Research Article

Attainability Analysis for Entry Vehicles Based on Differential Evolution

Songtao Chang, Yongji Wang, Lei Liu, and Xing Wei

Key Laboratory of Ministry of Education for Image Processing and Intelligent Control, School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China

Correspondence should be addressed to Yongji Wang; wangyjch@mail.hust.edu.cn

Received 5 November 2013; Accepted 7 January 2014; Published 17 March 2014

Academic Editor: Shuhui Bi

Copyright © 2014 Songtao Chang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Attainable region provides crucial information on mission planning of entry vehicles. In order to obtain it, a series of nonlinear optimal control problems which have similar formulations are needed to be solved. However, it is difficult to compute due to severe nonlinearity of the dynamics and various constraints. In this paper, a novel method is established to generate the attainable region at the end of the entry phase. It utilizes the parallel feature of differential evolution (DE) and the high accuracy of Chebyshev polynomial interpolation. By using the Chebyshev polynomial interpolation, the original problem is transformed to several nonlinear programming problems to facilitate employing DE. Each individual in DE's population represents a candidate point on the boundary of the attainable region. In order to lead the population to the boundary simultaneously, a scheme is devised by exploiting the parallel feature of DE. Different from conventional methods which generate one point of the boundary in each run, our proposed method generates one side of the boundary of the attainable region. A scenario is presented to evaluate the designed method and some analyses are conducted to evaluate the influence of the vehicle's design parameters on the attainable region.

1. Introduction

Attainable region at the end of the entry phase (i.e., landing footprint) provides critical information on mission planning and evaluation of an entry vehicle's design. In order to obtain it, a series of nonlinear optimal control problems which have similar formulations are needed to be solved. The optimal goal is maximizing the terminal crossrange at each potential downrange under path constraints and terminal constraints. However, it is challenging to get accurate solutions of these problems due to severe nonlinearity of the vehicle's dynamics and various constraints. Some researchers derived approximate solution of the original problems by simplifying the vehicle's dynamics [1–3], in which path constraints were not considered, and the Earth's rotation was omitted. Therefore, the solution was inadequately accurate and poorly reliable. Saraf et al. [4] presented a rapid landing footprint computation algorithm, which was built around acceleration-based trajectory planner. Meanwhile, a near-optimal angle of attack control law was developed. But the footprint was inadequately accurate and the computed maximum downrange was much

smaller than the one that the vehicle could achieve. Lu and Xue [5] developed a near-optimal closed-form control law by using quasiequilibrium glide condition (QEGC) [6]. It transformed the original problem to several closest-approach problems which were univariate root-finding ones. Path constraints were enforced through inflicting limitations on bank angle. However, only bank angle was optimized, and angle of attack was predetermined. Benito and Mease [7] used a direct method to solve the entry trajectory optimization problem. Direct method [8–11] transcribed the original optimal control problem to nonlinear programming problem (NLP) which could be solved easily. The method had the capacity to deal with path constraints and terminal constraints simultaneously. The accuracy of the solution can be improved by adjusting some parameters used in the transcription.

It is noted that all of the aforementioned methods use determined optimization algorithm (e.g., sequential quadratic programming [12]) to solve the transcribed nonlinear programming problem. The performances of conventional optimization algorithms are highly affected by the initial guess of the solution since they are local optimization

algorithms. To overcome this deficiency, some metaheuristic optimization algorithms [13–16] were proposed to approximate the global optimal solution. They were often inspired by some natural phenomena, such as genetic algorithm (GA) [13] which mimicked the process of natural selection, particle swarm optimization (PSO) [14] which was inspired from the movement of organisms like a flock of birds or a school of fishes, differential evolution (DE) [15] which mimicked the process of natural evolution, and simulated annealing (SA) [16] which was inspired by annealing in metallurgy. They made few or no assumptions on the problem to be optimized, could search very large spaces of candidate solutions, and did not need to have a good initial guess of the solution. Due to these advantages, they were widely applied in many fields, such as controller design [17], damage detection [18, 19], economic dispatch problem [20], and water distribution system optimization [21].

In this paper, a parallel method is developed to calculate the landing footprint of entry vehicles based on DE and Chebyshev polynomial interpolation. The original optimal control problems are transcribed to NLPs by Chebyshev polynomial interpolation. After that, an improved DE is employed with each agent representing a candidate point on the boundary of the landing footprint. A new selection strategy is designed to drive the agents evolving to the boundary of the landing footprint simultaneously. It does not need to provide a good initial guess of the solution and can generate one side rather than one point of the boundary of the landing footprint by exploiting the parallel feature of DE. Furthermore, it does not rely on simplification of the system and considers all control signals which make it have high fidelity. A scenario is presented to evaluate the method. Through the analysis of the dynamic equations of the vehicle, design configuration can be translated to the lift and drag coefficients equivalently. In order to evaluate the effect of the vehicle's design parameters on the attainable region, various lift and drag configurations are used to generate the landing footprints.

The remainder of this paper is organized as follows. Formulation of the landing footprint problem is given in Section 2. Landing footprint computation algorithm is established in Section 3. In Section 4, a scenario is presented to validate the algorithm and some analyses are conducted. Finally, some remarks are concluded in Section 5.

2. Problem Formulation

Three degrees of freedom dynamic equations of an entry vehicle are as follows:

$$\dot{r} = v \sin \gamma, \quad (1)$$

$$\dot{\theta} = \frac{v \cos \gamma \sin \psi}{r \cos \phi}, \quad (2)$$

$$\dot{\phi} = \frac{v \cos \gamma \cos \psi}{r}, \quad (3)$$

$$\dot{v} = \omega^2 r \cos \phi (\sin \gamma \cos \phi - \cos \gamma \sin \phi \cos \psi) - D - g \sin \gamma, \quad (4)$$

$$\dot{\gamma} = \frac{L \cos \sigma}{v} - \frac{g}{v} \cos \gamma + \frac{v}{r} \cos \gamma + 2\omega \cos \phi \sin \psi + \frac{\omega^2 r \cos \phi (\cos \gamma \cos \phi + \sin \gamma \cos \psi \sin \phi)}{v}, \quad (5)$$

$$\dot{\psi} = \frac{L \sin \sigma}{v \cos \gamma} + \frac{v}{r} \cos \gamma \sin \psi \tan \phi - 2\omega (\tan \gamma \cos \psi \cos \phi - \sin \phi) + \frac{\omega^2 r}{v \cos \gamma} \sin \psi \sin \phi \cos \phi, \quad (6)$$

$$g = \frac{\mu}{r^2}, \quad \rho = \rho_0 e^{-(r-r_0)/h_s}, \quad (7)$$

$$L = \frac{\rho v^2 S C_L}{(2m)}, \quad D = \frac{\rho v^2 S C_D}{(2m)},$$

where r is the radial distance from the earth center to the vehicle, θ and ϕ are the longitude and latitude, respectively, v is the earth-relative velocity, γ is the relative flight-path angle, ψ is the relative velocity heading angle measured clockwise from the north, ω is the earth self-rotation rate, m is the vehicle's mass, S is the reference area, ρ is the density of atmosphere, ρ_0 is the density of atmosphere at sea level, r_0 is the earth's radius, C_L and C_D are the lift and drag coefficients determined by angle of attack α and Mach Ma , g is the gravity acceleration, independent variable is the time t , and control variables are angle of attack α and bank angle σ . The geometric sketch of entry flight is illustrated by Figure 1.

Entry process must satisfy constraints of heating rate, dynamic pressure, and normal acceleration:

$$\dot{Q} = c \sqrt{\rho} v^{k_q} \leq \dot{Q}_{\max}, \quad (8)$$

$$q = \frac{\rho v^2}{2} \leq q_{\max}, \quad (9)$$

$$n = L \cos \alpha + D \sin \alpha \leq n_{\max}, \quad (10)$$

where $\dot{Q}_{\max}$, $q_{\max}$, and $n_{\max}$ represent maximum allowable heating rate, dynamic pressure, and acceleration, respectively.

In order to facilitate the attitude controller design, entry trajectory should not oscillate acutely and should satisfy QEGC as follows:

$$L \cos \sigma - g + \frac{v^2}{r} \approx 0. \quad (11)$$

Following the entry phase is Terminal Area Energy Management (TAEM) phase. Terminal state must be restrained as follows:

$$\begin{aligned} |r(t_f) - r_f| &\leq \Delta r, & |v(t_f) - v_f| &\leq \Delta v, \\ |\gamma(t_f) - \gamma_f| &\leq \Delta \gamma, \end{aligned} \quad (12)$$

FIGURE 1: Geometric sketch of entry flight.

where t_f is the terminal time of the entry phase and r_f, v_f , and γ_f represent the ideal terminal states of the entry phase, respectively.

Constraints of heating rate, dynamic pressure, normal acceleration, and terminal states are hard constraints, which must be satisfied. QEGC is a soft constraint which may be violated a little.

Landing footprint of an entry vehicle is the attainable region when it reaches TAEM. It can be formulated to a series of optimal control problems. Each has prescribed terminal downrange (the range along the norm-plan) and the optimal objective is maximizing or minimizing terminal crossrange (the range perpendicular to the norm-plan).

Let $R(t_f), Z(t_f)$ be the downrange and the crossrange at the end of the entry phase, respectively. To calculate the landing footprint of an entry vehicle, we have to solve $(2N+2)$ optimal control problems whose objectives are described as follows:

$$R_{\min} = \min(J_1) = \min(R(t_f)).$$

$$R_{\max} = \max(J_1) = \max(R(t_f)).$$

$$\min(J_2) = \min(Z(t_f)) \text{ subject to } R(t_f)$$

$$= R_{\min} + k \frac{R_{\max} - R_{\min}}{N-1}, \quad k = 0, 1, \dots, N-1.$$

$$\max(J_2) = \max(Z(t_f)) \text{ subject to } R(t_f)$$

$$= R_{\min} + k \frac{R_{\max} - R_{\min}}{N-1}, \quad k = 0, 1, \dots, N-1, \quad (13)$$

where N is the number of the calculated sites to approximate the lateral boundary:

$$R(t_f) = r(t_f) \cdot \arccos \frac{\mathbf{r}_b(t_f) \cdot \mathbf{r}_i}{|\mathbf{r}_b(t_f)| \cdot |\mathbf{r}_i|},$$

$$Z(t_f) = r(t_f) \cdot \left(\frac{\pi}{2} - \arccos \frac{\mathbf{r}(t_f) \cdot \mathbf{n}}{|\mathbf{r}(t_f)| \cdot |\mathbf{n}|} \right),$$

$$\mathbf{r}_b(t_f) = \mathbf{r}(t_f) - (\mathbf{r}(t_f) \cdot \mathbf{n}) \cdot \mathbf{n},$$

$$\mathbf{r}_i = r_i \cdot [\cos \phi_i \cos \theta_i, \cos \phi_i \sin \theta_i, \sin \phi_i],$$

$$\mathbf{r}(t_f) = r(t_f) \cdot [\cos \phi(t_f) \cos \theta(t_f), \cos \phi(t_f) \sin \theta(t_f), \sin \phi(t_f)],$$

$$\mathbf{n} = \frac{\mathbf{v}_i \times \mathbf{r}_i}{|\mathbf{v}_i \times \mathbf{r}_i|},$$

$$\begin{aligned} \mathbf{v}_i = v_i \cdot [& \cos \theta_i \cos \phi_i \sin \gamma_i - \sin \theta_i \cos \gamma_i \sin \psi_i \\ & - \cos \theta_i \sin \phi_i \cos \gamma_i \cos \psi_i, \sin \theta_i \cos \phi_i \sin \gamma_i \\ & + \cos \theta_i \cos \gamma_i \sin \psi_i - \sin \theta_i \sin \phi_i \cos \gamma_i \cos \psi_i, \\ & \sin \phi_i \sin \gamma_i + \cos \phi_i \cos \gamma_i \cos \psi_i], \end{aligned} \quad (14)$$

where $r_i, \theta_i, \phi_i, v_i, \gamma_i$, and ψ_i are the states at the initial time of the entry phase.

3. Landing Footprint Computation Algorithm

In order to calculate the landing footprint of an entry vehicle, we must solve a series of multiconstraints optimal control problems whose objectives are described as (13), state equations are described as (1)–(6), and constraints are described as (8)–(12). These optimal control problems have similar formulations; the only difference among them is the way to deal with the terminal crossrange and downrange. By applying this characteristic and the parallel feature of DE, an algorithm is designed to solve half of these problems (maximizing or minimizing) simultaneously. First, the angle

of attack's scope is determined by path constraints to simplify the problem. And the QEGC constraint (11) is guaranteed by inflicting the low bound of the angle of attack. Then, control profiles are parameterized by Chebyshev polynomial interpolation to facilitate employing DE algorithm. After that, DE is given. At last, an improved DE is applied to calculate the landing footprint. Each run of the algorithm generates one side of the landing footprint. To get the full landing footprint, it just needs to run the algorithm twice.

3.1. Discretization of the Control Profiles. From (8)-(9) and the expression of density ρ in (7), we have

$$\begin{aligned} r &\geq r_0 - 2h_s \ln \left(\frac{\dot{Q}_{\max}}{\sqrt{\rho_0} c v^{k_q}} \right) \triangleq r_Q(v), \\ r &\geq r_0 - h_s \ln \frac{2q_{\max}}{\rho_0 v^2} \triangleq r_q(v). \end{aligned} \quad (15)$$

Let $y(\alpha, v, r, \sigma) = L \cos \sigma - g + v^2/r = (S/2m)\rho v^2 \cos \sigma C_L(\alpha, v, r) - \mu/r^2 + v^2/r$, according to (11), $y \approx 0$. Solve the equation $y = 0$ with dependent variable being α ; we have $\alpha = \alpha(v, r, \sigma)$,

$$\begin{aligned} \frac{\partial y}{\partial \sigma} &= -L \sin \sigma, & \frac{\partial y}{\partial \alpha} &= \frac{\rho v^2 S}{2m} \frac{\partial C_L}{\partial \alpha} \cos \sigma, \\ \frac{\partial y}{\partial r} &= \frac{2g}{r} - \frac{v^2}{r^2} - \frac{L}{h_s} \cos \sigma + \frac{L}{C_L} \frac{\partial C_L}{\partial r} \cos \sigma, \end{aligned} \quad (16)$$

$\sigma \in [-90^\circ, 90^\circ]$, $L > 0$; thus, $\partial y/\partial \sigma > 0$ when $\sigma \in [-90^\circ, 0^\circ]$ and $\partial y/\partial \sigma < 0$ when $\sigma \in (0^\circ, 90^\circ]$. Lift coefficient C_L increases as α increases; that is, $\partial y/\partial \alpha > 0$. Consider $\partial \alpha(v, r, \sigma)/\partial \sigma = -(\partial y/\partial \sigma)/(\partial y/\partial \alpha)$; then, $\partial \alpha(v, r, \sigma)/\partial \sigma < 0$ when $\sigma \in [-90^\circ, 0^\circ]$ and $\partial \alpha(v, r, \sigma)/\partial \sigma > 0$ when $\sigma \in (0^\circ, 90^\circ]$; therefore, $\alpha(v, r, \sigma) \geq \alpha(v, r, 0)$.

$\partial C_L/\partial r$ and $2g/r$ are significantly smaller than L/h_s , so they can be ignored; then $\partial y/\partial r < 0$, $\partial \alpha(v, r, \sigma)/\partial r = -(\partial y/\partial r)/(\partial y/\partial \alpha) > 0$; therefore

$$\begin{aligned} \alpha &= \alpha(v, r, \sigma) \geq \alpha(v, r, 0) \\ &\geq \alpha(v, \max(r_Q(v), r_q(v)), 0) \triangleq \alpha_{\min}(v). \end{aligned} \quad (17)$$

Suppose the maximum angle of attack and bank angle are $\alpha_{\max}$ and $\sigma_{\max}$, respectively, and the minimum bank angle is $\sigma_{\min}$. Chebyshev interpolation polynomial is an optimal approximation of the original function in all interpolation polynomials in L_∞ norm-sense [22]. Herein, control profiles are approximated by Chebyshev interpolation polynomials as follows:

$$\begin{aligned} \eta(\tau) &= \sum_{i=1}^{N_1} \eta_i \Phi_i(\tau), & \xi(\tau) &= \sum_{i=1}^{N_2} \xi_i Y_i(\tau). \\ \alpha(v) &= (1 - \eta(\tau)) \alpha_{\min}(v) + \eta(\tau) \alpha_{\max}, \\ \sigma(v) &= (1 - \xi(\tau)) \sigma_{\min} + \xi(\tau) \sigma_{\max}, \end{aligned} \quad (18)$$

where $\tau = 2(v - v_0)/(v_f - v_0) - 1 \in [-1, 1]$, $\Phi_i(\tau)$ is the i th function of basis functions of N_1 th order Chebyshev

polynomial interpolation, and $Y_i(\tau)$ is the i th function of basis functions of N_2 th order Chebyshev polynomial interpolation. N_1 and N_2 are the orders of the interpolation polynomials of α and σ , respectively.

Let $\mathbf{X} = [\eta_1, \eta_2, \dots, \eta_{N_1}, \xi_1, \xi_2, \dots, \xi_{N_2}]$, $\eta_k, \xi_k \in [0, 1]$, and $\mathbf{X}$ is the decision vector. When $\mathbf{X}$ is known, control profiles are determined by (18); then we can integrate (1)-(6) to figure out the objective function (13), and the constraints (8)-(10) and (12). The nonlinear programming problems are well defined.

3.2. Differential Evolution. DE is a simple, efficient, and robust evolutionary algorithm that deals with unconstrained parameter optimization problem [15]. Unconstrained parameter optimization can be formulated to minimize a single D -dimensional function $f(\mathbf{X})$, $\mathbf{X} = [x_1, x_2, \dots, x_D]$, $x_i \in [x_{l,i}, x_{u,i}]$. Let $P_G = \{\mathbf{X}_{G,1}, \mathbf{X}_{G,2}, \dots, \mathbf{X}_{G,NP}\}$ represent the G th generation of the population, where the i th individual's position is $\mathbf{X}_{G,i} = [x_{G,i,1}, x_{G,i,2}, \dots, x_{G,i,D}]$, $i = 1, 2, \dots, NP$; NP is the population's scale.

DE involves two stages: initialization and evolution. Initialization stage uses (19) to generate the initial population P_0 randomly. Then P_0 evolves to P_1 , P_1 evolves to $P_2, \dots$, and so on, until the terminal conditions are reached. Each evolving phase includes three operations, namely, differential mutation, crossover, and selection:

$$\begin{aligned} x_{0,i,j} &= x_{l,j} + \text{rand}[0, 1] \times (x_{u,j} - x_{l,j}), \\ i &= 1, 2, \dots, NP, & j &= 1, 2, \dots, D, \end{aligned} \quad (19)$$

where $\text{rand}[0, 1]$ is a uniform random number in $[0, 1]$.

Mutation and crossover create offsprings of the population, and the selection operation controls the direction of evolution. The following will describe these operations.

Let $\mathbf{V}_{G,i} = [v_{G,i,1}, v_{G,i,2}, \dots, v_{G,i,D}]$ denote the i th mutant individual. Classical mutation operator is described as follows:

$$\mathbf{V}_{G,i} = \mathbf{X}_{G,r_1} + F \times (\mathbf{X}_{G,r_2} - \mathbf{X}_{G,r_3}), \quad (20)$$

where r_1, r_2 , and r_3 are mutually exclusive indices randomly chosen in the range $[1, NP]$, which are different from the base index i . $F > 0$ is the mutant factor.

The mutant individual $\mathbf{V}_{G,i}$ exchanges its components with the base vector $\mathbf{X}_{G,i}$ through crossover operation to get the trial vector $\mathbf{U}_{G,i} = [u_{G,i,1}, u_{G,i,2}, \dots, u_{G,i,D}]$. Common crossover operation is described as follows:

$$u_{G,i,j} = \begin{cases} v_{G,i,j}, & j = j_{\text{rand}} \text{ or } \text{rand}[0, 1] \leq C_R \\ x_{G,i,j}, & \text{otherwise,} \end{cases} \quad (21)$$

where j_{rand} is a random index in the range $[1, D]$. It ensures that the trial vector has at least one component from the mutant vector. $C_R > 0$ is the crossover probability.

By using one to one greedy selection operation to decide whether the trial vector $\mathbf{U}_{G,i}$ substitutes $\mathbf{X}_{G,i}$ to the next iteration. This strategy guarantees that the population evolves to superior region.

Step 1. Set the population scale NP, the mutant factor F , the crossover probability C_R , the maximum generations $G_{\max}$, the orders of the interpolation polynomials of α and σ (N_1 and N_2), the number of subintervals for downrange splitting N , and let $G = 0$. Initialize the population P_0 using (19), and calculate the downrange, the crossrange and the constraints of each individual in P_0 by integrating the state equations (1)–(6) using the control curves described as (18) with decision vector $\mathbf{X} = \mathbf{X}_{G,i}$.

Step 2. For $i = 1$ to NP
 Do mutation and crossover with the i th individual $\mathbf{X}_{G,i}$ to generate the offspring $\mathbf{U}_{G,i}$. Calculate the downrange, the crossrange and the constraints of $\mathbf{U}_{G,i}$ by integrating the state equations using the control curves described as (18) with decision vector $\mathbf{X} = \mathbf{U}_{G,i}$.
 End For

Step 3. Calculate the individuals' fitness of P_G and its offspring.

Step 4. Choose NP most favorite individuals (have larger fitness) from P_G and its offspring to form the next population P_{G+1} and set $G = G + 1$.

Step 5. If $G \leq G_{\max}$
 Go to *Step 2*.
 Else
 Print the boundary of the landing footprint and the according trajectories, and then terminate the procedure.
 End If

PSEUDOCODE 1: Pseudocode of the landing footprint computation algorithm.

FIGURE 2: Schematic diagram for classification of all feasible individuals.

3.3. Landing Footprint Generation Algorithm. When coming to calculate the landing footprint, the algorithm must guarantee that none of the constraints is violated, and it should have extreme crossrange and maintains diversity in downrange distribution. DE uses fitness to drive the evolving of the population. Here, we design a scheme to determine the fitness of each agent to lead the agent to the boundary of the landing footprint.

Let T_G represent the population constituted by P_G and its offspring. At the first stage, sort all individuals in T_G according to the violation magnitude of the constraints (magnitude of the normalized constraints vector) in ascending order. Suppose it has M sets that are denoted as F_k , $k = 1, 2, \dots, M$. All individuals who do not violate any constraint (these individuals also called feasible individuals) have the highest

priority, and they are denoted as F_1 . At the second stage, downrange and crossrange are used to classify F_1 . Suppose it has L classifications. We set the fitness of each individual in the j th level classification in F_1 as $M + L - j$ ($j = 1, 2, \dots, L$), the fitness of each individual in F_k as $M + 1 - k$ ($k = 2, 3, \dots, M$). In this manner, the fitness is higher when the violation magnitude of the constraints is smaller.

The fitness of feasible individuals (individuals in F_1) is determined by the classification according to the downrange and crossrange. We will detail the classification approach of the individuals in F_1 . First, split the whole interval into N subintervals evenly according to the span of the downrange of F_1 , and then fill all individuals in each subinterval. Denote all individuals in the i th subinterval as χ_i (suppose it has κ_i individuals); then $L = \max(\kappa_i)$. The j th level classification includes each subinterval's individual corresponding to the j th most favorite (largest/smallest) crossrange. The individuals corresponding to the minimum downrange and the maximum downrange of F_1 are in the first classification too. By picking up favorite individuals from each subinterval, this method balances the optimality in crossrange and the diversity in the distribution of downrange.

In order to make it easier to understand, a schematic diagram is shown in Figure 2 ($N = 3$ in this case). The whole interval is divided into three subintervals evenly by the two vertical dashed lines. Fill all individuals in each subinterval; then, $\chi_1 = \{1, 2, 3, 4\}$, $\chi_2 = \{5, 6, 7, 8\}$, and $\chi_3 = \{9, 10\}$. Therefore, $\kappa_1 = \kappa_2 = 4$, $\kappa_3 = 2$, $L = 4$. The first level classification includes $\{2, 3, 9, 1, 10\}$; the second level classification includes $\{3, 8\}$; the third level classification includes $\{4, 6\}$; and the fourth level classification includes $\{7\}$. This scheme for assigning the fitness considers the constraints firstly and then leads the feasible individuals to the boundary of the landing footprint. Through splitting the potential downrange interval into several subintervals evenly and making each subinterval's agent have the same probability to

TABLE 1: Constraints of the trajectories according to the landing footprint.

	$\max(\dot{Q})$ (W/m ²)	$\max(q)$ (N/m ²)	$\max(n)$ (g)	$ \Delta r(t_f) $ (m)	$ \Delta v(t_f) $ (m/s)	$ \Delta \gamma(t_f) $ (degree)
Maximum	431249	9916	2.41	1952	19.7	0.96
Mean	406557	8614	1.83	955	9.6	0.44
Minimum	346819	7242	1.56	5	0.8	0.01
Standard	26432	584	0.13	442	5.9	0.28

survive, the algorithm balances the optimality in crossrange and the diversity in the distribution of downrange.

The pseudocode of the landing footprint computation method is described as Pseudocode 1. First, initialize the population and calculate the corresponding downrange, crossrange, and constraints. Then, evolve the population of DE until the maximum generations are reached. And, the evolving direction is determined by superior agents (i.e., have large fitness) of the population. Run the algorithm to generate one side of the landing footprint according to the maximum crossrange due to different downrange with positive bank angle. Then, use the same algorithm to generate the other side of the landing footprint with negative bank angle.

4. Simulation

An entry scenario of X-33 [23, 24] is used to validate the algorithm described in this paper.

The parameters of the entry problem are as follows:

$$\begin{aligned}
 [r_i, \theta_i, \phi_i, v_i, \gamma_i, \psi_i] &= [6499518 \text{ m}, -117.01^\circ, -18.255^\circ, \\
 &\quad 7622 \text{ m/s}, -1.4379^\circ, 30.0^\circ] \\
 [r_f, v_f, \gamma_f] &= [6408427 \text{ m}, 908 \text{ m/s}, -7.5^\circ], \\
 [\Delta r, \Delta v, \Delta \gamma] &= [2000 \text{ m}, 20 \text{ m/s}, 1.0^\circ], \\
 Q_{\max} &= 431259 \text{ W/m}^2, \\
 q_{\max} &= 11970 \text{ N/m}^2, \\
 n_{\max} &= 2.5 \text{ g}, \quad 5^\circ \leq \alpha \leq 45^\circ, \\
 |\dot{\alpha}| &\leq \frac{5^\circ}{s}, \quad |\sigma| \leq 90^\circ, \quad |\dot{\sigma}| \leq \frac{5^\circ}{s}.
 \end{aligned} \tag{22}$$

The parameters of the algorithm: NP = 200, F = 0.7, C_R = 0.5, G_{max} = 500, N = 50, N₁ = 5, N₂ = 12.

4.1. Landing Footprint of X-33 in Nominal Configuration. The landing footprint of X-33 in nominal configuration is described in Figure 3, and the center of each circle in the figure is the terminal site of each trajectory. Control variables are described in Figures 4(a) and 4(b), and the constraints are described in Table 1. It can be found that all constraints are satisfied and the control profiles are smooth. By observing the control profiles, the angle of attack varies a lot to achieve

FIGURE 3: Landing footprint with ground tracks of X-33 in nominal configuration.

optimality which indicates that it is necessary to take the angle of attack under consideration.

The time consumption to calculate the landing footprint is 323.2 s; that is, the mean cost to calculate each landing point is 3.2 s, under the environment:

Compiler: Microsoft Visual Studio 2010
CPU: AMD Athlon(tm) II X3 435
RAM: 4 G.

A conventional method is used to verify our method. This method solves $(2N + 2)$ optimal control problems (optimal objectives are described as (13)). Each run generates one point in the boundary of the landing footprint. This method discretizes the control profiles as (18) and applies the conventional DE algorithm to solve each optimal control problem. The landing footprint calculated by this method is shown in Figure 3 (the red line with +) too. It shows that the two boundaries are almost the same, except the near boundary of the landing footprint. Near boundary corresponds to large path constraints and these trajectories are dangerous to operate, so the conservatism of our method has little effect in engineering. The time consumption of the conventional method is 15852.3 s under the same environment; that is, it takes 155.4 s to calculate one point of the boundary. It demonstrates that our method is very efficient (nearly 50 times fast) compared with the conventional method.

4.2. Influences of Vehicle's Design Parameters on Landing Footprint. By analyzing the dynamic equations of entry vehicles (1)–(6), the vehicle is maneuvered by the lift and drag acceleration (L and D). From (7), L and D are determined by four design parameters: the mass of the vehicle m , the reference area of the vehicle S , the lift coefficient C_L , and the drag coefficient C_D . The deviations of the mass m and the reference area S can be transformed to the deviations of the lift and drag coefficients C_L and C_D equivalently. For

TABLE 2: Deviations of the lift and drag coefficients.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
ΔC_L	-5.0%	-5.0%	0.0%	-5.0%	0.0%	+5.0%	+5.0%	0.0%	+5.0%
ΔC_D	+5.0%	0.0%	+5.0%	-5.0%	0.0%	+5.0%	0.0%	-5.0%	-5.0%
$\frac{(1 + \Delta C_L)}{(1 + \Delta C_D)} - 1$	-9.5%	-5.0%	-4.8%	0.0%	0.0%	0.0%	5.0%	5.3%	10.5%

FIGURE 4: Control curves according to the boundary of the landing footprint in nominal configuration.

FIGURE 5: Landing footprints with ground tracks in various lift and drag configurations of X-33.

example, increasing the reference S is equivalent to increase C_L and C_D simultaneously at the same rate. To analyze the effect of the design parameters on the landing footprint of

the entry vehicle, nine different deviations of the lift and drag coefficients described in Table 2 are used to generate the landing footprints. Cases (1)–(9) are in ascending order according to the ratio between the lift and drag coefficients (lift-to-drag ratio). The landing footprint is described by Figure 5. From the figure, we can find out that the landing footprint is larger when the lift coefficient becomes bigger or the drag coefficient becomes smaller, but the landing footprint is almost the same when the deviations of the lift and drag coefficients are in the same manner (i.e., lift-to-drag ratio does not change). It is indicated that the lift-to-drag ratio is the most important factor to the attainability of the entry vehicle and the attainability enhances as the lift-to-drag becomes larger.

5. Conclusion

This paper developed a parallel algorithm to generate the attainable region at the end of the entry phase based on DE and Chebyshev polynomial interpolation. By exploiting the parallel feature of DE, a scheme is designed to lead the individuals of DE's population evolving to the boundary of the landing footprint. It generates one side of the boundary of the attainable region while conventional methods generate one point of the boundary in each run. Furthermore, the algorithm takes all control signals into consideration and enforces path and terminal constraints, which make

it have high fidelity. An entry scenario of X-33 is used to validate the algorithm and some analyses are conducted. It is indicated that the algorithm has the capacity to deal with multiconstraints and to maintain the diversity in the distribution of downrange in the landing footprint. Through the analyses, we can find out that the lift-to-drag ratio is the most important factor to the attainability of the entry vehicle and the attainability enhances as the lift-to-drag gets larger.

Conflict of Interests

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

Acknowledgments

This work is partially supported by National Nature Science Foundation of China no. 61203081 and no. 61174079, Doctoral Fund of Ministry of Education of China no. 20120142120091, and Precision Manufacturing Technology and Equipment for Metal Parts no. 2012DFG70640.

References

- [1] N. X. Vinh, *Optimal Trajectories in Atmospheric Flight*, Elsevier, New York, NY, USA, 1981.
- [2] D. N. Anhtuan and B. B. William, "Longitudinal control and footprint analysis for a reusable military launch vehicle," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, vol. 5738, AIAA, Austin, Tex, USA, 2003.
- [3] D. N. Anhtuan and B. D. David, "Footprint determination for reusable launch vehicles experiencing control effector failures," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, vol. 4775, AIAA, Monterey, Calif, USA, 2002.
- [4] A. Saraf, J. A. Leavitt, K. D. Mease, and M. Ferch, "Landing footprint computation for entry vehicles," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, vol. 4774, pp. 228–241, AIAA, Providence, RI, USA, 2004.
- [5] P. Lu and S. Xue, "Rapid generation of accurate entry landing footprints," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 3, pp. 756–767, 2010.
- [6] P. Lu, "Asymptotic analysis of quasi-equilibrium glide in lifting entry flight," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 3, pp. 662–670, 2006.
- [7] J. Benito and K. D. Mease, "Reachable and controllable sets for planetary entry and landing," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 3, pp. 641–654, 2010.
- [8] A. L. Herman and B. A. Conway, "Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 3, pp. 592–599, 1996.
- [9] M. El-Kady and E. M. E. Elbarbary, "A Chebyshev expansion method for solving nonlinear optimal control problems," *Applied Mathematics and Computation*, vol. 129, no. 2-3, pp. 171–182, 2002.
- [10] 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.
- [11] D. Garg, M. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, vol. 46, no. 11, pp. 1843–1851, 2010.
- [12] J. Nocedal and S. J. Wright, *Numerical Optimization*, Series in Operations Research, edited by P. Glynn and S. M. Robinson, Springer, Berlin, Germany, 1999.
- [13] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, 1975.
- [14] R. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.
- [15] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [16] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [17] J. O. Pedro, M. Dangor, O. A. Dahunsi, and M. M. Ali, "Differential evolution-based PID control of nonlinear full-car electrohydraulic suspensions," *Mathematical Problems in Engineering*, vol. 2013, Article ID 261582, 13 pages, 2013.
- [18] F. Kang, J. Li, and S. Liu, "Combined data with particle swarm optimization for structural damage detection," *Mathematical Problems in Engineering*, vol. 2013, Article ID 416941, 10 pages, 2013.
- [19] L. Vincenzi, G. de Roeck, and M. Savoia, "Comparison between coupled local minimizers method and differential evolution algorithm in dynamic damage detection problems," *Advances in Engineering Software*, vol. 65, pp. 90–100, 2013.
- [20] A. Elaiw, X. Xia, and A. Shehata, "Hybrid DE-SQP method for solving combined heat and power dynamic economic dispatch problem," *Mathematical Problems in Engineering*, vol. 2013, Article ID 982305, 7 pages, 2013.
- [21] F. Zheng, A. C. Zecchin, and A. R. Simpson, "Self-adaptive differential evolution algorithm applied to water distribution system optimization," *Journal of Computing in Civil Engineering*, vol. 27, no. 2, pp. 148–158, 2012.
- [22] W. Gautschi, *Numerical Analysis*, Birkhauser, Boston, Mass, USA, 2nd edition, 2012.
- [23] P. Lu and J. M. Hanson, "Entry guidance for the X-33 vehicle," *Journal of Spacecraft and Rockets*, vol. 35, no. 3, pp. 342–349, 1998.
- [24] Z. Shen and P. Lu, "Onboard generation of three-dimensional constrained entry trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 1, pp. 111–121, 2003.