NSGA-III	$e_r(ft)$	$e_v(ft/s)$	$e_\gamma(deg)$
Maximum	2773.92	240.88	0.95
Minimum	16.60	5.27	0.00
Average	1047.74	106.50	0.34
Standard deviation	786.57	64.43	0.26
MOPSO	$e_r(ft)$	$e_v(ft/s)$	$e_\gamma(deg)$
Maximum	2977.40	316.04	1.00
Minimum	24.18	19.49	0.00
Average	1313.51	127.52	0.50
Standard deviation	979.56	85.45	0.29
MOEA/D	$e_r(ft)$	$e_v(ft/s)$	$e_\gamma(deg)$
Maximum	2943.56	267.81	0.99
Minimum	29.27	12.03	0.00
Average	1169.23	162.53	0.39
Standard deviation	866.93	85.71	0.29

TABLE IV: Statistical objective results (mission case 1)

Extended NSGA-III	J_1	J_2	J_3	J_4
Maximum	32.51	126.99	1.17	150.01
Minimum	30.26	106.32	0.73	143.77
Average	31.52	115.27	0.95	145.96
Standard deviation	0.55	5.29	0.11	1.74
Variance	0.31	27.99	0.01	3.05
MOPSO	J_1	J_2	J_3	J_4
Maximum	32.23	131.63	1.17	154.19
Minimum	30.31	114.21	0.82	145.27
Average	31.14	120.66	1.01	148.08
Standard deviation	0.69	6.21	0.08	2.03
Variance	0.48	38.59	0.01	4.11
MOEA/D	J_1	J_2	J_3	J_4
Maximum	32.22	128.67	1.20	152.85
Minimum	29.59	107.86	0.78	144.78
Average	31.16	117.40	0.98	148.29
Standard deviation	0.73	6.15	0.12	1.82
Variance	0.54	37.89	0.01	3.32

evolutionary-based solvers to run each mission scenario in 20 trials independently. The statistical comparison of the solutions obtained in 20 trials is tabulated in Table.III and Table.IV. Specifically, the final boundary results are shown in Table.III, whereas the objective results are shown in Table.IV. It can be observed from Table.III that compared with other MOEA solvers, the proposed strategy tends to have a robust and stable behaviour in terms of achieving the final boundary conditions.

Regarding the specific efficiency of the calculation, the average processing time for the optimization procedure is about 1h 12m (4331.2s). It is important to remark that for practical spacecraft guidance and control systems, the design of optimal flight trajectories is usually carried out offline. In addition, if parallel computing or high-performance computers can be implemented to optimize the flight path, this processing time can be further decreased.

D. Flight trajectory results with tighter path constraints

To further test the performance of the solver and constraint handling strategy, another mission scenario (Case 2) which sets tighter requirements on the path constraints was carried out using the improved NSGA-III algorithm. In this case, the aerodynamic heating, dynamic pressure and load factor constraints are restricted to $[\dot{Q}^{max}, P_d^{max}, n_L^{max}] = [150, 250, 1.25]$, respectively. The time history of the optimization variable is plotted in Fig.4. Detailed results in terms of terminal error values and different objective function values are given in Table.V and Table.VI, respectively.

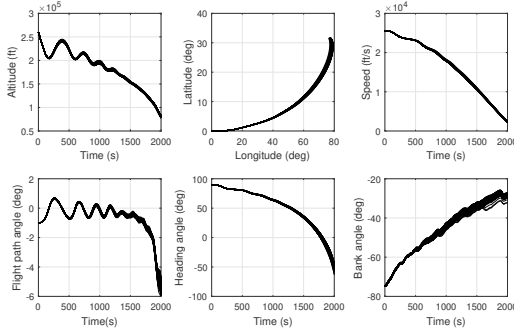


Fig. 4: Time history of states and controls (case 2)

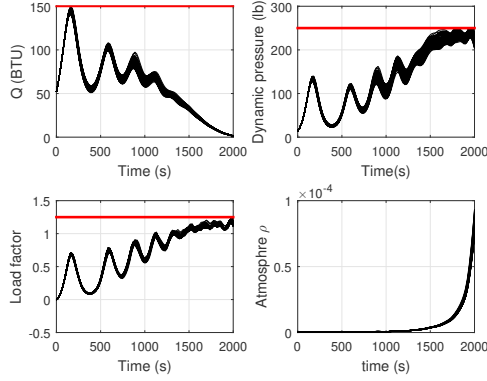


Fig. 5: Time history of path constraints (case 2)

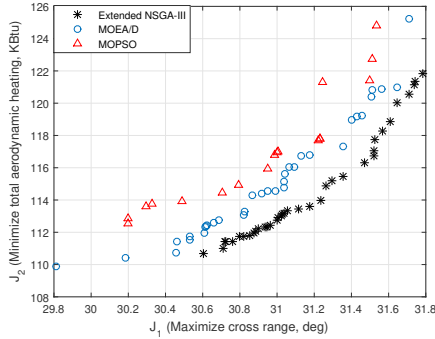


Fig. 6: Pareto front obtained via three solvers (case 2)

TABLE V: Statistical boundary results (mission case 2)

Extended NSGA-III	$e_r(ft)$	$e_v(ft/s)$	$e_\gamma(deg)$
Maximum	1962.88	246.97	0.95
Minimum	2.84	27.31	0.01
Average	486.15	173.15	0.55
Standard deviation	604.14	47.37	0.21
MOPSO	$e_r(ft)$	$e_v(ft/s)$	$e_\gamma(deg)$
Maximum	2546.23	339.40	1.00
Minimum	7.08	102.37	0.23
Average	840.22	193.29	0.67
Standard deviation	639.43	86.54	0.31
MOEA/D	$e_r(ft)$	$e_v(ft/s)$	$e_\gamma(deg)$
Maximum	2947.33	307.27	0.96
Minimum	7.29	48.21	0.09
Average	888.19	189.94	0.57
Standard deviation	862.86	65.92	0.24

TABLE VI: Statistical objective results (mission case 2)

Extended NSGA-III	J_1	J_2	J_3	J_4
Maximum	31.78	121.83	1.19	146.46
Minimum	30.61	110.70	0.84	142.23
Average	31.16	114.58	1.04	143.65
Standard deviation	0.35	3.31	0.07	0.93
Variance	0.12	10.98	0.01	0.86
MOPSO	J_1	J_2	J_3	J_4
Maximum	31.53	124.82	1.20	148.44
Minimum	29.81	109.88	0.92	142.01
Average	30.91	117.02	1.10	144.73
Standard deviation	0.45	3.55	0.08	0.94
Variance	0.21	12.61	0.01	0.89
MOEA/D	J_1	J_2	J_3	J_4
Maximum	31.71	125.20	1.25	148.32
Minimum	29.81	109.87	0.91	142.02
Average	30.95	115.25	1.05	144.60
Standard deviation	0.44	3.69	0.07	1.17
Variance	0.19	13.64	0.01	1.37

From Fig.4, the obtained state and control profiles are smooth enough and can vary in their tolerant set during the entire time history. It should be noted that one important factor that can have significant influences in the results is the atmospheric model ρ . Similarly with the result presented in Fig.2, the path constraint profiles, together with the ρ profile, are displayed in Fig.5. From Fig.5, it is obvious that the dynamic pressure and load factor constraint values increase significantly as ρ increases. This phenomena affects the characteristic of the altitude, flight path angle and bank angle trajectories shown in Fig.1 and Fig.4. For example, after around 1500s, the altitude profile tends to be much smoother, which is mainly influenced by the flight path angle. This can be explained by the fact that the oscillations in the trajectory tend to make the dynamic pressure and load constraints becoming active. In order to decrease these two constraint values, the variance of the flight path angle should be decreased and this is mainly achieved by increasing the bank angle gradually.

According to the results shown in Table.V and Table.VI, the proposed solver can again achieve smaller final state errors and better objective results than other multi-objective algorithms. In addition, Fig.6 gives the Pareto fronts obtained by the three approaches on maximizing final latitude versus minimizing heating pane. Again, the improved NSGA-III performs better than its counterparts since the pareto front obtained using improved NSGA-III can cover fronts calculated using other methods and tends to be well-distributed. Specifically, for this case study, the improved NSGA-III algorithm obtains 37 pareto-optimal solutions in the first front set, while the first front set calculated using MOEA/D and MOPSO approaches only contains 34 and 18 elements, respectively. Therefore, although all the evolutionary-based solvers considered in this paper can be applied to generate feasible solutions for solving multi-objective trajectory optimization problems, the improved NSGA-III approach has quicker convergence speed and better global search ability under limited computational power. This further confirms that the designed method can have the advantage over other evolutionary-based multi-discipline solvers considered in this paper.

E. Flight trajectory problem with no-fly zone constraint

In this case scenario, the no-fly zone constraint is taken into account during the entry phase (Case 3). The testing of the results is to determine how well the numerical solution obtained using the proposed algorithm avoids different types of constraints and achieves multiple performance indices. From the results, the maximum value of heating, dynamic pressure and load factor among all the population are 154.93, 279.56 and 1.4268, respectively. All the final state errors are in the tolerance region (defined in Eq.(10)), and the violation degree of no-fly zone path constraints is equal to 0. That means the proposed algorithm can also be effective for this mission case. To better illustrate the no-fly zone constraints, the position profile has been plotted in Fig.7 with the solution without considering no-fly zone constraints.

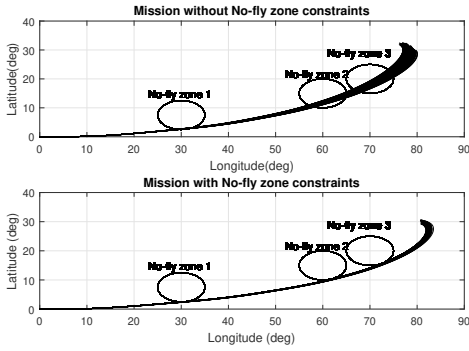


Fig. 7: Flight path with and without No-fly zone constraints

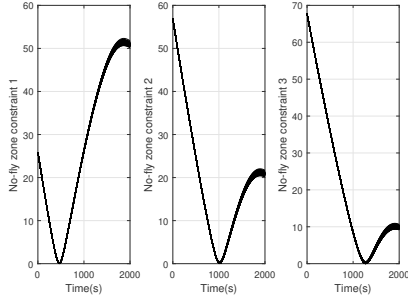


Fig. 8: No-fly zone path constraints

The time history with respect to the three no-fly zone path constraints is illustrated in Fig.8, where the first no-fly zone path constraint becomes active (shown in Eq.(11)) at around 500s, while the second and third no-fly path constraints become active at around 1000s and 1300s, respectively. In order to avoid the no-fly region, the vehicle should use a higher longitudinal acceleration, thereby allowing more flexibilities on the lateral direction. Compared with state and control variables shown in Fig.1, the bank angle tends to increase slower when the time past 500 second. This can result a smaller heading angle of the vehicle, which implies that the variance of longitude tends to increase, whereas the variance of latitude tends to decrease. Moreover, a higher speed of the vehicle can also have positive influences in terms of increasing the acceleration of longitudinal speed. Therefore, the variance

of longitude tends to become higher than the one without considering no-fly zone constraints.

To better investigate the performance of different multi-objective methods, the statistical results of the HV metric obtained using different algorithms for the three mission cases are analyzed. By setting the reference point R as $R = [28.5, 150, 1.5, 200]^T$, the HV results are then tabulated in Table.VII.

TABLE VII: Best, medium and worst HV results

Mission case	HV		
	MOEA/D	MOPSO	Proposed method
1	5.0207E+03	4.0151E+03	6.3310E+03
	5.0036E+03	4.0042E+03	6.3043E+03
	4.9848E+03	3.9938E+03	6.2908E+03
2	3.8943E+03	2.9168E+03	4.5112E+03
	3.8863E+03	2.9129E+03	4.5020E+03
	3.8797E+03	3.9093E+03	4.4919E+03
3	2.1338E+03	2.1290E+03	2.4064E+03
	2.1272E+03	2.1249E+03	2.3995E+03
	2.1224E+03	2.1154E+03	2.3916E+03

From this table, it is clear that the extended NSGA-III method can generally outperform the other two MOEAs in terms of HV for the three flight mission cases. Therefore, it can be concluded that it advantageous to implement the proposed algorithm for addressing the multi-objective spacecraft entry trajectory design problem.

VI. PARAMETER ANALYSIS

In this section, the impact of different variables on the obtained pareto solutions will be studied. The analysis includes: 1). the impact of the initial population; 2). the impact of the uncertain variables V and m ; and 3). the influence of the number of discretization node N_k .

A. Impact of initial population N_p

The impact of the initial population is firstly studied. As suggested in [31], the population size of the NSGA-III algorithm should be assigned as the smallest multiple of four greater than $H = C_{4+p-1}^p$. Hence, the testing size levels are assigned as: $N_p^{(1)} = 56$, $N_p^{(2)} = 120$ and $N_p^{(3)} = 220$ (referred as *Level 1*, *Level 2* and *Level 3*, respectively). The statistical results of the HV value for the three algorithms are tabulated in Table.VIII.

TABLE VIII: Best, medium and worst HV results (Different N_p)

Level	HV		
	MOEA/D	MOPSO	Proposed method
1	1.2811E+03	6.0568E+02	9.1488E+02
	1.2753E+03	6.0138E+02	9.1076E+02
	1.2693E+03	5.9757E+02	9.0799E+02
2	2.2316E+03	1.7224E+03	2.8136E+03
	2.2266E+03	1.7145E+03	2.8084E+03
	2.2192E+03	1.7086E+03	2.8017E+03
3	5.0170E+03	4.0143E+03	6.3282E+03
	5.0037E+03	4.0024E+03	6.3059E+03
	4.9910E+03	3.9911E+03	6.2844E+03

As can be observed from Table.VIII, under a relatively small population setting, the MOEA/D algorithm has the best performance among the three algorithms investigated in this paper. However, when the population size experiences an

increase, the performance of the presented approach becomes better than its counterparts. This indicates that the developed algorithm tends to work well if a relatively large population size setting can be available for the optimization process.

B. Impact of uncertain variables

This subsection analyzes how the uncertain variables affect the pareto results. Specifically, when the vehicle enters the atmosphere, the vehicle's mass m may experience a fraction. Therefore, an uncertain assessment is firstly performed with regard to the uncertain mass value.

Take mission case 1 as an example, we randomly pick a point from the obtained pareto front shown in Fig.3 (e.g. $p_1(J_1, J_2) = [32.05, 118.37]$). When mass uncertainty is introduced into the problem (e.g. $\delta m = +10\%$), the resulting uncertainty-perturbed objective values become $[\bar{J}_1, \bar{J}_2] = [32.13, 121.61]$, which is deviated from the original pareto front result. This further confirms that it is necessary to carry out an uncertain assessment. By setting the mass value uncertainty as $\delta m = +5\%$ and $\delta m = +10\%$, the updated pareto front result is displayed in Fig.9, from where it can be seen that as δm increases, the obtained pareto front result moves away from the original pareto solutions. Therefore, it is meaningful to take into account the mass uncertainty in the practical design of optimal spacecraft entry trajectories.

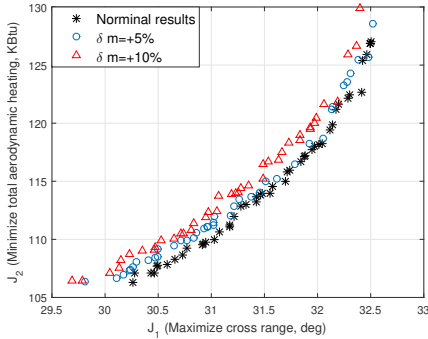


Fig. 9: Pareto results with uncertainty

Another uncertain assessment is also performed to investigate the impact of the uncertain variable V . It is worth noting that $\dot{V}$ is a function of gravity acceleration g , which is treated as a constant in the previous experiments. To capture more reality, g is modeled as $g = \mu/r^2$ in the analysis. By performing the simulation, it was found that the difference between using a constant g and $g = \mu/r^2$ in the $\dot{V}(g)$ equation is not obvious.

C. Impact of N_k and other discretization schemes

The effect of the number of discretized time nodes N_k is analyzed in this subsection. Since the objective value of J_2 will experience a significant variation during the evolutionary process, the goal attainment value of J_2 is used as the indicator to evaluate the convergence speed of the evolutionary process. The goal attainment value μ_{J_2} is calculated by $\mu_{J_2} = 1 - \frac{J_2 - J_2^*}{J_2^{max} - J_2^*}$, where J_2^* and J_2^{max} are set to 135 and

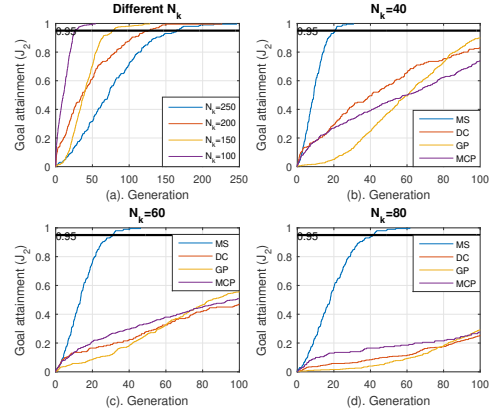


Fig. 10: Convergence results for different N_k and methods

1000, respectively. By specifying $N_k = (100, 150, 200, 250)$, the result is presented in Fig.10(a).

Fig.10(a) displays the result on the generation versus goal attainment value plane. As discussed in Section III, the continuous-time system is discretized at temporal nodes. As a result, a larger N_k can improve the approximation accuracy of the state and control trajectories. However, a larger N_k will increase the number of optimization parameter and therefore, it might have negative influences in terms of the convergence speed of the optimization process. As shown in Fig.10(a), the speed of achieving a higher goal attainment value experiences a decrease as the number of N_k increases.

An attempt is made to apply the NSGA-III method with other discretization algorithms such as the direct collocation (DC) analyzed in [11], the Gauss pseudospectral method (GP) outlined in [13], and a modified Chebyshev pseudospectral technique (MCP) reported in [12]. By setting $N_k = (40, 60, 80)$, the results are displayed in Fig.10(b)-(d). From the results, it can be found that compared with other discretization schemes, applying the multiple shooting-based (MS) NSGA-III algorithm can achieve satisfactory results in fewer iterations and it has the flexibility to use a relatively large temporal set. This is because in direct collocation and pseudospectral methods, both the state and control variables will be discretized and consequently, the equations of motion are transcribed to a series of algebraic equations (equality constraints). This implies that the resulting number of optimization parameters and equality constraints tends to be large and cannot be satisfied automatically. For a given optimal control problem which contains n_s state variables and n_c control variables, if N_k collocation points are used to parameterized the problem, then the number of NLP decision variables will be $n_s \times (N_k + 1) + (n_c \times N_k)$. This number is relatively large compared with applying multiple shooting method $n_c \times N_k$. When an NLP problem contains too many optimization parameters and equality constraints, it tends to consume the evolutionary solver a large amount of iteration times to capture the true behaviour or even fail to satisfy all of the constraints. Therefore, it is recommended to use the stochastic-based solvers with the multiple shooting discretization scheme, which can eventually reduce the number

of equality constraints and achieve enhanced convergence performance.

VII. CONCLUSION

In this study, a multiple shooting-based NSGA-III algorithm was designed to address the multi-objective spacecraft trajectory optimization problem. Several constraint handling algorithms were embedded in the proposed framework to deal with various mission constraints. Comparative simulations with other typical evolutionary multi-objective solvers were carried out and the results demonstrate that the proposed approach can generate more well-distributed Pareto fronts and has a better convergence performance than the other algorithms.

An important concern coming from the implementation of this approach is the threats of validity. For example, the randomness caused by stochastic process with respect to control variables can be regarded as one possible internal threat of validity. To address this issue, the lag equation is used to smooth the control profile. This mechanism allows users to apply a large number of temporal nodes, thereby decreasing the error order of the multiple shooting scheme and restricting the control rates. In addition, the generalizability of the result can also be considered as an external threat. By testing the proposed solver for various challenging mission scenarios, all the obtained results are still promising. Therefore, it can be concluded that the extended NSGA-III method can be an effective and reliable alternative to produce optimal control profiles of the multi-objective spacecraft entry trajectory planning problem.

REFERENCES

- [1] P. Narayan, P. Meyer, and D. Campbell, "Embedding human expert cognition into autonomous UAS trajectory planning," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 530–543, 2013.
- [2] A. Rucco, G. Notarstefano, and J. Hauser, "An efficient minimum-time trajectory generation strategy for two-track car vehicles," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1505–1519, 2015.
- [3] Z. Chen and H. T. Zhang, "A minimal control multiagent for collision avoidance and velocity alignment," *IEEE Transactions on Cybernetics*, vol. 47, no. 8, pp. 2185–2192, 2017.
- [4] A. J. Hausler, A. Saccon, A. P. Aguiar, J. Hauser, and A. M. Pascoal, "Energy-optimal motion planning for multiple robotic vehicles with collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 867–883, 2016.
- [5] L. Paull, C. Thibault, A. Nagaty, M. Seto, and H. Li, "Sensor-driven area coverage for an autonomous fixed-wing unmanned aerial vehicle," *IEEE Transactions on Cybernetics*, vol. 44, no. 9, pp. 1605–1618, 2014.
- [6] H. Peng, B. Chen, and Z. Wu, "Multi-objective transfer to libration-point orbits via the mixed low-thrust and invariant-manifold approach," *Nonlinear Dynamics*, vol. 77, no. 1, pp. 321–338, 2014.
- [7] H. Peng and W. Wang, "Adaptive surrogate model based multi-objective transfer trajectory optimization between different libration points," *Advances in Space Research*, vol. 58, no. 7, pp. 1331–1347, 2016.
- [8] W. Wang and H. Peng, "A fast multi-objective optimization design method for emergency libration point orbits transfer between the sun-earth and the earth-moon systems," *Aerospace Science and Technology*, vol. 63, pp. 152–166, 2017.
- [9] C. Yang, Z. Li, and J. Li, "Trajectory planning and optimized adaptive control for a class of wheeled inverted pendulum vehicle models," *IEEE Transactions on Cybernetics*, vol. 43, no. 1, pp. 24–36, 2013.
- [10] T. Albin, D. Ritter, N. Liberda, R. Quirynen, and M. Diehl, "In-vehicle realization of nonlinear mpc for gasoline two-stage turbocharging airpath control," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1606–1618, 2018.
- [11] H. Duan and S. Li, "Artificial bee colony based direct collocation for reentry trajectory optimization of hypersonic vehicle," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 615–626, 2015.
- [12] R. Genest and J. V. Ringwood, "Receding horizon pseudospectral control for energy maximization with application to wave energy devices," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 29–38, 2017.
- [13] D. A. Benson, G. T. Huntington, T. P. Thorvaldsen, and A. V. Rao, "Direct trajectory optimization and costate estimation via an orthogonal collocation method," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, pp. 1435–1440, 2006.
- [14] Y. J. Gong, J. J. Li, Y. Zhou, Y. Li, H. S. H. Chung, Y. H. Shi, and J. Zhang, "Genetic learning particle swarm optimization," *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2277–2290, 2016.
- [15] N. Yokoyama and S. Suzuki, "Modified genetic algorithm for constrained trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 1, pp. 139–144, 2005.
- [16] K. Subbarao and B. M. Shippey, "Hybrid genetic algorithm collocation method for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 4, pp. 1396–1403, 2009.
- [17] J. J. Kim and J. J. Lee, "Trajectory optimization with particle swarm optimization for manipulator motion planning," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 620–631, 2015.
- [18] B. A. Conway, "A survey of methods available for the numerical optimization of continuous dynamic systems," *Journal of Optimization Theory and Applications*, vol. 152, no. 2, pp. 271–306, 2012.
- [19] R. Chai, A. Savvaris, and A. Tsourdos, "Violation learning differential evolution-based hp-adaptive pseudospectral method for trajectory optimization of space maneuver vehicle," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 4, pp. 2031–2044, 2017.
- [20] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "Improved gradient-based algorithm for solving aeroassisted vehicle trajectory optimization problems," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 8, pp. 2093–2101, 2017.
- [21] S. Jiang and S. Yang, "Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons," *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 198–211, 2017.
- [22] P. C. Roy, M. M. Islam, K. Murase, and X. Yao, "Evolutionary path control strategy for solving many-objective optimization problem," *IEEE Transactions on Cybernetics*, vol. 45, no. 4, pp. 702–715, 2015.
- [23] B. Ji, X. Yuan, and Y. Yuan, "Modified NSGA-II for solving continuous berth allocation problem: Using multiobjective constraint-handling strategy," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2885–2895, 2017.
- [24] M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, and Y. Ong, "A new decomposition-based nsga-ii for many-objective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 7, pp. 1191–1210, 2018.
- [25] W. Hu, G. G. Yen, and G. Luo, "Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system," *IEEE Transactions on Cybernetics*, vol. 47, no. 6, pp. 1446–1459, 2017.
- [26] N. M. Hamza, D. L. Essam, and R. A. Sarker, "Constraint consensus mutation-based differential evolution for constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 447–459, 2016.
- [27] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "Trajectory optimization of space maneuver vehicle using a hybrid optimal control solver," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, pp. 1–14, 2017.
- [28] T. R. Jorris and R. G. Cobb, "Three-dimensional trajectory optimization satisfying waypoint and no-fly zone constraints," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 2, pp. 551–572, 2009.
- [29] R. Chai, A. Savvaris, and A. Tsourdos, "Fuzzy physical programming for space manoeuvre vehicles trajectory optimization based on hp-adaptive pseudospectral method," *Acta Astronautica*, vol. 123, pp. 62–70, 2016.
- [30] R. Chai, A. Savvaris, A. Tsourdos, and Y. Xia, "An interactive fuzzy physical programming for skip entry problem," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 5, pp. 2385–2398, 2017.
- [31] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [32] R. Chai, A. Savvaris, A. Tsourdos, and S. Chai, *Solving Multi-objective Aeroassisted Spacecraft Trajectory Optimization Problems Using Extended NSGA-II*, ser. AIAA SPACE Forum. American Institute of Aeronautics and Astronautics, 2017.

- [33] —, “Multi-objective trajectory optimization of space manoeuvre vehicle using adaptive differential evolution and modified game theory,” *Acta Astronautica*, vol. 136, pp. 273–280, 2017.
- [34] S. Kukkonen and K. Deb, “Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems,” in *2006 IEEE International Conference on Evolutionary Computation*, 2006, Conference Proceedings, pp. 1179–1186.
- [35] W. W. Hager, “Runge-kutta methods in optimal control and the transformed adjoint system,” *Numerische Mathematik*, vol. 87, no. 2, pp. 247–282, 2000.
- [36] Q. Zhang and H. Li, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [37] M. Elarbi, S. Bechikh, and L. B. Said, “On the importance of isolated solutions in constrained decomposition-based many-objective optimization,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, New York, USA, 2017, Conference Proceedings, pp. 561–568.
- [38] H. Jain and K. Deb, “An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.
- [39] 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.
- [40] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. d. Fonseca, “Performance assessment of multiobjective optimizers: an analysis and review,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.
- [41] B. Xue, M. Zhang, and W. N. Browne, “Particle swarm optimization for feature selection in classification: A multi-objective approach,” *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1656–1671, 2013.



Antonios Tsourdos (M'99) obtained an M.Eng. on electronic, control and systems engineering from the University of Sheffield, in 1995, an M.Sc. on systems engineering from Cardiff University, in 1996, and a Ph.D. on nonlinear robust missile autopilot design and analysis from Cranfield University, in 1999. His research interests include UAV guidance and control, trajectory optimization, and advanced control systems.



Senchun Chai received his Ph.D. degree and then held a Postdoc fellowship with School of Electronics, University of Glamorgan, UK, in 2007 and 2009, respectively. His current research interests include design of unmanned aerial vehicles (UAV), wireless sensor network control, networked control systems and multi-agent control systems.



Yuanqing Xia (M'15-SM'16) received the B.S. degree from the Department of Mathematics, Chuzhou University, Chuzhou, China, in 1991, the M.S. degree in fundamental mathematics from Anhui University, Wuhu, China, in 1998, and the Ph.D. degree in control theory and control engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2001. His current research interests are in the fields of networked control systems, robust control and signal processing.



Runqi Chai (SM'15-M'18) received the B.S. degree in information and computing science from the North China University of Technology, Beijing, China, in 2015 and the Ph.D. degree in Aerospace Engineering from Cranfield University, Cranfield, U.K, in 2018. His research interests include trajectory optimization, networked control systems, and multi-agent control systems.



Al Savvaris received the M.Eng. degree in aerospace systems engineering from the University of Hertfordshire, Hertfordshire, U.K., in 1998 and the Ph.D. degree in radiowave propagation and system design from the University of South Wales, Pontypridd, U.K., in 2004. His research interests include trajectory optimization, networked control systems, and aircraft control systems.

2018-11-22

Solving multiobjective constrained trajectory optimization problem by an extended evolutionary algorithm

Chai, Runqi

IEEE

Chai R, Savvaris A, Tsourdos A, et al., (2020) Solving multiobjective constrained trajectory optimization problem by an extended evolutionary algorithm. IEEE Transactions on Cybernetics, Volume 50, Issue 4, April 2020, pp. 1630-1643

<https://doi.org/10.1109/TCYB.2018.2881190>

Downloaded from Cranfield Library Services E-Repository