Journal of Systems Engineering and Electronics
Vol. 28, No. 4, August 2017, pp.732 – 744



# Multi-objective reentry trajectory optimization method via GVD for hypersonic vehicles

Chaofang Hu*, Yue Xin, and Hao Feng

School of Electrical Engineering and Automation, Tianjin University, Tianjin 300072, China

**Abstract:** In the constrained reentry trajectory design of hypersonic vehicles, multiple objectives with priorities bring about more difficulties to find the optimal solution. Therefore, a multi-objective reentry trajectory optimization (MORTO) approach via generalized varying domain (GVD) is proposed. Using the direct collocation approach, the trajectory optimization problem involving multiple objectives is discretized into a nonlinear multi-objective programming with priorities. In terms of fuzzy sets, the objectives are fuzzified into three types of fuzzy goals, and their constant tolerances are substituted by the varying domains. According to the principle that the objective with higher priority has higher satisfactory degree, the priority requirement is modeled as the order constraints of the varying domains. The corresponding two-side, single-side, and hybrid-side varying domain models are formulated for three fuzzy relations respectively. By regulating the parameter, the optimal reentry trajectory satisfying priorities can be achieved. Moreover, the performance about the parameter is analyzed, and the algorithm to find its specific value for maximum priority difference is proposed. The simulations demonstrate the effectiveness of the proposed method for hypersonic vehicles, and the comparisons with the traditional methods and sensitivity analysis are presented.

**Keywords:** hypersonic vehicle, reentry trajectory design, multi-objective optimization, generalized varying domain, direct collocation method.

**DOI:** 10.21629/JSEE.2017.04.13

## 1. Introduction

In recent years, many efforts have been devoted to the reentry trajectory design for hypersonic vehicles. Various indirect and direct methods were developed to tackle the reentry trajectory optimization problem [1]. In the early times, the indirect method was applied widely by transforming the optimal control problem into the Hamiltonian boundary value problem [2]. However, its computation was very complicated, and its solution might be unable to converge. Consequently, many direct methods (numerical approaches) attracted more and more attention, e.g. the direct shooting, direct collocation and pseudospectral methods. In the direct shooting approach, only control variables were discretized, and dynamic equations were approximated via numerical integration [3]. In the direct collocation method, both state variables and control variables were discretized, and dynamic equations were approximated by Gauss-Lobatto polynomials [4]. The pseudospectral method was also called the orthogonal collocation method [5], where the state and control variables were approximated by global polynomials, and the difference between the dynamic equation and the derivative of polynomial was required to be zero. In addition, some other optimization methods were also introduced to the reentry trajectory design, such as rapid-exploring random tree algorithm (applied in robot trajectory planning) [6] and receding horizon optimization method (derived from optimal control) [7].

Nowadays, the ability to improve performance and reduce costs in the reentry trajectory optimization simultaneously becomes more and more important. Therefore, the conventional single objective trajectory optimization method is not appropriate. This boosts the development of the multi-objective reentry trajectory optimization (MORTO). For example, the traditional weighted method is used for hypersonic vehicles [8]. Nevertheless, it is known that determining the accurate weights is very hard. Following the intelligent multi-objective optimization (MOO) methods [9], Chen et al. utilized the NSGA-II algorithm to design the multi-objective reentry trajectory of the reusable launch vehicle [10]. Mooij and Hänninen designed a distributed global optimization framework via the particle swarm optimization algorithm to find the Pareto front of trajectory for the moderate lift-to-drag reentry vehicle [11]. In addition, Yong et al. used

Manuscript received November 06, 2015.
*Corresponding author.
This work was supported by the Natural Science Foundation of Tianjin (12JCZDJC30300), the Research Foundation of Tianjin Key Laboratory of Process Measurement and Control (TKLPMC-201613), and the State Scholarship Fund of China.

the physical programming method to deal with MORTO for the gliding vehicle [12]. Although the optimal results are obtained by the above methods, the importance or priority requirements of each objective cannot be guaranteed. Moreover, the computational burden of the iterative optimization is heavy, and the reformulation is strongly nonlinear.

As a special case of importance preference, priority has a stricter order requirement, where all the objectives must be optimized in sequence. Although there are many works on MOO with priorities, such as lexicographic optimization [13,14], two-step satisfactory optimization [15], and fuzzy satisfactory goal programming [16], how to find the preferred solution is still a difficult problem. Especially when MOO is involved in the reentry trajectory design, the optimization problem will become more complex. The original reentry trajectory optimization is continuous, and then it is equivalent to a large infinite-dimensional optimization problem. Thereby, the above approaches might be unsuitable since their effectiveness is only verified by some small-scale optimization problems.

In this paper, a MORTO method via the generalized varying domain (GVD) [17,18] is proposed for the hypersonic vehicle. By the direct collocation method, the trajectory optimization problem is discretized into a nonlinear multi-objective programming problem with priorities. The attack angles and bank angles at time nodes and collocation points are taken as control variables. In order to acquire the satisfactory reentry trajectory, the concept of fuzzy set is introduced to convert the objectives into fuzzy goals. The principle that the objective with higher priority has higher satisfactory degree is followed [14], and then priority requirement is modeled into the order of the satisfactory degrees. To guarantee the feasibility of the fuzzy multi-objective programming with the strict order constraints, the GVD strategy is introduced and enhanced. The constant domain of each objective is converted into a varying domain so that the order constraints of the varying domains are formulated to relax the strict order constraints of the satisfactory degrees. Correspondingly, three varying domain models including two-side, single-side, and hybrid-side formulations are presented for three fuzzy relations respectively. By regulating the parameter $\lambda$, we can obtain the optimal reentry trajectory satisfying the requirements of optimization and priority of all the objectives. Likewise, the effect of this parameter on the optimization result is analyzed. The algorithm to seek the parameter $\lambda^*$ for maximum difference of priority is proposed, where the solution will keep identical when $\lambda \geqslant \lambda^*$.

Section 2 describes the reentry trajectory design problem of hypersonic vehicles. The MORTO method via GVD is proposed in Section 3. Section 4 demonstrates its power by simulation. In Section 5 the conclusions are drawn.

## 2. Problem formulation

### 2.1 Reentry dynamic equations of hypersonic vehicles

Assumptions are given as follows.

(i) The Earth is uniform sphere and flat.

(ii) The Earth's revolution is ignored.

(iii) The vehicle is a point.

(iv) The sideslip angle is zero.

The dynamic equations are presented [19] as follows:

$$\dot{h} = v \sin\gamma \tag{1}$$

$$\dot{\phi} = \frac{v\cos\gamma\sin\chi}{(R_0+h)\cos\theta} \tag{2}$$

$$\dot{\theta} = \frac{v\cos\gamma\cos\chi}{(R_0+h)} \tag{3}$$

$$\dot{v} = -\frac{D}{m} - g\sin\gamma \tag{4}$$

$$\dot{\chi} = \frac{L\sin\sigma}{mv\cos\gamma} + \frac{v}{(R_0+h)}\cos\gamma\sin\chi\tan\theta \tag{5}$$

$$\dot{\gamma} = \frac{L\cos\sigma}{mv} - \left(\frac{g}{v} - \frac{v}{(R_0+h)}\right)\cos\gamma \tag{6}$$

where $h, \phi, \theta, v, \chi$ and $\gamma$ are state variables including altitude, longitude, latitude, velocity, heading angle and flight-path angle. $\alpha$ and $\sigma$ are attack angle and bank angle, taken as control variables. $R_0$ is the radius of the Earth, $m$ is the mass of the vehicle, and $g = \mu/(R_0+h)^2$ is the gravity with $\mu$ being the gravitational parameter of the Earth.

$L$ and $D$ are the aerodynamic forces, i.e. lift and drag, defined as

$$L = \rho v^2 C_L S/2, \quad D = \rho v^2 C_D S/2. \tag{7}$$

$S$ is the reference area of the vehicle, and $\rho$ is the atmospheric density which is presented as the exponential atmosphere model

$$\rho = \rho_0 \mathrm{e}^{-kh} \tag{8}$$

where $\rho_0$ is sea level atmospheric density.

$C_L, C_D$ are lift and drag coefficients

$$C_L = C_{L0} + C_{L1}\alpha \tag{9}$$

$$C_D = C_{D0} + C_{D1}\alpha + C_{D2}\alpha^2 \tag{10}$$

where $C_{L0}$ and $C_{L1}$ are lift increment coefficients for the vehicle and $C_{D0}$, $C_{D1}$, and $C_{D2}$ are drag increment coefficients.

### 2.2 Objectives and priorities

To improve the reentry performance of hypersonic vehicles, multiple objectives are considered in this paper.

Firstly, the maximum cross range, i.e. the maximum terminal latitude, is an important indicator for the ability of the reentry flight. Meanwhile, a larger terminal velocity is also indispensable to obtain more energy for some special missions. Therefore, they are the most important tasks for reentry trajectory optimization of the hypersonic vehicle, and must be optimized at first. Secondly, reducing aerodynamic heating should be considered to prevent the harm to the structure of the vehicle. Finally, a severe oscillation is not desired since it might lead to the unstable trajectory. These two objectives should be optimized as much as possible after the maximum terminal latitude and terminal velocity are realized.

Hence, the four corresponding objectives are presented as follows.

(i) The maximum terminal latitude

$$\max \quad f_1 = \theta(t_f) \tag{11}$$

(ii) The maximum terminal velocity

$$\max \quad f_2 = v(t_f) \tag{12}$$

(iii) The minimum total aerodynamic heating

$$\min \quad f_3 = \int_{t_0}^{t_f} \dot{Q}(t)\mathrm{d}t \tag{13}$$

(iv) The minimum oscillation

$$\min \quad f_4 = \int_{t_0}^{t_f} \dot{\gamma}^2(t)\mathrm{d}t \tag{14}$$

where $\dot{Q}$ means the heat flux.

According to their importance, objectives (i) and (ii) have the highest priority, and then (iii) and (iv) will be optimized successively. Consequently, they are assigned the following priorities:

Priority one: objective (i) and objective (ii);

Priority two: objective (iii);

Priority three: objective (iv).

### 2.3 Constraints

Firstly, in order to ensure the normal operation of the thermal protection system of hypersonic vehicles and save the cost of heat-resistant materials, the heat flux constraint is given as

$$\dot{Q} = C\rho^{0.5}v^{3.07}(h_a + h_b\alpha + h_c\alpha^2 + h_d\alpha^3) \leqslant \dot{Q}_{\max} \tag{15}$$

where $\dot{Q}_{\max}$ is the maximum heat flux, and $C$, $h_a$, $h_b$, $h_c$, $h_d$ are vehicle-dependent constant coefficients.

Secondly, the following dynamic pressure constraint should be considered, which contributes to structure protection and attitude control

$$q = \rho v^2/2 \leqslant q_{\max} \tag{16}$$

where $q$ denotes dynamic pressure, and $q_{\max}$ is the maximum dynamic pressure.

Then, the overload $n$ associated with flight safety is subjected to

$$n = \sqrt{L^2 + D^2}/(mg) \leqslant n_{\max} \tag{17}$$

where $n_{\max}$ is the maximum overload.

In addition, the initial and terminal conditions of states are described as $[h_0, \phi_0, \theta_0, v_0, \chi_0, \gamma_0]$ and $[h_f, v_f, \gamma_f]$.

Finally, the following inequalities are the boundary constraints of the state and control variables

$$\begin{aligned} &h_{\min} \leqslant h \leqslant h_{\max}, \quad \phi_{\min} \leqslant \phi \leqslant \phi_{\max} \\ &\theta_{\min} \leqslant \theta \leqslant \theta_{\max}, \quad v_{\min} \leqslant v \leqslant v_{\max} \\ &\chi_{\min} \leqslant \chi \leqslant \chi_{\max}, \quad \gamma_{\min} \leqslant \gamma \leqslant \gamma_{\max} \\ &\alpha_{\min} \leqslant \alpha \leqslant \alpha_{\max}, \quad \sigma_{\min} \leqslant \sigma \leqslant \sigma_{\max}. \end{aligned} \tag{18}$$

The subscripts “min” and “max” represent the lower and the upper boundaries of the variables respectively.

## 3. Optimization method

During the reentry trajectory optimization, (1) – (10) are also reckoned the state constraints. However, it is very hard to find the optimal solution using general nonlinear programming directly because (1) – (6) are differential formulations. In this paper, a direct collocation method is utilized to transform them into algebraic equations, and then the continuous reentry trajectory optimal control problem is discretized into the nonlinear multi-objective programming problem with priorities.

In order to satisfy the priority requirement, the objectives are fuzzified into fuzzy goals, and the concept of varying domain is introduced to substitute the satisfactory degree of each objective to reformulate the priority order. The different varying domain models with respect to different fuzzy relations are proposed to realize tradeoff between optimization and priority.

The state variable vector and control variable vector are defined as $\boldsymbol{x} = (h, \phi, \theta, v, \chi, \gamma) \in \mathbf{R}^6$ and $\boldsymbol{u} = (\alpha, \sigma) \in \mathbf{R}^2$, respectively.

### 3.1 Direct collocation method

In the direct collocation method [4,20], the state and control variables are discretized at time nodes, and the state variables between nodes are approximated by polynomials with Gauss-Lobatto rules [21]. In this paper, the third order Simpson method [22] is used.

Firstly, suppose the whole reentry time period of the hypersonic vehicle is $[t_0, t_f]$. It is divided into $N$ segments. Each segment is written as $[t_{i-1}, t_i]$ $(i = 1, 2, \ldots, N)$, and its length is defined as $r_i = t_i - t_{i-1}$. Taking two boundary points of each segment as nodes, we will get $N+1$ nodes. Then, each state variable and control variable at each time node $i$ are written as $x_m^i$ $(m = 1, 2, \ldots, 6)$ and $u_n^i$ $(n = 1, 2)$.

Secondly, the third order Hermite polynomial is used to approximate state variable $x_m^{si}$ in $[t_{i-1}, t_i]$. There are

$$\begin{aligned} x_m^{si} &= c_0^{si} + c_1^{si}s + c_2^{si}s^2 + c_3^{si}s^3 \\ \dot{x}_m^{si} &= c_1^{si} + 2c_2^{si}s + 3c_3^{si}s^2 \end{aligned} \tag{19}$$

where $s = (t - t_{i-1})/r_i$, then $s \in [0, 1]$.

The boundary conditions are

$$\begin{aligned} x_m^{i-1} &= x_m^{si}\big|_{s=0}, \quad x_m^i = x_m^{si}\big|_{s=1} \\ \dot{x}_m^{i-1} &= \dot{x}_m^{si}\big|_{s=0}, \quad \dot{x}_m^i = \dot{x}_m^{si}\big|_{s=1}. \end{aligned} \tag{20}$$

Combining (19) with (20), we can obtain

$$\begin{aligned} c_0^{si} &= x_m^{i-1} \\ c_1^{si} &= \dot{x}_m^{i-1} \\ c_2^{si} &= -3x_m^{i-1} - 2\dot{x}_m^{i-1} + 3x_m^i - \dot{x}_m^i \\ c_3^{si} &= 2x_m^{i-1} + \dot{x}_m^{i-1} - 2x_m^i + \dot{x}_m^i. \end{aligned} \tag{21}$$

The midpoint of each segment, i.e. $s = 1/2$, is taken as the collocation point. Substituting (21) into (19), the state variables and their derivatives at the collocation point can be presented as

$$\begin{aligned} x_m^{ci} &= \frac{x_m^i + x_m^{i-1}}{2} + \frac{r_i(d_m^{i-1} - d_m^i)}{8} \\ \dot{x}_m^{ci} &= \frac{3(x_m^i - x_m^{i-1})}{2r_i} - \frac{(d_m^{i-1} + d_m^i)}{4} \end{aligned} \tag{22}$$

where $d_m^i$ is the derivative of the state variable at the time node $i$, calculated by dynamic equations. The superscript "$c$" means the collocation point.

Equations (1)–(6) are integrated by the third order Simpson formula at each interval, so it is concluded that

$$x_m^i = x_m^{i-1} + \frac{h_i}{6}(d_m^{i-1} + 4d_m^{ci} + d_m^i). \tag{23}$$

Hence, we can define the Hermite-Simpson defect vector $\boldsymbol{D}$ as

$$\boldsymbol{D} = x_m^{i-1} - x_m^i + \frac{h_i}{6}(d_m^{i-1} + 4d_m^{ci} + d_m^i) \tag{24}$$

where $d^{ci}$ is derived from dynamic equations at the collocation point.

In order to make (19) approximate real state variables as much as possible, $\boldsymbol{D}$ should be zero such that $\dot{x}_m^{ci}$ at the collocation point computed by (22) will equal the values obtained by (1)–(6). Consequently, the dynamic differential equations can be converted into algebraic constraints.

For the objectives and their priority requirements in Section 2.2, the following nonlinear multi-objective programming model with priorities is formulated by combining all the algebraic constraints

$$\begin{aligned} &\begin{bmatrix} P_1(\max f_1(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c), \max f_2(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c)), \\ P_2(\min f_3(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c)), \\ P_3(\min f_4(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c)) \end{bmatrix} \\ &\text{s.t.} \quad (\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c) \in H(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c) \cap G(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c). \end{aligned} \tag{25}$$

In (25), optimization variables include the state variables $\boldsymbol{X} = (\boldsymbol{x}^0, \ldots, \boldsymbol{x}^N)$ and control variables $\boldsymbol{U} = (\boldsymbol{u}^0, \ldots, \boldsymbol{u}^N)$ at nodes, and control variables $\boldsymbol{U}^c = (\boldsymbol{u}^{c1}, \ldots, \boldsymbol{u}^{cN})$ at collocation points. $H(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c)$ represents the algebraic constraints about dynamic equations, and $G(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c)$ is the intersection of constraints (15)–(18). $P_i$ $(i = 1, 2, 3)$ is the priority factors of the objectives, and $P_i \ll P_{i+1}$ denotes that the objectives in $P_i$ have a higher priority than those in $P_{i+1}$.

### 3.2 Fuzzy goals

In order to obtain the satisfactory reentry trajectory, fuzzy sets are introduced to deal with the objectives (11)–(14). Generally, the MOO problem with fuzzy goals is written as [13,23]

$$\begin{aligned} &\text{Find} \quad \boldsymbol{z} \\ &\text{s.t.} \quad f_j(\boldsymbol{z}) \to f_j^*, \quad j = 1, \ldots, M \\ &\qquad \boldsymbol{z} \in G(\boldsymbol{z}) \end{aligned} \tag{26}$$

where $\boldsymbol{z}$ represents the decision variable vector, $f_j(\boldsymbol{z})$ $(j = 1, \ldots, M)$ is the objective function, and $f_j^*$ is its goal value. "$\to$" denotes different fuzzy relations, including "$\tilde{\leqslant}$" (fuzzy minimization), "$\tilde{\geqslant}$" (fuzzy maximization) and "$\cong$" (fuzzy equality).

Based on the concept of fuzzy sets, we define different membership functions $\mu_{f_j}(\boldsymbol{z})$ $(j = 1, \ldots, M)$ to model three fuzzy relations. They respectively represent that the objective is approximately less than or equal to, more than or equal to, or in the vicinity of its goal value. The value of the membership function is called the satisfactory degree. The linear triangle membership functions are used.

Fuzzy minimization "$\tilde{\leqslant}$" means the objective is approximately less than or equal to its goal value, and then there is

$$\mu_{f_j}(\boldsymbol{z}) = \begin{cases} 1, & f_j(\boldsymbol{z}) \leqslant f_j^* \\ \dfrac{f_j^{\max} - f_j(\boldsymbol{z})}{f_j^{\max} - f_j^*}, & f_j^* \leqslant f_j(\boldsymbol{z}) \leqslant f_j^{\max} \\ 0, & f_j(\boldsymbol{z}) \geqslant f_j^{\max} \end{cases}. \tag{27a}$$

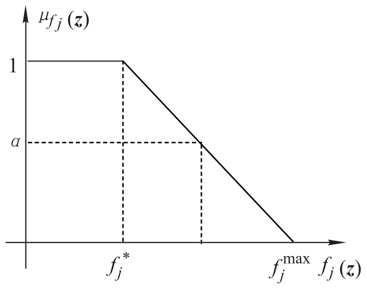
For the fuzzy maximization "$\tilde{\geqslant}$", the objective is required to be approximately more than or equal to its prospective value. The membership function is defined as

$$\mu_{f_j}(\boldsymbol{z}) = \begin{cases} 1, \quad f_j(\boldsymbol{z}) \geqslant f_j^* \\ \dfrac{f_j(\boldsymbol{z}) - f_j^{\min}}{f_j^* - f_j^{\min}}, \quad f_j^{\min} \leqslant f_j(\boldsymbol{z}) \leqslant f_j^* \\ 0, \quad f_j(\boldsymbol{z}) \leqslant f_j^{\min} \end{cases}. \tag{27b}$$

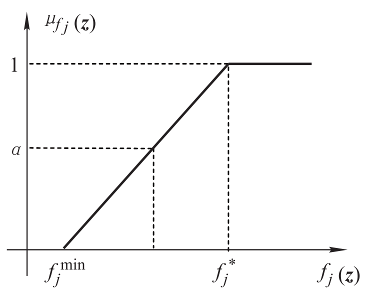
The fuzzy equality "$\tilde{=}$" demands that the objective is in the vicinity of its target. Its membership function formulation is

$$\mu_{f_j}(\boldsymbol{z}) = \begin{cases} 0, \quad f_j(\boldsymbol{z}) \geqslant f_j^{\max} \\ \dfrac{f_j^{\max} - f_j(\boldsymbol{z})}{f_j^{\max} - f_j^*}, \quad f_j^* \leqslant f_j(\boldsymbol{z}) \leqslant f_j^{\max} \\ 1, \quad f_j(\boldsymbol{z}) = f_j^* \\ \dfrac{f_j(\boldsymbol{z}) - f_j^{\min}}{f_j^* - f_j^{\min}}, \quad f_j^{\min} \leqslant f_j(\boldsymbol{z}) \leqslant f_j^* \\ 0, \quad f_j(\boldsymbol{z}) \leqslant f_j^{\min} \end{cases}. \tag{27c}$$

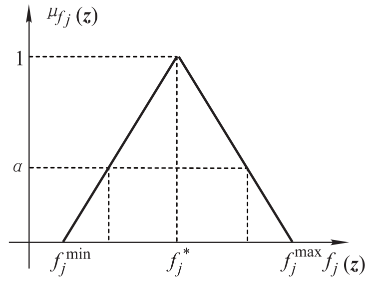
Fig. 1 shows the graphs of three membership functions.




(a) Fuzzy minimization "$\tilde{\leqslant}$"




(b) Fuzzy maximization "$\tilde{\geqslant}$"




(c) Fuzzy equality "$\tilde{=}$"

**Fig. 1 Membership functions for three fuzzy relations**

In (27a)–(27b), $f_j^{\min}$ and $f_j^{\max}$ are the tolerant limits of $f_j(\boldsymbol{z})$. They can be determined via the payoff table (Table 1) which is constructed using the optimal solution of single objective optimization. The fuzzy minimization "$\tilde{\leqslant}$" is taken as an example. $\boldsymbol{z}_j^*$ is the optimal solution of the $j$th objective, then the tolerant limit $f_k^{\max}$ of the $k\mathrm{th}$ objective can be achieved by the means of $f_k^{\max} = \max\limits_{j=1,\dots,M} f_k(\boldsymbol{z}_j^*)$.

**Table 1 Payoff table**

| Single objective function | $f_1$ | $\cdots$ | $f_M$ |
|---|---|---|---|
| $\min\ f_1(\boldsymbol{z})$ | $f_1(\boldsymbol{z}_1^*)$ | $\cdots$ | $f_M(\boldsymbol{z}_1^*)$ |
| $\min\ f_2(\boldsymbol{z})$ | $f_1(\boldsymbol{z}_2^*)$ | $\cdots$ | $f_M(\boldsymbol{z}_2^*)$ |
| $\min\ f_3(\boldsymbol{z})$ | $f_1(\boldsymbol{z}_3^*)$ | $\cdots$ | $f_M(\boldsymbol{z}_3^*)$ |
| $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ |
| $\min\ f_M(\boldsymbol{z})$ | $f_1(\boldsymbol{z}_M^*)$ | $\cdots$ | $f_M(\boldsymbol{z}_M^*)$ |

### 3.3 GVD

For the priority requirement, the lexicographic method is the most conventional strategy, where the objectives are solved in the lexicographic order [24]. Nevertheless, this approach might cause heavy computational burden or degenerative optimization.

Therefore, the principle that the objective with a higher priority will have a higher satisfactory degree is considered [14]. Suppose that the priority of the objective $f_s(\boldsymbol{z})$ is higher than that of $f_t(\boldsymbol{z})$, $s, t \in \{1, \dots, M\}$, $s \neq t$. Then the priority can be modeled into the following order of satisfactory degrees

$$\mu_{f_t}(\boldsymbol{z}) \leqslant \mu_{f_s}(\boldsymbol{z}). \tag{28}$$

This inequality is introduced into the MOO model (26) as the priority constraint.

Generally, there are two classical decision making formulations in MOO [14], i.e. min-max criterion and additive criterion. Based on min-max criterion, we have

$$\max \quad \delta$$

$$\text{s.t.} \quad \mu_{f_j}(\boldsymbol{z}) \geqslant \delta, \quad j = 1, 2, \dots, M$$

$$\mu_{f_t}(\boldsymbol{z}) \leqslant \mu_{f_s}(\boldsymbol{z}), \quad s,t \in \{1,2,\ldots,M\}, \quad s \neq t$$
$$\boldsymbol{z} \in G(\boldsymbol{z}). \tag{29}$$

In terms of the additive criterion, there is

$$\begin{aligned} &\max \ \sum_{j=1}^{M} \mu_{f_j}(\boldsymbol{z}) \\ &\text{s.t.} \quad \mu_{f_t}(\boldsymbol{z}) \leqslant \mu_{f_s}(\boldsymbol{z}), \quad s,t \in \{1,2,\ldots,M\}, \quad s \neq t \\ &\qquad \boldsymbol{z} \in G(\boldsymbol{z}). \end{aligned} \tag{30}$$

Although (28) can represent the priority order, it is difficult to quantify the difference between the priorities. Furthermore, (28) is too strict to ensure the feasibility of (29) or (30). Hence, the priority variable $\eta$ is proposed to relax (28)

$$\mu_{f_t}(\boldsymbol{z}) - \mu_{f_s}(\boldsymbol{z}) \leqslant \eta. \tag{31}$$

When $\eta \leqslant 0$, (28) must hold, and the priority difference will be enlarged with the decrease of $\eta$.

In this paper, the decision criteria of (29) and (30) are synthesized to optimize all the objectives as much as possible.

As we know, the GVD method [17,18] is an effective strategy to deal with the priority requirement. Wherein, the constant tolerances of the objectives are substituted by the domain variable to realize the tradeoff between optimization and priority. Thereby, this strategy is introduced. However, in the original GVD method, the objective with the lowest priority cannot be guaranteed to have the inferior satisfactory degree. This will result in the decrease of the priority difference such that the solution is not the most satisfactory. In addition, the GVD does not consider the effect of the parameter on the optimization solution. Generally, the best parameter is obtained by trail and error. During regulation, the parameter will possibly be assigned a big value, which leads to more iterations. Thereby, the GVD method is improved in this paper. The enhanced GVD can regulate the priority difference fully, and ensure to find the best solution. Furthermore, the specific value of the parameter for maximum priority difference can be acquired, which is helpful to determine the most approximate parameter in few iterations.

### 3.4 Optimization formulations based on GVD

#### 3.4.1 Two-side varying domain optimization

In this case, only the fuzzy equality "$\cong$" is considered. Each objective is normalized as

$$f'_j(\boldsymbol{z}) = \frac{2[f_j(\boldsymbol{z}) - f_j^*]}{f_j^{\max} - f_j^{\min}}, \quad j = 1,2,\ldots,M. \tag{32}$$

Consequently, the normalized objective will have the same goal value "0" and the same symmetrical domain $[-1,1]$. From (27c), it is known that the normalized objective $f'_j(\boldsymbol{z})$ will also have the same satisfactory degree $\delta$ as the original $f_j(\boldsymbol{z})$. That is

$$\delta = \mu_{f_j}(\boldsymbol{z}) = \mu_{f'_j}(\boldsymbol{z}) = 1 - f'_j(\boldsymbol{z}). \tag{33}$$

Combining (31) and (33), the relaxed satisfactory degree order (31) will be equivalent to

$$f'_s(\boldsymbol{z}) - f'_t(\boldsymbol{z}) \leqslant \eta. \tag{34}$$

From (33), we know that the satisfactory degree of $f_j(\boldsymbol{z})$ will approximate "1" when $f'_j(\boldsymbol{z})$ approaches "0". If $f'_j(\boldsymbol{z})$ has a smaller domain than $[-1,1]$, then the possibility approaching "0" will increase. Therefore, the varying domain $[-\beta_j, \beta_j]$ $(0 \leqslant \beta_j \leqslant 1)$ is used to replace the constant domain $[-1,1]$. When the following inequality holds, $f'_j(\boldsymbol{z})$ can locate in the interval about $\delta$

$$\delta\beta_j - \beta_j \leqslant f'_j(\boldsymbol{z}) \leqslant \beta_j - \delta\beta_j. \tag{35}$$

To enhance the flexibility of the new priority order (34), the order of varying domains is proposed to replace (34). This means that it is easy to obtain a higher priority in the smaller varying domain. Consequently, (34) can be reformulated as

$$\beta_s - \beta_t \leqslant \eta \tag{36}$$

which will be taken as the new priority constraint of the MOO model.

According to references [17,18], we know that (36) is reasonable and effective. However, it cannot guarantee that the objective with the lowest priority must have the inferior satisfactory degree. Consequently, the priority difference is possibly decreased, and then to find the satisfactory solution becomes difficult. Hence, the varying domain strategy is enhanced in this paper. While the objective $f_L(\boldsymbol{z})$ is assigned the lowest priority, its domain should be set up as $[-1,1]$, i.e. $\beta_L = 1$. Furthermore, its satisfactory degree equation $\delta = 1 - f'_L(\boldsymbol{z})$ is added as the equality constraint.

Correspondingly, the two-side varying domain optimization model for fuzzy equality "$\cong$" is formulated in the following, and it is shown in Fig. 2.

$$\begin{gathered} \max \ (\delta - \lambda\eta) \\ \text{s.t.} \quad \delta\beta_j - \beta_j \leqslant f'_j(\boldsymbol{z}) \leqslant \beta_j - \delta\beta_j \\ \delta = 1 - f'_L(\boldsymbol{z}) \\ \beta_s - \beta_t \leqslant \eta, \quad s,t,L \in \{1,2,\ldots,M\}, \quad s \neq t \neq L \\ \beta_t - \beta_L \leqslant \eta \\ 0 \leqslant \beta_j \leqslant 1, \quad \beta_L = 1, \quad j = 1,2,\ldots,M, \quad j \neq L, \quad 0 \leqslant \delta \leqslant 1 \\ \boldsymbol{z} \in G(\boldsymbol{z}) \end{gathered} \tag{37}$$

where $\lambda$ is the compromise factor balancing optimization and priority requirement. Therefore, it is considered an optimization parameter in (37). When $\lambda \to 0$, all the objectives are optimized as much as possible, and the priorities are ignored. In contrast, we will get the solution satisfying priorities instead of considering optimization of the objectives while $\lambda \to \infty$. Therefore, the preferred solution is compromising, and it can be obtained by choosing the appropriate $\lambda$. Then, (37) is called parametric programming model.

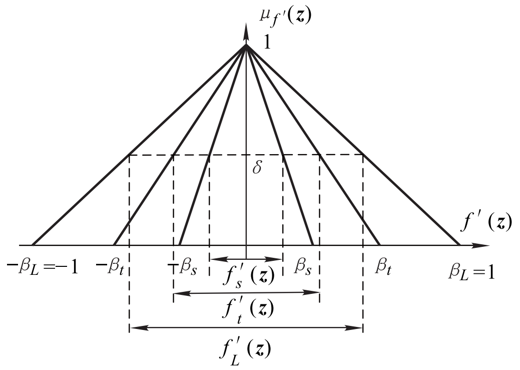



**Fig. 2 Two-side varying domain optimization model**

For (37), if $\eta \leqslant 0$, then $\beta_s \leqslant \beta_t$. This means that $f_s(\boldsymbol{z})$ has a higher priority than $f_t(\boldsymbol{z})$ in the probability perspective.

### 3.4.2 Single-side varying domain optimization

Only fuzzy inequalities are involved, i.e. fuzzy minimization "$\widetilde{\leqslant}$" and fuzzy maximization "$\widetilde{\geqslant}$". Wherein, "$\widetilde{\geqslant}$" case can be transformed into "$\widetilde{\leqslant}$" by minimizing $-f_j(\boldsymbol{z})$. Then, the objective function is normalized as $f'_j(\boldsymbol{z}) = [f_j(\boldsymbol{z}) - f_j^*]/[f_j^{\max} - f_j^*]$. Its varying domain is presented as $[0, \beta_j]$. Following (34)–(37), the single-side varying domain optimization model is formulated as

$$\begin{gathered}
\max\ (\delta - \lambda\eta) \\
\text{s.t.}\quad 0 \leqslant f'_j(\boldsymbol{z}) \leqslant \beta_j - \delta\beta_j \\
\delta = 1 - f'_L(\boldsymbol{z}) \\
\beta_s - \beta_t \leqslant \eta, \quad s,t,L \in \{1,2,\dots,M\}, \quad s \neq t \neq L \\
\beta_t - \beta_L \leqslant \eta \\
0 \leqslant \beta_j \leqslant 1, \quad \beta_L = 1, \quad j = 1,2,\dots,M, \quad j \neq L \\
0 \leqslant \delta \leqslant 1 \\
z \in G(\boldsymbol{z}). \qquad (38)
\end{gathered}$$

It is shown in Fig. 3.

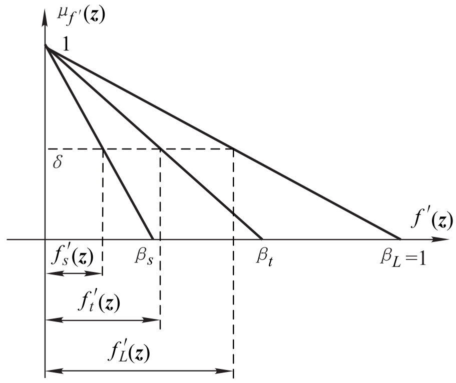



**Fig. 3 Single-side varying domain optimization model**

### 3.4.3 Hybrid-side varying domain optimization

In a hybrid case, fuzzy inequality and equality exist simultaneously. Similarly, the hybrid-side varying domain optimization model is constructed as

$$\begin{gathered}
\max\ (\delta - \lambda\eta) \\
\text{s.t.}\quad 0 \leqslant f'_k(\boldsymbol{z}) \leqslant \beta_k - \delta\beta_k, \quad k \in I_1 \\
\delta\beta_j - \beta_j \leqslant f'_j(\boldsymbol{z}) \leqslant \beta_j - \delta\beta_j, \quad j \in I_2 \\
\delta = 1 - f'_L(\boldsymbol{z}) \\
\beta_s - \beta_t \leqslant \eta, \quad s,t,L \in \{1,2,\dots,M\}, \quad s \neq t \neq L \\
\beta_t - \beta_L \leqslant \eta \\
0 \leqslant \beta_k \leqslant 1, \quad 0 \leqslant \beta_j \leqslant 1, \quad \beta_L = 1 \\
0 \leqslant \delta \leqslant 1 \\
\boldsymbol{z} \in G(\boldsymbol{z}) \qquad (39)
\end{gathered}$$

where $I_1$ is the set of the objectives with fuzzy inequalities, $I_2$ is the set for fuzzy equalities, and $I_1 \cup I_2 = \{1,2,\dots,M\}$. The hybrid-side case is presented in Fig. 4.

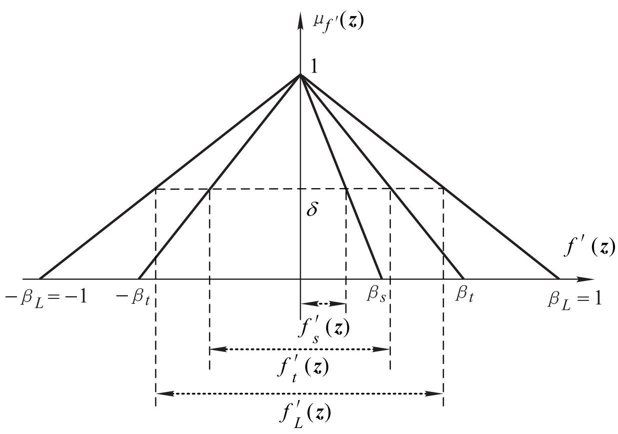



**Fig. 4 Hybrid-side varying domain optimization model**

### 3.5 Algorithm

The steps of our multi-objective optimization method via GVD is summarized as follows.

**Step 1** Calculate the tolerant limits of objectives under the given constraints, and normalize the objectives.

**Step 2** According to fuzzy relations, formulate the corresponding GVD optimization model as (37), (38) or (39), and set up an initial value to $\lambda$.

**Step 3** Solve the optimization formulation in Step 2, and obtain the solution.

**Step 4** When $\eta > 0$, the priority order is violated, then increase $\lambda$, and return to Step 3. When $\eta \leqslant 0$, if the solution is not satisfactory, then assign a larger value to $\lambda$, and go back to Step 3. Otherwise, the optimal solution is achieved, and the optimization stops.

### 3.6 Analysis of the parameter $\boldsymbol{\lambda}$ and the algorithm complexity

For (37) – (39), it is known that the optimization result depends on the parameter $\lambda$. In order to find the satisfactory solution, we need to increase $\lambda$ in few iterations. However, the effect of this parameter is not analyzed in the original GVD. Consequently, it is generally regulated by trail and error. Thus $\lambda$ will probably be given a large value, and more iterations will happen. Actually, there must exist a special $\lambda^*$ for the identical solution. This means the solutions of (37) – (39) will remain unchanged when $\lambda > \lambda^*$. Therefore, the computational burden of iteration will be reduced. $\lambda^*$ can be determined by combining (37) – (39) with the K-T condition. In the following, the method to determine $\lambda^*$ is presented by taking two-side model as an example.

At first, the maximum difference of priorities $\eta^*$ is got by means of solving

$$
\begin{aligned}
&\max\ (-\eta) \\
&\text{s.t.}\quad \text{constraints of (37).}
\end{aligned}
\tag{40}
$$

Secondly, based on $\eta^*$ and the K-T condition, the Lagrange function $F(\boldsymbol{z}, \boldsymbol{\beta}, \delta, \eta)$ is formulated and then differentiated. Finally, $\lambda^*$ is calculated by the following model:

$$
\begin{aligned}
&\min\ \ \lambda \\
&\text{s.t.}\quad \frac{\partial F}{\partial(\boldsymbol{z}^*, \boldsymbol{\beta}^*, \delta^*, \eta^*)} = 0 \\
&\qquad \text{Lagrange multipliers equations} \\
&\qquad \lambda \geqslant 0.
\end{aligned}
\tag{41}
$$

The corresponding algorithm steps are as follows.

**Step 1** When $\lambda \to \infty$, solve (40) and get the optimal solution $(\boldsymbol{z}^*, \boldsymbol{\beta}^*, \delta^*, \eta^*)$.

**Step 2** Formulate Lagrange function and K-T condition using $(\boldsymbol{z}^*, \boldsymbol{\beta}^*, \delta^*, \eta^*)$.

**Step 3** Solve (41) and find $\lambda^*$.

The similar algorithms can be derived for the single-side and hybrid-side cases.

From the above analysis of the parameter $\lambda$, it is seen that the computational complexity of the proposed MORTO method via GVD is not heavy. When the specific $\lambda^*$ can be got from the above algorithm, the iteration number of the original GVD approach is reduced greatly. So there are just limited iterations in this paper. Additionally, in the Step 3 of Section 3.5, the optimization models (37), (38) or (39) are solved using sequential quadratic programming (SQP). As we know, SQP is a successful nonlinear programming algorithm, and it is implemented by solving SQP problems. Therefore, the time complexity of our approach depends on the SQP algorithm. Since SQP has good convergence and fast computation performance, the proposed MORTO method via GVD also has little computational time. In this paper, although the number of optimization variables of (37) – (39) will increase for reasons of using direct collocation method to discretize the reentry trajectory optimization problem, actually this only enlarge the dimension of the SQP model. This does not result in the heavy computation. Hence, the effective strategy to balance precision and computational complexity is to determine a reasonable number of time nodes in the direct collocation method.

### 3.7 Reentry trajectory varying domain optimization model

Using the direct collocation method, the MORTO problem has been modeled as (25). In terms of Section 3.4, the four objectives need to be fuzzified. Firstly, the objectives (i) and (ii) are changed to "$\min\ \ f_1 = -\theta(t_f)$" and "$\min\ \ f_2 = -v(t_f)$" respectively. Then, their goal values and tolerances are determined by payoff table. Furthermore, the objectives are normalized and their varying domains are defined. Finally, combining (38), the reentry trajectory single-side varying domain model with the priorities is formulated as

$$
\begin{aligned}
&\max\ (\delta - \lambda\eta) \\
&\text{s.t.}\quad 0 \leqslant f_1'(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c) \leqslant \beta_1 - \delta\beta_1 \\
&\qquad 0 \leqslant f_2'(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c) \leqslant \beta_2 - \delta\beta_2 \\
&\qquad 0 \leqslant f_3'(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c) \leqslant \beta_3 - \delta\beta_3 \\
&\qquad \delta = 1 - f_4'(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c) \\
&\qquad \beta_1 - \beta_3 \leqslant \eta, \quad \beta_2 - \beta_3 \leqslant \eta, \quad \beta_3 - \beta_4 \leqslant \eta \\
&\qquad 0 \leqslant \beta_1, \beta_2, \beta_3 \leqslant 1, \quad \beta_4 = 1, \quad 0 \leqslant \delta \leqslant 1 \\
&\qquad (\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c) \in H(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c) \cap G(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c).
\end{aligned}
\tag{42}
$$

$\beta_j$ $(j = 1,2,3), \delta, \eta, \boldsymbol{X}, \boldsymbol{U}, \boldsymbol{U}^c$ are decision variables. Different solutions can be obtained by setting up different $\lambda$.

## 4. Simulations

### 4.1 Conditions

The constants of the hypersonic vehicle [25] are given in Table 2. The total time of the whole reentry process is 2 010 s, which is divided into 20 equal intervals. Thus there are 21 time nodes in simulations.

**Table 2 Constants of the hypersonic vehicle**

| Constant | Value | Constant | Value |
|---|---|---|---|
| $m$/slugs | 7 008 | $C_{L0}$ | −0.207 |
| $S$/ft$^2$ | 2 690 | $C_{L1}$ | 1.676 |
| $\mu$/ft$^3$/s$^2$ | $1.407\,7\times10^6$ | $C_{D0}$ | 0.078 54 |
| $R_0$/ft | $2.090\,29\times10^7$ | $C_{D1}$ | −0.352 9 |
| $\rho_0$/(slug/ft$^3$) | $2.378\times10^{-3}$ | $C_{D2}$ | 2.04 |
| $k$/ft | $4.201\,68\times10^{-5}$ | $h_a$ | 1.067 |
| $\dot{Q}_{\max}$/(Btu/ft$^2$/s) | 200 | $h_b$ | −1.101 |
| $q_{\max}$/(lb/ft$^2$) | 280 | $h_c$ | 0.698 8 |
| $n_{\max}$ | 2.5 | $h_d$ | −0.190 3 |
| $C$/(Btu·s$^{2.07}$/ft$^{3.57}$/slug$^{0.5}$) | $9.289\times10^{-9}$ | | |

The initial and terminal conditions, as well as the ranges of state and control variables are listed in Table 3.

**Table 3 Initial and terminal conditions, ranges of state, and control variables**

| Variable | Minimum | Maximum | Initial | Terminal |
|---|---|---|---|---|
| $\alpha/(^\circ)$ | −10 | 30 | | |
| $\sigma/(^\circ)$ | −80 | 80 | | |
| $h$/ft | 1 | $3\times10^5$ | $2.6\times10^5$ | $8\times10^4$ |
| $\phi/(^\circ)$ | −89 | 89 | 0 | |
| $\theta/(^\circ)$ | −90 | 90 | 0 | |
| $v/(\text{ft/s})$ | 1 | $3\times10^4$ | $2.56\times10^4$ | $\geqslant 2\,000$ |
| $\chi/(^\circ)$ | −180 | 180 | 90 | |
| $\gamma/(^\circ)$ | −89 | 89 | −1 | −5 |

### 4.2 Simulation results of the proposed method

According to the constants and constraints in Tables 2 and 3, the corresponding payoff table is constructed in Table 4.

**Table 4 Payoff table of MORTO problem**

| Single objective function | $f_1$ | $f_2$ | $f_3$ | $f_4$ |
|---|---|---|---|---|
| max $f_1$ | 34.313 | 2 000.8 | $1.486\times10^5$ | $4.812\,7\times10^{-4}$ |
| max $f_2$ | 22.283 8 | 2 605.4 | $1.248\,2\times10^5$ | 0.002 2 |
| min $f_3$ | 10.776 6 | 2 000.8 | $6.342\,4\times10^4$ | 0.002 1 |
| min $f_4$ | 18.930 9 | 2 183.5 | $1.200\,2\times10^5$ | $1.269\,6\times10^{-10}$ |

From Table 4, it is concluded that

$$f_1^* = 34.313, \quad f_1^{\min} = 10.776\,6$$

$$f_2^* = 2\,605.4, \quad f_2^{\min} = 2\,000.8$$

$$f_3^* = 6.342\,4\times10^4, \quad f_3^{\max} = 1.486\times10^5$$

$$f_4^* = 1.269\,6\times10^{-10}, \quad f_4^{\max} = 0.002\,2.$$

The goal values and the tolerant limits are substituted into (42) and then solved.

The simulation is executed on a general PC with 2G memory. Different parameter values for $\lambda$ are set up to achieve different optimal trajectories. The mean simulation time over each value is about 30 s. The optimization results are shown in Table 5.

**Table 5 Optimization results for different $\boldsymbol{\lambda}$**

| $\lambda$ | $\eta$ | $(\mu_{f_1}, \mu_{f_2}, \mu_{f_3}, \mu_{f_4})$ |
|---|---|---|
| 0 | 1 | (0.756 8, 0.756 8, 0.756 8, 0.756 8) |
| 1 | −0.227 2 | (0.790 5, 0.790 5, 0.703 4, 0.616 2) |
| 2 | −0.330 2 | (0.820 3, 0.820 3, 0.645 7, 0.471) |
| 3 | −0.371 3 | (0.820 7, 0.820 7, 0.562 3, 0.303 8) |

From the table, it is seen that with the change of $\lambda$, the satisfactory degree of each objective changes towards the requirement of priority. When $\lambda = 0$, there is $\eta = 1$ and all the satisfactory degrees are the same. This means each objective is optimized but the priorities are not considered. With the increase of $\lambda$, the value of $\eta$ decreases gradually and the difference between the satisfactory degrees expands. When $\lambda = 3$, the satisfactory degree of the objective (iv) equals “0.303 8”, which denotes this objective has a bad optimization. On the contrary, the objectives (i) and (ii) obtain better results “0.820 7”. The significant difference of the satisfactory degrees indicates the severe conflicts among the objectives.

Figs. 5 – 15 show all the optimal reentry trajectories for different $\lambda$. Fig. 5 and Fig. 6 present the profiles of attack angle and bank angle. In the initial reentry phase, the attack angle maintains a larger value even reaches the maximum, which is helpful to reduce the aerodynamic heating and stabilize the trajectory. However, in the later phase, it decreases gradually to achieve a larger flight range. Bank angle profile has a large vibration in the initial stage, but finally tends to zero. In Fig. 7, the profile of altitude has great oscillations in the early stage of reentry, and then it becomes smooth gradually. As shown in Fig. 8 and Fig. 9, longitude and latitude increase smoothly, while velocity and heading angle decrease sharply in Fig. 10 and Fig. 11. Wherein, the terminal latitude and velocity performances reflect the enhancement of the objectives (i) and (ii) with the increase of $\lambda$. Flight-path angle shown in Fig. 12 has the smaller oscillation when $\lambda$ decreases. Heat flux, dynamic pressure and overload in Figs. 13 – 15 are limited in the maximum permissible values.

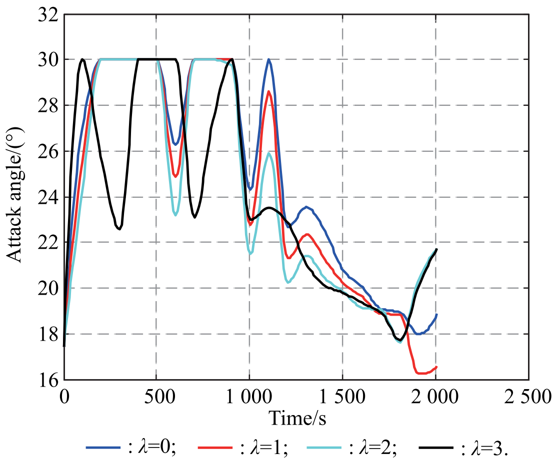



**Fig. 5 Attack angle profile about different λ**

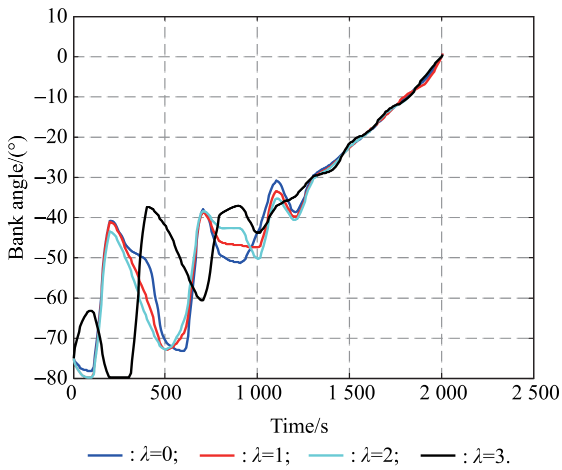



**Fig. 6 Bank angle profile about different λ**

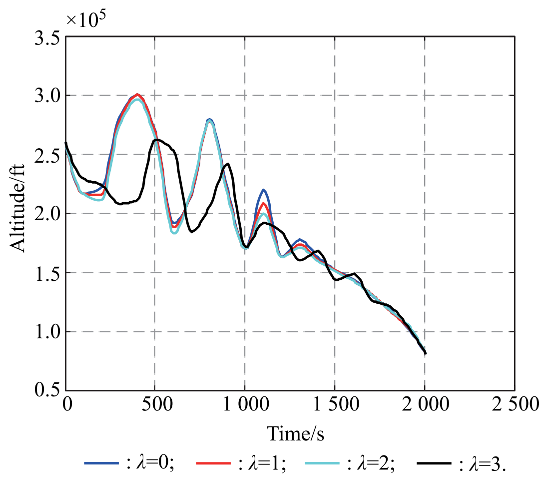



**Fig. 7 Altitude profile about different λ**

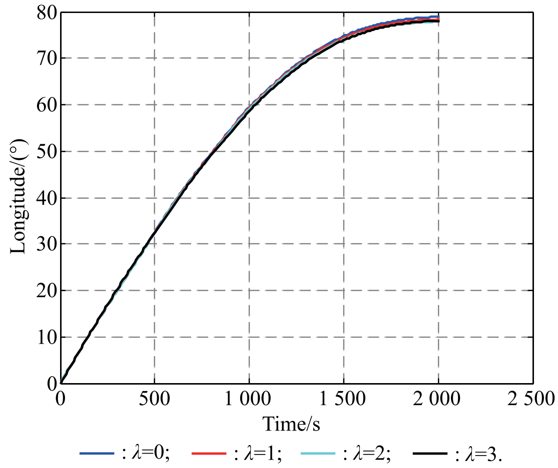



**Fig. 8 Longitude profile about different λ**

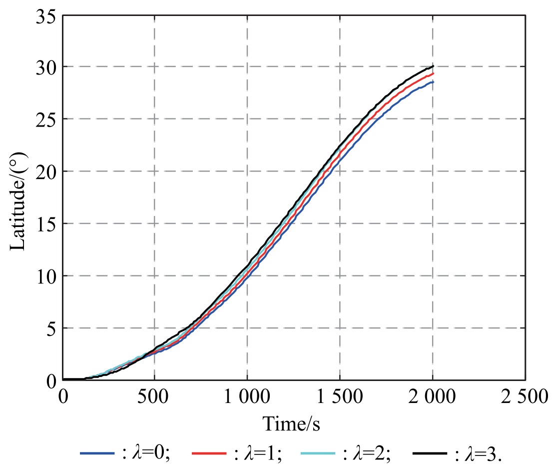



**Fig. 9 Latitude profile about different λ**

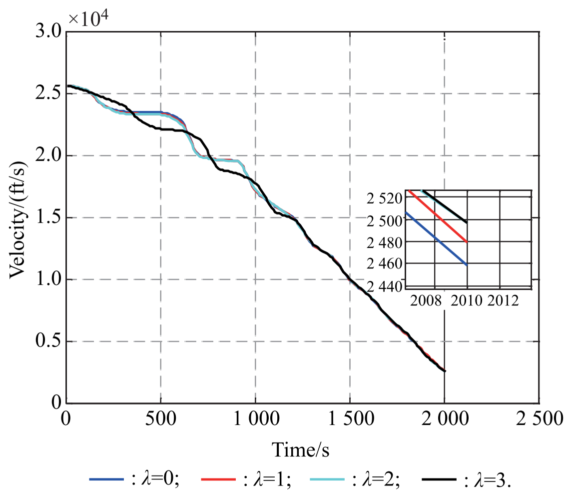



**Fig. 10 Velocity profile about different λ**

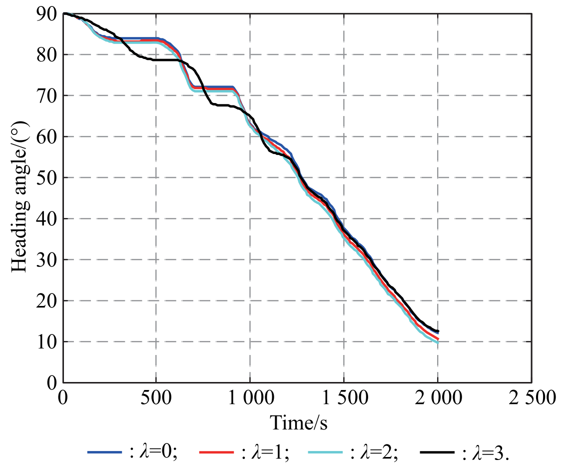



**Fig. 11 Heading angle profile about different λ**

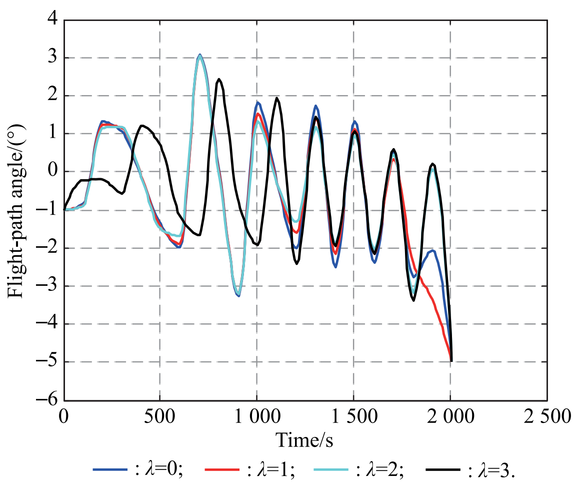



**Fig. 12 Flight-path angle profile about different λ**

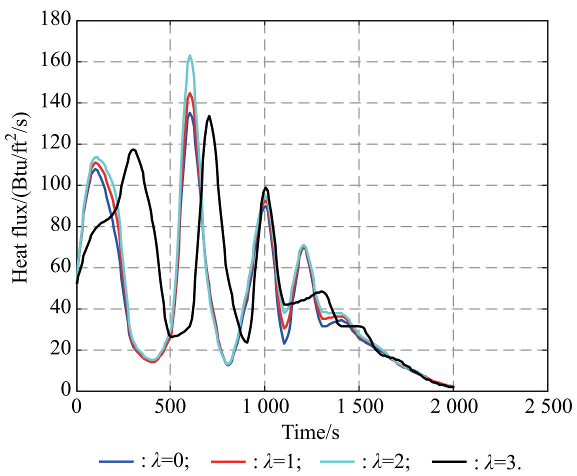



**Fig. 13 Heat flux profile about different λ**

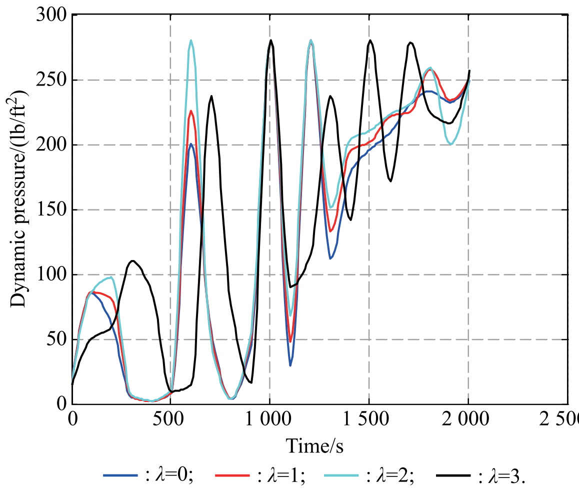



**Fig. 14 Dynamic pressure profile about different λ**

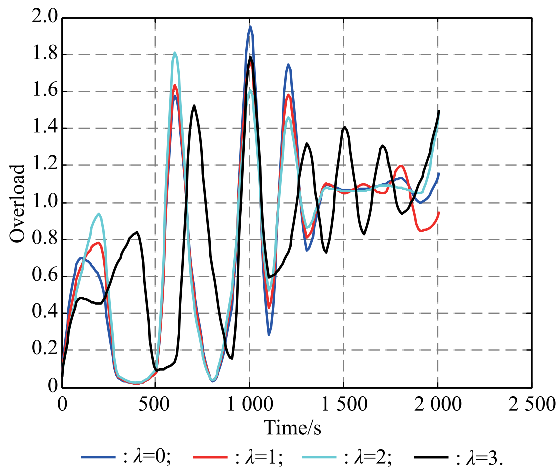



**Fig. 15 Overload profile about different λ**

When $\lambda = 2$, the trajectory is regarded as the most satisfactory since its priority difference is reasonable and the objectives are optimized as much as possible. Therefore, it can be chosen as the optimal trajectory.

## 4.3 Comparisons with traditional methods

(i) Lexicographic method

To validate its effectiveness, the traditional lexicographic method is adopted to solve the MORTO problem. The objectives (i) and (ii) in Priority one are optimized at first, and the satisfactory degrees of all the objectives are (0.991 4, 1, 0, 0.959 7). Then, (iii) and (iv) are optimized separately in sequence. However, the satisfactory degrees of the objectives remain unchanged, and so (0.991 4, 1, 0, 0.959 7) is the final result. This means that the degenerative optimization happens.

(ii) Weighted approach

In addition, the classical weighted approach is also used. By setting up different combinations of weights, we obtain various optimization results shown in Table 6.

**Table 6 Optimization results for weighted method**

| Weight | $(\mu_{f_1}, \mu_{f_2}, \mu_{f_3}, \mu_{f_4})$ |
|---|---|
| (0.3, 0.3, 0.25, 0.15) | (0.798 9, 1, 0.659 8, 0.968 4) |
| (0.35, 0.35, 0.2, 0.1) | (0.526, 0.832 4, 0.346 3, 0.430 3) |
| (0.4, 0.4, 0.15, 0.05) | (0.505 9, 0.857 4, 0.364, 0.068 6) |
| (0.45, 0.45, 0.099, 0.001) | (0.959 2, 1, 0.258 9, 0.907 7) |

From Table 6, it is known that even though various weight combinations are given for the same priority requirement, the order of satisfactory degrees does not conform with priority requirement. e.g., in the first case (0.3, 0.3, 0.25, 0.15), the satisfactory degree of (i) is lower than (iv). Moreover, in the second and the fourth combinations, the result of (iv) is higher than (iii). They destroy the priority requirement. Although the third weighted combination satisfies the priority order, the solution is inferior and cannot be considered as the optimal solution.

Compared with the weighted approach, the MORTO method via GVD is more effective. The priority order can be strictly satisfied by means of the enhanced GVD approach when $\eta \leqslant 0$. Nevertheless, the weighted method cannot guarantee this. Moreover, different optimization results of the objectives can be obtained by adjusting $\lambda$ in the GVD method. This is much simpler and easier than determining multiple weights. In addition, the case of the inferior solution does not exist in the GVD method, which is not ensured in the weighted approach.

### 4.4 Sensitivity analysis

In Fig. 5, the attack angle has reached its boundary in the early phase of reentry trajectory, which is mainly caused by the objective (iii) “minimum total aerodynamic heating”. Thus it is supposed that the attack angle will maintain smaller values if the importance of (iii) decreases. To demonstrate the impact from (iii) and to test the effectiveness of the GVD method, the sensitivity simulation with new priorities is executed. In the reassigned priorities, the objectives (i) and (ii) still belong to Priority one, but (iii) and (iv) are both reckoned Priority two. Correspondingly, the reentry trajectory varying domain optimization model is reformulated and solved again. The new optimization results are shown in Table 7, and the curve of attack angle is presented in Fig. 16.

**Table 7 Optimization results for new priorities**

| $\lambda$ | $\eta$ | $(\mu_{f_1}, \mu_{f_2}, \mu_{f_3}, \mu_{f_4})$ |
|---|---|---|
| 0 | 1 | (0.756 8, 0.756 8, 0.756 8, 0.756 8) |
| 1 | −0.816 2 | (0.901 2, 0.901 2, 0.462 4, 0.462 4) |
| 2 | −0.915 2 | (0.942 7, 0.942 7, 0.324 9, 0.324 9) |
| 3 | −0.946 9 | (0.96, 0.96, 0.248, 0.248) |

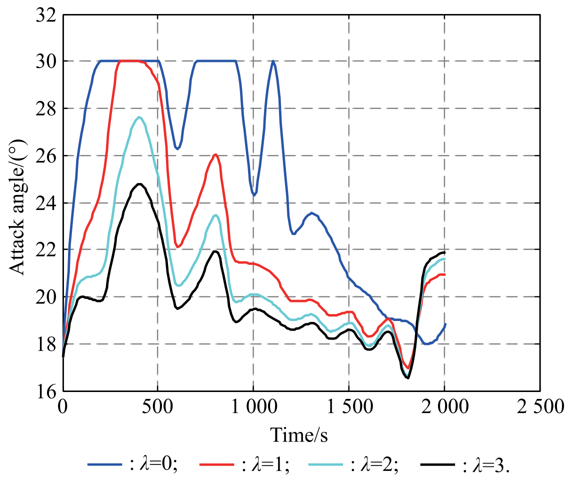



**Fig. 16 Attack angle profile for new priorities**

We can see that the attack angle has reduced obviously in the early times of the reentry phase when the new priority requirement is satisfied. The maximum of attack angle decreases greatly when $\lambda$ increases. This means that the effect of (iii) is weakened since it has the same priority as (iv).

## 5. Conclusions

In this paper, the MORTO method via enhanced GVD is designed to obtain the optimal reentry trajectory with multiple objectives and priorities for hypersonic vehicles. The direct collocation method is employed to convert the original MORTO problem into a nonlinear programming problem with priorities. The GVD strategy is introduced and enhanced. Priorities of three types of fuzzy objectives are modeled respectively using order constraints of varying domains. The specific $\lambda^*$ can be determined for maximum priority difference, and the algorithm computational performance is analyzed. By constructing the appropriate optimization models and regulating parameters, the optimal reentry trajectory can be acquired according to the actual requirement.

## References

[1] J. T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 1998, 21(2): 193 – 207.

[2] J. M. Lewallen, B. D. Tapley, S. D. Williams. Iteration procedures for indirect trajectory optimization methods. *Journal of Spacecraft and Rockets*, 1968, 5(3): 321 – 327.

[3] R. Hannemann-Tamás, W. Marquardt. How to verify optimal controls computed by direct shooting methods? — a tutorial. *Journal of Process Control*, 2012, 22(2): 494 – 507.

[4] C. R. Hargraves, S. W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *Journal of Guidance, Control, and Dynamics*, 1987, 10 (4): 338 – 342.

[5] G. T. Huntington, A. V. Rao. Optimal reconfiguration of spacecraft formations using the Gauss pseudospectral method. *Jour-*

*nal of Guidance, Control, and Dynamics*, 2008, 31(3): 689 – 698.

[6] P. Cheng, Z. J. Shen, S. L. Valle. RRT-based trajectory design for autonomous automobiles and spacecraft. *Archives of Control Sciences*, 2001, 11(3/4): 167 – 194.

[7] M. W. Sun, G. L. Jiao, Z. Q. Chen. Chaotic optimization based receding horizon trajectory design for reentry vehicle. *Journal of Astronautics*, 2012, 33(12): 1711 – 1718. (in Chinese)

[8] L. H. Tu, J. P. Yuan, Q. Fang, et al. Reentry skipping trajectory optimization using direct parameter optimization method. *Proc. of the 14th AIAA/AHI International Space Planes and Hypersonic Systems and Technologies Conference*, 2006: AIAA 2006 – 7993.

[9] D. B. Chen, F. Zou, R. Q. Lu, et al. Multi-objective optimization of community detection using discrete teaching-learning-based optimization with decomposition. *Information Sciences*, 2016, 369: 402 – 418.

[10] G. Chen, M. Xu, Z. M. Wan, et al. RLV reentry trajectory multi-objective optimization design based on NSGA-II algorithm. *Proc. of the AIAA Atmospheric Flight Mechanics Conference and Exhibit*, 2005: AIAA 2005 – 6131.

[11] E. Mooij, P. G. Hänninen. Distributed global trajectory optimization of a moderate lift-to-drag re-entry vehicle. *Proc. of the AIAA Guiance, Navigation, and Control Conference*, 2009: AIAA 2009 – 5770.

[12] E. M. Yong, L. Chen, G. J. Tang. Trajectory optimization of hypersonic gliding reentry vehicle based on the physical programming. *Acta Aeronautica et Astronautica Sinica*, 2008, 29(5): 1091 – 1097. (in Chinese)

[13] B. B. Pal, B. N. Moitra. A goal programming procedure for solving problems with multiple fuzzy goals using dynamic programming. *European Journal of Operational Research*, 2003, 144(3): 480 – 491.

[14] L. H. Chen, F. C. Tsai. Fuzzy goal programming with different importance and priorities. *European Journal of Operational Research*, 2001, 133(3): 548 – 556.

[15] S. Y. Li, C. F. Hu. Two-step interactive satisfactory method for fuzzy multiple objective optimization with preemptive priorities. *IEEE Trans. on Fuzzy Systems*, 2007, 15(3): 417 – 425.

[16] C. F. Hu, S. Y. Li. Enhanced interactive satisfying optimization approach to multiple objective optimization with preemptive priorities. *International Journal of Information Technology & Decision Making*, 2006, 5(1): 47 – 63.

[17] S. Y. Li, C. F. Hu, C. J. Teng. A generalized varying-domain optimization method for fuzzy goal programming with priorities based on a genetic algorithm. *Control and Cybernetics*, 2004, 33(4): 633 – 652.

[18] C. F. Hu, C. J. Teng, S. Y. Li. A fuzzy goal programming approach to multi-objective optimization problem with priorities. *European Journal of Operational Research*, 2007, 176(3): 1319 – 1333.

[19] W. W. Cai, Y. W. Zhu, L. P. Yang, et al. Optimal guidance for hypersonic reentry using inversion and receding horizon control. *IET Control Theory and Applications*, 2015, 9(9): 1347 – 1355.

[20] D. G. Hull. Conversion of optimal control problems into parameter optimization problems. *Journal of Guidance, Control, and Dynamics*, 1997, 20(1): 57 – 60.

[21] A. L. Herman, B. A. Conway. Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules. *Journal of Guidance, Control, and Dynamics*, 1996, 19(3): 592 – 599.

[22] L. H. Tu, J. P. Yuan, X. K. Yue, et al. Improving design of reentry vehicle trajectory optimization using direct collocation method. *Journal of Northwestern Polytechnical University*, 2006, 24(5): 653 – 657. (in Chinese)

[23] R. E. Bellman, L. A. Zadeh. Decision-making in a fuzzy environment. *Management Science*, 1970, 17(4): B141 – B164.

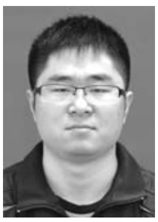
[24] R. N. Tiwari, S. Dharmar, J. R. Rao. Fuzzy goal programming — an additive model. *Fuzzy Sets and Systems*, 1987, 24(1): 27 – 34.

[25] S. Josselyn, I. M. Ross. Rapid verification method for the trajectory optimization of reentry vehicles. *Journal of Guidance, Control, and Dynamics*, 2003, 26(3): 505 – 508.

## Biographies

**Chaofang Hu** was born in 1973. He received his Ph.D. degree from Shanghai Jiao Tong University, China in 2007. He is currently an associate professor at School of Electrical Engineering and Automation, Tianjin University. His research interests include aircraft optimization and control, multi-objective optimization and predictive control.
E-mail: cfhu@tju.edu.cn

**Yue Xin** was born in 1990. He received his B.S. degree from Central South University, China in 2012. He is currently a graduate student at School of Electrical Engineering and Automation, Tianjin University. His research interests include trajectory optimization and guidance of aircraft.
E-mail: xinyue_tju@163.com



**Hao Feng** was born in 1992. He received his B.S. degree from University of Science and Technology Liaoning, China in 2014. He is currently a graduate student at School of Electrical Engineering and Automation, Tianjin University. His research interests include guidance of hypersonic vehicle.
E-mail: fenghaozz@126.com