ICAS 2021 SHANGHAI
International Council of the Aeronautical Sciences
32nd Congress of the International Council of the Aeronautical Sciences
September 6-10, 2021
Pudong Shangri-La, Shanghai, China

Ascent Trajectory Optimization of Hypersonic Vehicle Using Swarm Intelligent Method

Zhiyu Wang, Peng Chen, Yalong Wang, Weidong Huang

Department of Aerospace Science and Technology

Space Engineering University, Beijing 101416

Abstract

The optimization of ascent phase trajectory for a single-stage-to-orbit, air-breathing hypersonic vehicle is investigated using improved Particle Swarm Optimization (IPSO). The strong coupling relationship of trajectory, aerodynamics and propulsion system is examined, which makes the problem notoriously difficult. Aerodynamic and thrust model are developed in couple with trajectory state. The optimal problems are hence formulated as fuel-optimal control problem and time-optimal control problem. This paper presents transformation of such optimal control problem into parameters optimization problem by analytic function expression with several parameters providing advantages for PSO solution. Traditional PSO has advantages in complicated optimization problem in terms of global optimization, and does not need any gradient information to solve optimization problem numerically, while tends to evolve to local optimum. To handle constraints imposed by trajectory state and propulsion system, a modification in the fitness function of particles is executed through penalty function. To avoid local optimum, a novel mechanism of neighbor component in PSO is proposed. The basic scheme of improvement is to increase a dimension from which the particles can learn. After balancing learning process, diversity of particles in swarm is increased during the process of searching optimization solution by the improved algorithm which is capable to avoid local optimum where diversity is relatively small. Neighbor component model is calculated by K-means algorithm, an effective algorithm consuming less time, during each iteration. Time complexity of improved PSO algorithm is O(2n^2), which is very close to traditional PSO algorithm with O(n^2). Hence, IPSO method is implemented to determine the optimal solution without consuming more time significantly. The numerical example demonstrates that IPSO is superior to other PSO variants in terms of better solution and larger diversity of particles. Also in comparison with finite difference method, performance index of IPSO is nearly the same as finite method within error of 0.74%. Additionally, three sets of optimization simulation results for varied initial conditions with regard to initial altitude, velocity and weight are presented, which demonstrates the validation of proposed optimization algorithm. Initial altitudes are varied in the first set of solutions with constant initial velocity of 2500 m/s, and along the minimum-fuel trajectory, minimum fuel usage is reached at the cost of bigger amount of flight time. The second set of solution is determined when initial velocity is varied，and as the initial velocity increases, fuel usage increases and the final time decreases. The condition of third set of solutions is varied initial weight of vehicle. In the early stage of climbing, light vehicle will obtain larger velocity due to larger acceleration, and consequently larger thrust due to bigger velocity. Whereas in the later stage, increase of altitude results in decrease of thrust, which counteracts the influence caused by increase of velocity.

Keywords: Particle Swarm Optimization; Ascent Phase; Trajectory Optimization; Hypersonic Vehicle

1. Introduction

The payload cost for space shuttle comes out to $7000 to $8000 per pound [1] . In order to reduce the expensive cost, development of single-stage, air-breathing and reusable vehicle has been sought after for many years, which is powered by turbine engine and dual-mode scramjet engine that can take off horizontally. Such kind of engine utilizes air as the oxidizer, thus the specific impulse can reach approximately 3000s saving plenty of propulsion cost when compared with the rocket. Whereas, series of such vehicle characteristics e.g. thrust and fuel consumption, are influenced by vehicle altitude, angle-of-attack, and Mach number, especially during the ascent phase. Additionally, such vehicle often covers a large range and the ascent phase will last for several minutes. Thus, tremendous difficulty will be encountered in the design of the ascent trajectory.

Studies on supersonic vehicle trajectory optimization can retrospect to 1980s. Paris and Joosten [2] examined the coupling relationship between propulsion system and trajectory characteristics of a ramjet propelled missile. Transformation of continuous optimal control problem to discrete parameter optimization was done via Chebyshev polynomial representation of the state and controls. Subsequently, Paris and Hargraves [3] employed cubic polynomials to substitute Chebyshev polynomial and used collocation to satisfy the differential equations. It’s noted that this method is easy to program, and become one of mainstream approach in 1990s.

Some earliest researches of trajectory optimization were done by Corban and Calise in 1990s. Methods of singular perturbation and energy-state approximation were applied to solve ascent trajectory optimization problem [4] -[7] . These researches were not faultless and the state constraints were hard to handle. Another important branch of methods for trajectory optimization is inverse dynamics developed by Lu [8] [9] . By employing this approach, original optimal problem is reduced into a NLP (Nonlinear Programming problem).

A large stride was made after 2000 indicated by development of a kind of novel trajectory optimization method, namely, Gauss pseudo-spectral method (GPM). With regard to this pseudo-spectral method, control and state variables are discretized by global polynomials, and the differential-algebraic equations are estimated via orthogonal collocation [10] [11] . This method is still dominant and efficient today in research of various trajectory optimizations [12] -[15] .

While in recent years, PSO (Particle Swarm Optimization) is applied to trajectory optimization successfully [16] -[19] . As a novel heuristic algorithm, PSO is of high efficiency to convergence and easy to programming. However, PSO algorithm quite tends to evolve to local optimization. Accordingly, improvements of PSO are proposed in this paper to avoid local optimum.

Generally, numerical algorithms for optimal control problem can be divided into two categories: direct method and indirect method. The former, direct method, transcribes the optimal control problem to NLP, and the resulting NLP problem can be addressed numerically. Direct methods can be divided into several categories: state and control parameterization [2] -[6] , pseudo-spectral method [12] -[13] and direct shooting method. In indirect methods [20] [21] first-order necessary condition is derived from the optimal control problem via calculus of variations and Pontryagin maximum principle. Then

Hamiltonian boundary value problem is obtained to determine optimal solutions.

This paper makes efforts to develop suitable PSO algorithm to treat trajectory optimization problem. Ref. [22] mentioned that among heuristic method, PSO is of superiority due to its fast convergence speed, convenience, robust and effectiveness. PSO algorithm, as heuristic method, does not suffer from the discontinuity of the optimal problem, i.e. propulsion system's switching. In this paper, the optimal control problem is transformed into parameters optimization problem. Parameters required to be optimized are the inputs of the Improved PSO (IPSO).

The structure of this paper is organized as follows. The single-stage-to-orbit vehicle model and equation of motion are described in Sec. 2. Trajectory optimization problems aiming at minimum-fuel and minimum-time-to-climb are formulated in Sec.3. Numerical solution via improved particle swarm optimization and corresponding verification are presented in Sec.4. In Sec.5, three sets of simulation results and discussion are shown.

2. Hypersonic vehicle model and kinetic model

The equations of motion of the three-degree freedom point mass system over a spherical and non-rotating earth are used to describe the dynamics of the hypersonic vehicle.

$$\begin{cases} \dfrac{dv}{dt} = \dfrac{[T(v,h)cos\alpha - D(v,h)]}{m} - g(h)\sin\gamma \\ \dfrac{d\gamma}{dt} = \dfrac{T(v,h)\sin\alpha + L(v,h)}{mv} - [\dfrac{g(h)}{v} - \dfrac{v}{r}]\cos\gamma \\ \dfrac{dh}{dt} = v\sin\gamma \\ \dfrac{dl}{dt} = v\cos\gamma \\ \dfrac{dm}{dt} = -\dfrac{T}{I_{sp}g(h)} \end{cases} \tag{1}$$

Where, v is the velocity of the vehicle, γ is the flight path angle, h is the flight altitude, m is the mass of the vehicle, T is the thrust, α is the angle-of-attack, and L and D are lift force and drag force respectively.

The single-stage-to-orbit vehicle model used in this paper was based on Ref. 21. The GHAME (Generic Hypersonic Aerodynamic Model Example) model provides some important kinetic and propulsion parameters. The reference area S_{ref} is 6000 ft^2, and the area of the intake of the engine A_C is 300 ft^2, and the initial gross weight to take off has been estimated to 300,000 pounds.

The lift force and drag force for the GHAME model are calculated as following equation:

$$\begin{cases} L = qC_L S_{ref} \\ D = qC_D S_{ref} \\ q = \rho v^2/2 \end{cases} \tag{2}$$

Where $\rho(h)$ is the atmospheric density that decreases dramatically when the altitude increases. The lift and drag coefficient, C_L and C_D respectively, are actually the function of Mach number Ma and angle-of-attack α .

$$C_L = C_{L_0}(Ma,\alpha) + C_{L_\alpha}(Ma,\alpha)\alpha + \mathrm{C}_{L_{\delta e}}(Ma,\alpha)\delta e \tag{3}$$

$$C_D = C_{D_0}(Ma,\alpha) + C_{D\alpha}(Ma,\alpha)\alpha \tag{4}$$

In order to cover the range for both Mach and altitude, a combined cycle engine system, refer to as TBCC, is used to provide the propulsion of the hypersonic vehicle incorporating turbojet, ramjet and scramjet. And the switch points for different propulsion systems are: turbojet, $0 < Ma < 2$; ramjet, $2 < Ma < 6$;scramjet, $6 < Ma$.

Thrust was formulated neatly in this research through the following equation:

$$\begin{cases} T = \sigma g(h) I_{sp} \dfrac{dm}{dt} \\ \dfrac{dm}{dt} = C_T(Ma,\alpha)\rho v A_C \Rightarrow T = 0.029\phi I_{sp}(\phi, Ma)\rho g(h) v C_T(Ma,\alpha) A_C \\ \sigma = 0.029\phi \end{cases} \tag{5}$$

Where ϕ is the throttle command associated with fuel flow varying from 0 to 2, and σ=0.029ϕ is the stoichiometric fuel/air ratio of the engine. $C_T(Ma,\alpha)$, capture-area coefficient, was also introduced to handle the situation when the engine switches into the scramjet. $C_T(Ma,\alpha)$ is the function of Mach number and angle-of-attack, which is used to describe the effective weight flow under different condition of Mach and angle-of-attack since the air does not enter the cowl uniformly.

According to the eq. (5), the surface diagram of thrust with respect to Mach and height is shown in Fig. 1. It can be detected that the thrust decreases strongly as the flight height increases, which is mostly caused by the dramatic reduction of local atmosphere density to the extent of several orders of magnitude especially in the proximity space. While the height is beyond 30 km, the decease of thrust caused by reduction of atmosphere density becomes slowing down due to the steady tendency of atmosphere density in the proximity space, and the results reveals inherent connection between atmosphere density and thrust of air-breathing hypersonic vehicle. Large thrust provided by engine during the low altitude benefits the maneuver of climbing out in initial stage of trajectory. This is achieved by accelerating the vehicle to some extent when the lift force obtains enough dynamic pressure q . Benefiting from the increase of lift force, the vehicle can be able to climb out fast before leveling out at the final altitude.

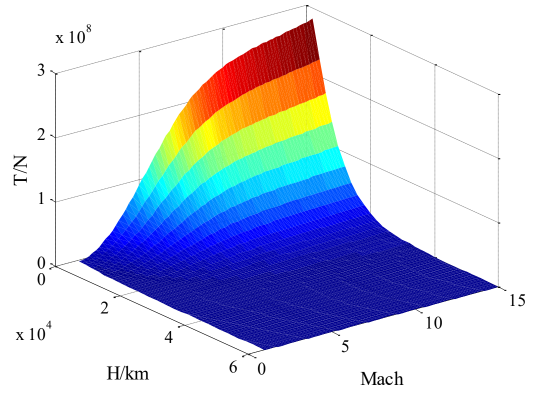


Fig 1. Surface diagram of thrust

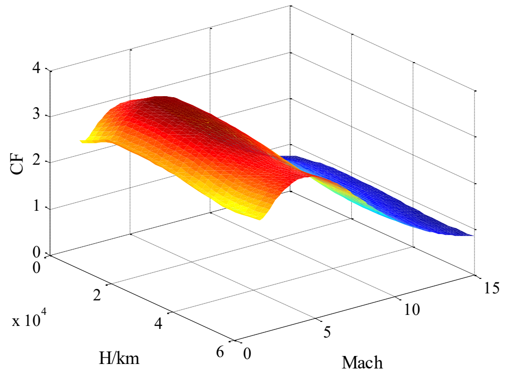


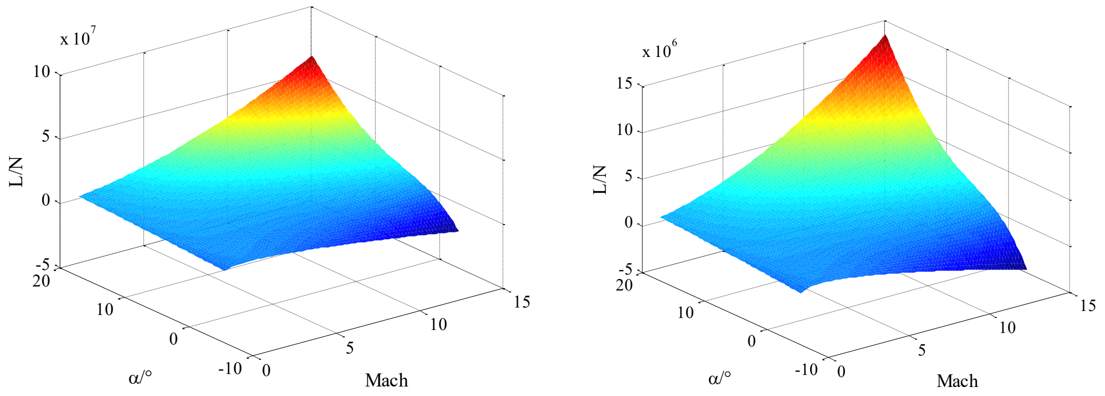
Fig 2. Surface diagram of thrust coefficient

In order to reveal the relationship between Mach and thrust, thrust coefficient is defined as follow.

$$CF = \frac{T(Ma,h)}{q(Ma,h)A_C} \tag{6}$$

The surface diagram of thrust coefficient with respect to Mach and height is shown in Fig 2.At the low Mach number, the thrust coefficient increases as Mach increases. However, the trend changes apparently when the Mach exceeds 3. Increase of Mach number brought about reduction of thrust coefficient.

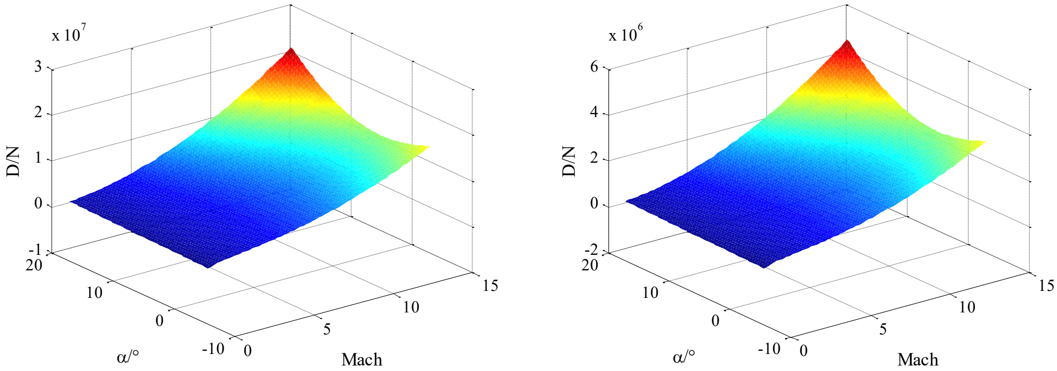
In Fig 3 (a) & (b), the surface diagram of lift force with respect to Mach and angle-of-attack at the altitude of 20 km and 30 km are shown, respectively. And in Fig 4 (a) & (b), the information of drag force under the same situations is given.



Altitude of 20 km

Altitude of 30 km

Fig.3 Lift of GHAME vs. Mach and angle-of-attack



Altitude of 20 km

Altitude of 30 km

Fig. 4 Drag of GHAME vs. Mach and angle-of-attack

In general, the lift and drag force at lower altitude are both larger than at higher altitude due to the dense air. When the angle-of-attack is negative, the lift force is also negative, and with the increase of Mach number, the absolute value of lift force increases.

3. Trajectory optimization problem statement

3.1 Optimal objectives

Based on the GHAME vehicle model, several variables depicting the dynamic characteristics of vehicle i.e. $\mathbf{x} = (h, \gamma, l, v, m)^{\mathrm{T}}$ were chose as the state variables and the angle-of-attack α was chose as the control variable. Details of these variables were presented in Table 1 denoting boundary values and constraints.

Table 1 Parameters constraints and boundary values

Types	Control parameter	State parameters					Time
	α / deg	h / m	γ / deg	v / (m/s)	l / m	m / kg	t / s
Initial	free	h_0	0	v_0	0	m_0	0
Min	-3	0	-90	$v_{\min}$	0	free	0
Max	21	free	90	$v_{\max}$	free	free	$t_{\max}$
Final	free	h_c	γ_c	v_f	l_f	m_f	t_f

The optimization objectives in this research are minimum-fuel and minimal time-to-climb trajectory. Consistent with the objectives, optimization variables were formulated by Eq. (7) and Eq. (8).

$$J_{\min-fuel} = m_0 - m_f \tag{7}$$

$$J_{\min-time} = t_f \tag{8}$$

3.2 Constraints

Considering optimization problem of hypersonic air-breathing vehicle in real world, variables must be regulated within appropriate value.

3.2.1 Dynamic pressure constraint

Dynamic pressure, $q(t) = \rho v^2 / 2$, needs to be regulated within boundary value because it can contribute large structural load in process of climbing as well as heat stress.

Constraints are defined as follows.

$$S_1 = q - q_{\max} < 0 \tag{9}$$

$$S_2 = -q + q_{\min} < 0 \tag{10}$$

3.2.2 Angle-of-attack constraint

The boundary values of angle-of-attack were given in Table 1. For control variables, examination of its boundary value is vital.

$$S_3 = \alpha - \alpha_{\max} < 0 \tag{11}$$

$$S_4 = -\alpha + \alpha_{\min} < 0 \tag{12}$$

3.2.3 Terminal constraints

The details of terminal constraints were also shown in Table 1, which were formulated by Eq. (13) and Eq. (14).

$$\varphi_{f1} = h_f - 32000 = 0 \tag{13}$$

$$\varphi_{f2} = \gamma_f - 3 = 0 \tag{14}$$

3.3 Summary of the optimization problem

According to optimal control theory, the trajectory optimization problem can be summarized through following equations. Determine the state, $\mathbf{x}(t) \in \mathbb{R}^5$, control, $u(t) \in \mathbb{R}$, initial time t_0, and final time, t_f, that minimize the cost function:

$$\min_{u(t)} J = \Phi(\mathbf{x}_f, t_f) + \int_{t_0}^{t_f} L(\mathbf{x}, u, t)\,\mathrm{d}t \tag{15}$$

For minimum-fuel objective, $J = m_0 - m_f$ (16)

For minimum time-to-climb objective, $J = t_f$ (17)

Subject to the constraints of Eq. (18) to Eq. (20):

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, u, t) \tag{18}$$

Obviously, this equation denotes Eq. (1) to Eq. (5).

$$\Phi_f(\mathbf{x}_f, t_f) = [\varphi_{f1}\ \varphi_{f2}]^{\mathrm{T}} = 0 \tag{19}$$

$$\mathbf{S}(\mathbf{x}, u, t) = [S_1\ S_2\ S_3\ S_4]^{\mathrm{T}} \le 0 \tag{20}$$

4 Numerical solution via IPSO

PSO, falling into the category of swarm intelligence methods [23] , was introduced by Eberhart and Kennedy in 1995. PSO is famous for its simplicity and convergence performance. The main advantage of traditional PSO is the unique information sharing mechanism to find optimal solution benefiting the fast convergence and convenience when treating constraints and discontinuities within physical model. This unique mechanism denotes cooperation of all particles in finding the best solution which will be formulated in subsequent section. Ref. [24] claims that particle swarm technique is more efficient with respect to generic algorithms. And it was mentioned in Ref. [25] that PSO outperforms the differential evolution algorithm also. Despite its performance and advantages, only a few of research aiming at trajectory optimization were developed via PSO. Because there are two main difficulties will be encountered when implementing PSO. Firstly, treating equality and inequality constraints is troublesome. Second, PSO algorithm quite tends to evolve to local optimum.

With regard to the two difficulties, penalty function and improvements of structure of PSO are proposed in this paper.

4.1 Traditional particle swarm optimization

During the iteration, a certain number of particles are involved to find the best solution together. Each particle owns two kinds of properties. One is solution property, referred to as position vector $\vec{x}$, and the other is mutation property, referred to as velocity vector $\vec{v}$. Parameters in optimal control problem constitute position vector and every particle can be recognized as a possible solution. The velocity vector includes the information of each particle's mutation which helps the particle to find better solution.

Without loss of generality, this paper assume M to be size of the swarm and D to be the dimension of parameters. Position vector and velocity vector can be formulated as $\vec{x} = (x_1, x_2, \cdots x_D)$ and $\vec{v} = (v_1, v_2, \cdots, v_D)$. For each particle the previous best solution property in history will be remarked as p_{best}, while the global best solution is g_{best}.Obviously, g_{best} is of superiority with respect to p_{best}. Particles' mutation in single iteration follows:

$$\begin{cases} v_{i,j}^{t+1} = wv_{i,j}^{t} + c_1 r_1 (p_{best}^{t}(i,j) - x_{i,j}^{t}) + c_2 r_2 (g_{best}^{t}(i,j) - x_{i,j}^{t}) \\ x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t} \end{cases}, i = 1, 2, \cdots, M,\ j = 1, 2, \cdots, D \tag{21}$$

Where $v_{i,j}^{t+1}$ denotes i th particle's mutation with regard to j th parameter in $(t+1)$ th iteration; w is the inertial weight; c_1 and c_2 are acceleration coefficients; c_1 represents the cognitive self-adjustment component, while c_2 represents the social influence on particle; r_1 and r_2 are random numbers, which are uniformly distributed on the interval between 0 and 1.

The time-dependent control variable angle-of-attack $\alpha(t)$ is defined through the following function.

$$\begin{cases} \alpha(t) = x_1 t^3 + x_2 t^2 + x_3 t + x_4 + \mathrm{x}_5 \sin(x_6 t) + x_7 \cos(x_8 t) \\ \vec{x} = (x_1, x_2, \cdots, x_8) \\ \vec{v} = (v_1, v_2, \cdots, v_8) \end{cases} \tag{22}$$

Position vector is regulated within the range as follows:

$$x_1 \in [-0.001, 0.001], x_2 \in [-0.1, 0.1], x_3 \in [-1,1], x_4 \in [-3, 25], x_5, x_7 \in [-5, 5], x_6, x_8 \in [-1,1] \tag{23}$$

4.2 Equality and inequality constrains

The most popular approach to handle equality constraints is penalty function adding to the objective function [26] . In this research, equality constrains include final constrains Eq. (19).

The penalty function is:

$$\Gamma_\gamma(\mathbf{x}) = k_1 |\gamma_f - \gamma_c| \tag{24}$$

$$\Gamma_h(\mathbf{x}) = k_2 |h_f - h_c| \tag{25}$$

Where k_1 and k_2 are used to normalize the parameters in the same numerical order, γ_c and h_c are set to be constant.

Inequality constrains include state constraints indicated by Eq. (20). State constraints associated with state variables of angle-of-attack, and dynamic pressure should be regulated more strictly. Once the state variables exceed the range, this solution is recognized as failure. The penalty function is

$$\Gamma_{S_1}(\mathbf{x}) = \max\left\{0, \frac{q - q_{\max}}{|q - q_{\max}|} N_{\max}\right\} \tag{26}$$

$$\Gamma_{S_2}(\mathbf{x}) = \max\left\{0, \frac{-q + q_{\min}}{|-q + q_{\min}|} N_{\max}\right\} \tag{27}$$

$$\Gamma_{S_3}(\mathbf{x}) = \max\left\{0, \frac{\alpha - \alpha_{\max}}{|\alpha - \alpha_{\max}|} N_{\max}\right\} \tag{28}$$

$$\Gamma_{S_4}(\mathbf{x}) = \max\left\{0, \frac{-\alpha + \alpha_{\min}}{|-\alpha + \alpha_{\min}|} N_{\max}\right\} \tag{29}$$

Considering the final constraints, the fitness function for PSO is defined as follows. The pseudo-code is shown in Table 2.

$$J_{\min-fuel} = (m_0 - m_f) + \Gamma_\gamma(\mathbf{x}) + \Gamma_h(\mathbf{x}) + \Gamma_{S_1}(\mathbf{x}) + \Gamma_{S_2}(\mathbf{x}) + \Gamma_{S_3}(\mathbf{x}) + \Gamma_{S_4}(\mathbf{x}) \tag{30}$$

$$J_{\min-time} = t_f + \Gamma_\gamma(\mathbf{x}) + \Gamma_h(\mathbf{x}) + \Gamma_{S_1}(\mathbf{x}) + \Gamma_{S_2}(\mathbf{x}) + \Gamma_{S_3}(\mathbf{x}) + \Gamma_{S_4}(\mathbf{x}) \tag{31}$$

Table 2 Pseudo-code of PSO

```
For $M$ particles:
    Determine initial value of $\vec{x}$ and $\vec{v}$
end
1. Repeat for $j$ th iteration
2. For $i = 1, 2, \cdots, M$ do
3.     Calculate $J^j_{\min-fuel}$ and $J^j_{\min-time}$ for $M$ particles according to Eq. 30 and Eq. 31
4.       For $i$ th particle, find its minimum fitness function in history by comparing $J^j_{\min-fuel}$ and
   $J^j_{\min-time}$ with its historical best value, i.e. $p_{best}$ . Determine the global minimum fitness by
   comparing it with $g_{best}$ which denotes the global historical best solution.
5. End for
```

```
Obtain the minimum fitness $J_{\min-fuel}^{gbest}$ and $J_{\min-time}^{gbest}$ globally.
For $i = 1, 2, \cdots, M$ do
    Update $\vec{v}$ and $\vec{x}$ a for $i$ th particle according to Eq. 21 which will be inputs in next iteration
End for
While $j$ is still below maximum iteration $G$
```

4.3 Improvement of PSO

Premature convergence exists in heuristic methods commonly [27] . Topological improvement is surveyed in this section. The basic idea underlying the improvement is to increase particles' diversity thus avoiding falling into local optimum exploration.

Traditional PSO (t-PSO) algorithm conducting in an intuitive and simple way where each particle utilizes the prior best solution itself p_{best} and the other particle with global best solution g_{best} actually wastes information of those particles with relative better solution. These particles represent different direction of exploration in search space which benefits the global exploration while being neglected in t-PSO often leading to stagnation in local optimum. A strong and efficient way to avoid premature convergence is increasing particles' diversity, which can be accomplished by learning from more particles.

The origin of PSO mimicking the process of searching for food of bird swarm indicates the difficulty for each particle to always follow the best particle. Note that in natural life, bird swarm's action is influenced by neighbors in a more direct way. However, this natural process wasn't considered in t-PSO model.

A considerable amount of references developing neighborhood topology are presented in Ref. [28] to utilize the neighbor particle information. Obviously, learning from all particles is unnecessary and will cost too much computational load. The distinct approach is to learn from each particle's cluster corresponding to the proverb that birds of feather flock together. Clustering algorithm, used to determine several categories for particles via particles' solution property, i.e. position vector, will be discussed in this section. Once the cluster is determined, it's easy for each particle to learn from best particle n_{best} from the cluster by adding term to Eq. 21 as follows.

$$\begin{cases} v_{i,j}^{t+1} = wv_{i,j}^{t} + c_1 r_1 (p_{best}^{t}(i,j) - x_{i,j}^{t}) + c_2 r_2 (g_{best}^{t}(i,j) - x_{i,j}^{t}) \\ \qquad + c_3 r_3 (n_{best}^{t}(i,j) - x_{i,j}^{t}) \\ x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t} \end{cases} , i = 1, 2, \cdots, M, \ j = 1, 2, \cdots, D \tag{32}$$

where c_3 is the cluster component depicting the influence of the cluster, r_3 is the random number uniformly distributing on the interval between -1 and 1.

In order to find n_{best}, K-means algorithm [30] is used in this paper to partition particles into several clusters for iterations. The reasons for choosing k-means are its simplicity and efficiency. For each particle, neighbor component n_{best} will be determined by specific particle with best solution within the cluster. In most situations, these two particles are not identical. This algorithm is performed to partition M particles in D dimensions into k clusters $\{C_1, C_2, \cdots, C_k\}$ .To keep the consistency and continuity with PSO algorithm as described before; pseudo-code for k-means method is shown in

Table 3.

Table 3 K-means algorithm

```
Inputs: $M$ particles' position during $t$ th iteration, $D=\{\vec{x}_1,\vec{x}_2,\cdots,\vec{x}_M\}$ $k$ cluster centers.
Process:
Initially, choose $k$ particles as mean-value vector randomly $\{\mu_1,\mu_2,\cdots,\mu_k\}$ and clustering $C_i=\varnothing\ (1\le i\le \mathrm{k})$
Repeat
    For $j=1,2,\cdots,M$ do
        Calculate the distance between particle $\vec{x}_j$ and every mean value vector $\mu_i$:
            $d_{ji}=\|x_j-\mu_i\|_2$; particle $\vec{x}_j$ will be divided into the cluster where the $d_{ji}$ is the closest one: $C_i=C_i\bigcup\{\vec{x}_j\}$
    End
    For $i=1,2,\cdots,k$ do
            Update the mean value vector: $\mu_i^*=\frac{1}{|C_i|}\sum_{\vec{x}\in C_i}\vec{x}$
            If $\mu_i^*\neq\mu_i$ then $\mu_i=\mu_i^*$
            Else $\mu_i$ won't be changed
          End if
    End for
Until all $\mu_i(i=1,2,\cdots,k)$ do not need to be changed
Output: clusters $\{C_1,C_2,\cdots,C_k\}$
```

For each particle $x_j\in C_i$, it's easy to find the n_{best} particle with the best fitness function within C_i, which will be used in Eq. 26 to update particles' velocity vector. During iteration, k-means algorithm will be implemented to find n_{best} and calculate Eq.26 until all calculations of PSO algorithm are done. Note that, the clusters result will be changed every time following the iteration of particles. Details of each particle's n_{best} will be presented in the remainder of this paper.

4.4 Summary of Improved PSO (IPSO)

Based on improvements of neighbor component n_{best} and acceleration coefficients above, the summary pseudo-code for IPSO is shown in table 4. Since the main contents are identical to PSO, with an exception for line 8 in Table 2, unnecessary lines are hided in Table 4.

Table 4 Summary algorithm flow of Improved PSO

```
For $M$ particles:
    Determine initial value of $\vec{x}$ and $\vec{v}$
end
1. Repeat for $j$ th iteration
2. (Identical lines with that in Table 2)
3. Obtain the minimum fitness $J_{\min-fuel}^{gbest}$ and $J_{\min-time}^{gbest}$ globally.
4. For $i=1,2,\cdots,M$ do
5. Update acceleration coefficients $c_1,c_2,c_3$ according to Eq. (27), which will be used to
   calculate $\vec{v}$ and $\vec{x}$ for $i$ th particle according to Eq. (26).
```

```
6.   End for
7.   While $j$ is still below maximum iteration $G$
```

5. Verification and validation

In order to check the validity of the improved PSO algorithm, results of ascent trajectory optimization of GHAME are compared with that in reference. Accordingly, initial and terminal conditions were set as the Ref. [21] : reference area, $S_{ref} = 6000\,ft^2 = 557.42\,\mathrm{m}^2$ and area of intake of the engine $A_C = 300\,ft^2 = 27.87\,\mathrm{m}^2$, initial velocity of 800 m/s, initial altitude of 23000m, initial weight of 12000 kg. Final flight path angle of 3° , final altitude of 32 km. Other regulations of parameters and boundary values are shown in Table 5

Table 5 Parameters constraints and boundary values

Types	Control parameter	State parameters					Time t / s
	α / deg	h / m	γ / deg	v / (m/ s)	l / m	m / kg	
Initial	free	16764	0	1676.4	0	12701	0
Min	-3	0	-90	$v_{\min}$	0	Free	0
Max	21	free	90	$v_{\max}$	free	12701	$t_{\max}$
Final	Free	39624	3	free	free	m_f	t_f

As a verification of proposed IPSO, comparisons of solution results via IPSO and that in Ref. [21] were listed in Table 6 incorporating altitudes, velocities, fuel usage, time consumption etc. Ref. [21] employs the finite difference method to determine the minimum-fuel trajectory of the same vehicle model. British units were used in that reference, and transformations between international units and British units were done in Table 6.

Given the same innitial conditions, final time for IPSO method is 96.6 seconds, while that in the reference determined by finite difference method is 101.3 seconds with relative error of 4.64%. Column corresponding to m_f shows that the objective function run by IPSO performed nearly the same as reference with 0.74% difference. Although trajectory profiles determined by two methods are quite different (trajectory profiles and solutions can be found in Ref. [21]), the main tendency of variebles via two methods show great consistency. Especially, final time needed for optimization with respect to IPSO is smaller than that in reference.

Table 6 Minimum-fuel trajectories comparison and verification

Types	Initial conditions			Terminal conditions		
	h_i	m_i	v_i	h_f	m_f	t_f
IPSO method	16764 m (55000 ft)	127010 kg (280000 ft)	1676.4 m/s (5500 ft/s)	39624 m (130000 ft)	104970 kg (231420 lb)	96.6 s
Reference	55000 ft	280000 ft	5500 ft/s	130000 ft	233138 lb	101.3 s
Relative error	—	—	—	—	0.74%	4.64%

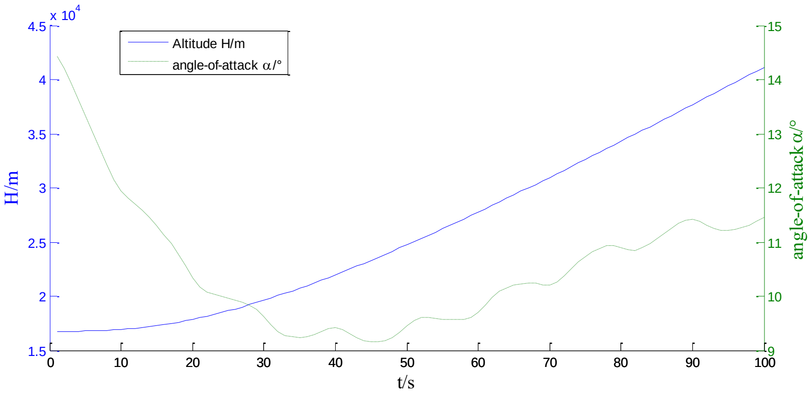
100 particles are used to determine optimal trajectories via 200 iterations. Best solution of the g_{best} particle is:

$$\vec{x}_{gbest} = \{\text{-2.5397e-05}, 0.0053, -0.3042, 14.6462, -0.0217,\ \ -1.1344,\ 0.1472, 0.5118\} \tag{33}$$

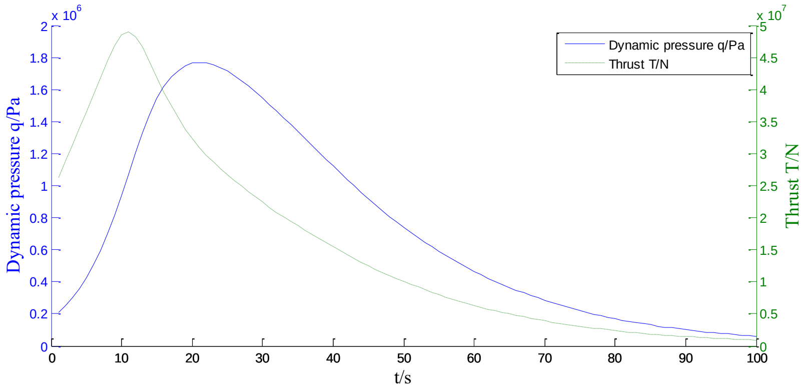
Based on Eq. 33, $\alpha(t)$ and trajectory profile are presented in Fig. 5 (a). Fig.5 (b) provides profiles of thrust of engine and dynamic pressure. Weights and speeds are drawn in the Fig. 5(c).

Duiring the early stage, angle-of-attack is relatively larger than subsequent phases shown in Fig.5 (a), which is because the current low thrust shown in Fig.5 (b) and current low lift force corresponding to low dynamic pressure are not able to overcome great gravity to climb out. The initial phase of climbing out could sustain for a long time to accelerate speed untill that is large enough to provide sufficient thrust and lift force to overcome drag force. Moreover, lower altitude allows vehicle to accelerate faster due to dense atmosphere. Thus, this phase has to proceed almost horizontally.

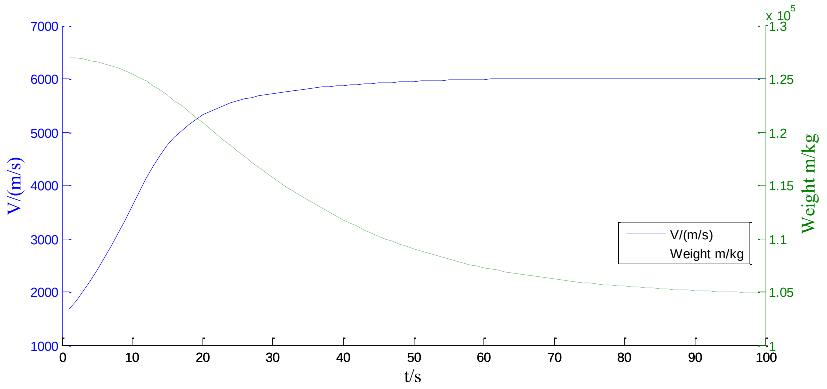
After early stage of ascent trajectory, speed accelerates dramatically as shown in Fig.5 (c). As the speed accelerates, lift force and thrust increase, thus smaller angle-of-attack would be needed to climb out, which benefiting acceleration for speed. Increase of speed and decrease of angle-of-attack during climb-out phase are consequenses of how vehicle’s propulsion system matches trajectory. After this phase, enough speed gained to support pulling up of vehicle thus no need for speed to accelerate drammatically, and this tensency is shown in Fig.5 (c). Decrease of density caused by increase of altitude and slow change of speed result in decrease of dynamic pressure and thrust, which are presented in Fig.5 (b). Note that this tendency is consistent with that in Fig.1.



(a) Altitude and angle-of-attack



(b) Dynamic pressure and thrust



(c) Velocity and weight

Fig. 5 Variables profiles of optimization trajectory

Superiority of IPSO is demonstrated by comparison with traditional PSO (t-PSO) and linearly decreasing inertial weight PSO (w-PSO)[32] showed in Fig. 6. In the very beginning of the algorithms, both methods reached nearly the same fitness value. However, IPSO determined better solution fastly and converged to smaller fitness value than t-PSO and w-PSO in the subsequent search stages. During the entire implementation process of IPSO, profile of fitness value decreases uniformly, and to this point, performance of exploration is quite stable. Note that, t-PSO method falled into local optimum during the iterations from 20 to 50 resulting in gentle change of fitness value. In each iteration, particles can learn to its best “neighbor” without confining to global best partile. This approach strongly increases particles’ diversity to determine best solution avoiding premature.

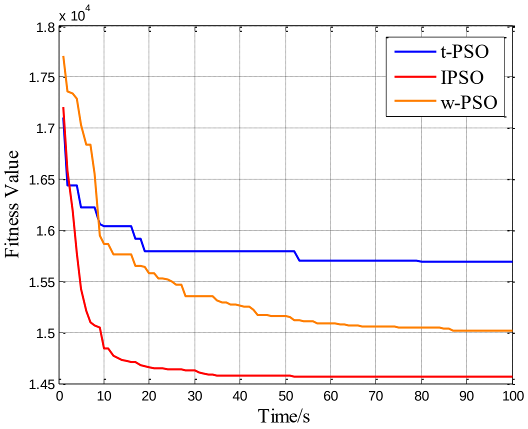


Fig.6 Fitness value comparion of IPSO and t-PSO

Further more, swarm’s diversity can be dipicted as:

$$\begin{cases} Div^t = \dfrac{1}{M}\sum_{i=1}^{M}\sqrt{\sum_{j=1}^{D}(x_{i,j}^t - \overline{x}_j^t)^2} \\ \overline{x}_j^t = \dfrac{1}{M}\sum_{i=1}^{M} x_{i,j}^t \end{cases}$$

High diversity means dispersive particle distributed over the whole search space. Figure 7 shows three PSO variants’ diversity performance. In the beginning of algorithm, particle’s position are initialised randomly, and thus three PSO methods are all at high level in the beginning. When premature accurs, particles tend to explore better solutions around one local optimum leading swarms’s diversity becoming small. After the beginning stage, diversity of IPSO algorithm is greater than t-PSO and w-PSO. Hence, IPSO algorithm is more convincible to avoid premature.

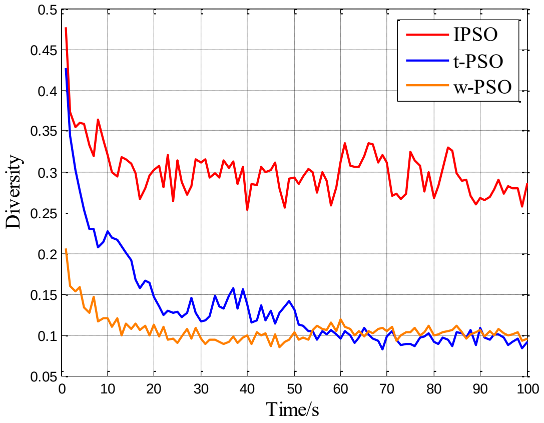


Fig. 7 Three kinds of PSO methods' diversity

6 Simulation results

This section provides several simulations of minimum-fuel trajectories and minimum-time-climb trajectories to gain some insights into the mechanism between propulsion and optimal trajectories for hypersonic vehicle.

6.1 Simulation with varied initial altitudes

Initial weight and final altitude are held constant as previous. While initial altitudes are varied in the first set of solutions with constant initial velocity of 2500 m/s. Table 7 provides 4 different initial altitudes for each run, as well as fuel usage and final time for both minimum-fuel and minimum time-to-climb trajectories.

Table 7 Initial altitude comparison with constant $v_i = 2500\,\mathrm{m/s}$, $h_f = 40\,\mathrm{km}$

Initial altitude	Minimum-fuel		Minimum time-to-climb	
	m_f ,kg	t_f ,s	m_f ,kg	t_f ,s
20 km	108420	133	107560	128
22 km	112570	150	111790	140
25 km	116760	154	116550	142
27 km	118120	174	118110	160

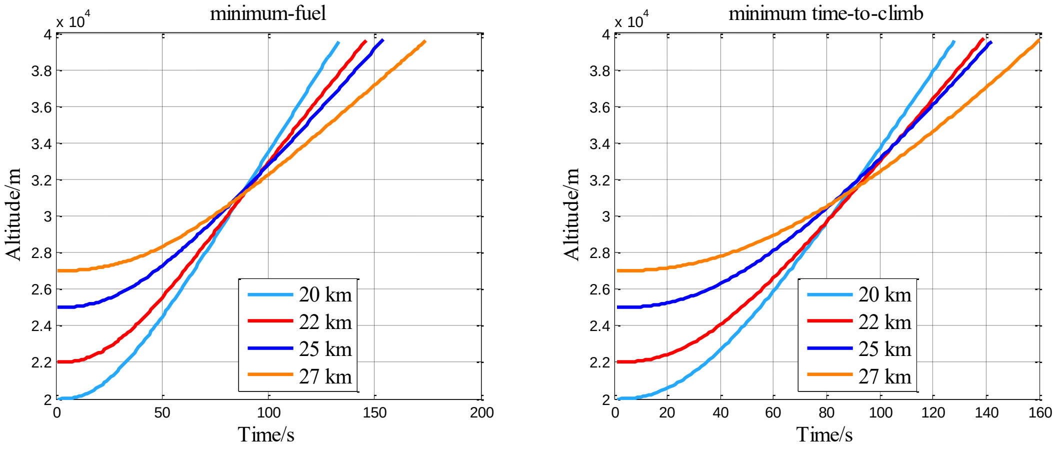


Fig.8 Flight altitude

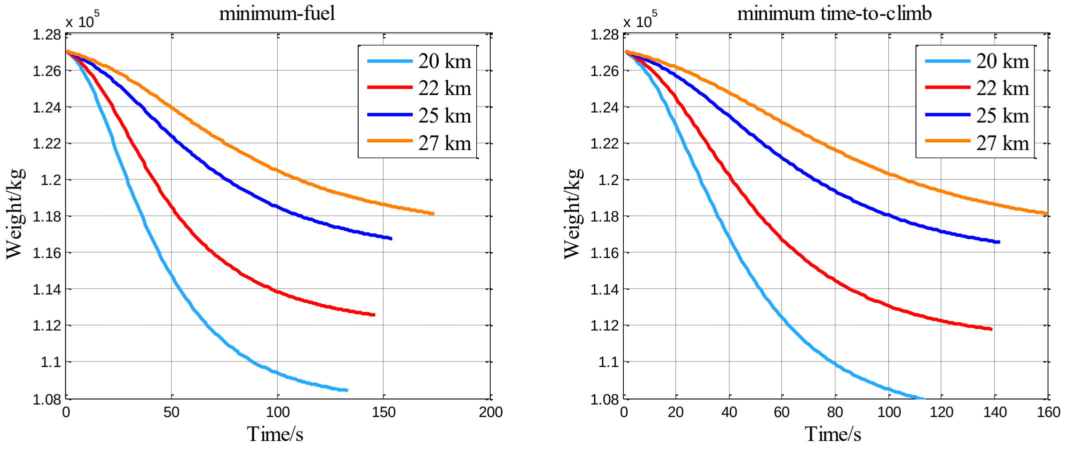


Fig.9 Weight of vehicle

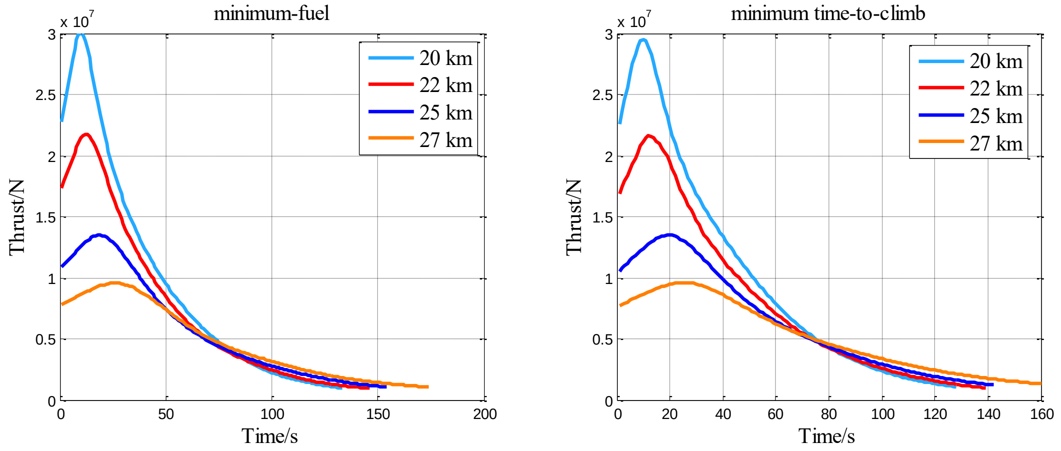


Fig.10 Thrust

It can be concluded from Table 7 that as the altitudes increase, fuel usage decreases (i.e. m_f increases) while final time increases for both kinds of optimum trajectories. The explanation can be induced from Fig. 8 to 10. Lower altitude means dense air to provide larger thrust, shown in Fig.10, resulting in greater acceleration for vehicle to speed, and thus cutting down the final time. Note that mass flow is associated with thrust in Eq. (5), and as a consequence, higher altitudes with smaller thrust result in smaller mass flow，i.e. greater m_f as shown in Fig.9. The main tendency for both two kinds of objective trajectories is regular. Along the minimum-fuel trajectory, minimum fuel usage is reached at the cost of bigger amount of flight time. Similarly, minimum final time is determined at the cost of more fuel used.

6.2 Simulation with varied initial velocity

The second set of solution is determined when initial velocity is varied and other initial conditions are held constant: initial altitude $h_i = 25\,\text{km}$ shown in Table 8. The simulation results are presented in Fig.11 to Fig. 13.

Table 8 Initial velocity comparison with constant $h_i = 25\,\text{km}$, $h_f = 40\,\text{km}$

Initial velocity ,m/s	Minimum-fuel		Minimum time-to-climb	
	m_f ,kg	t_f ,s	m_f ,kg	t_f ,s
2000	117097	163	117041	148

2500	116760	154	116550	142
2700	116625	151	116347	141
3000	116421	147	116095	138

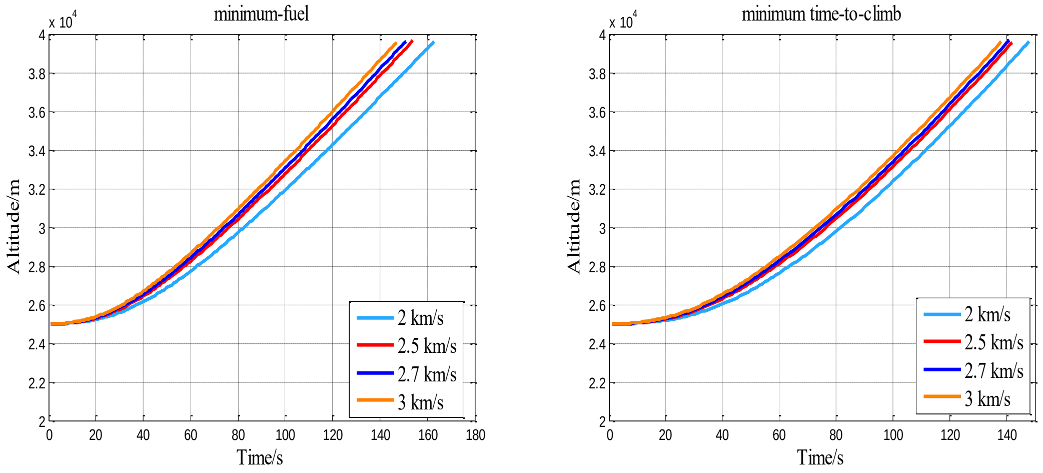


Fig.11 Altitude

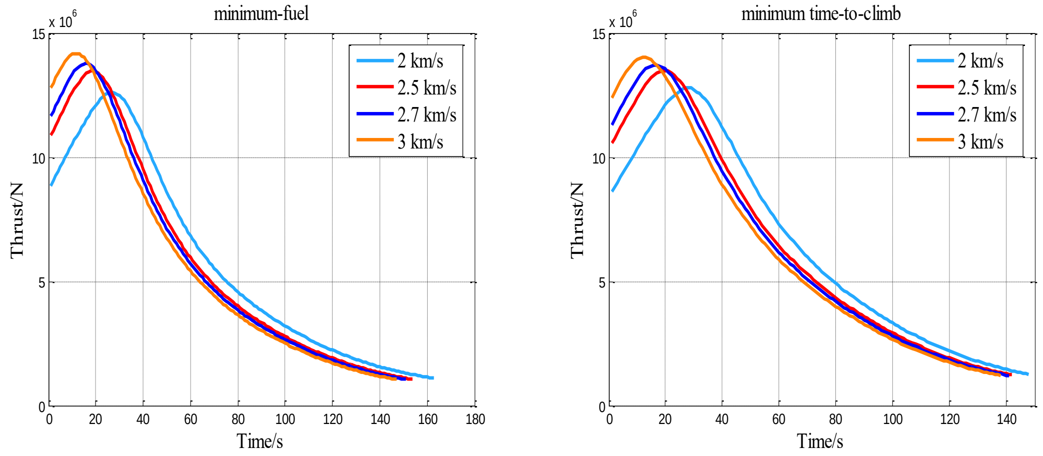


Fig. 12 Thrust

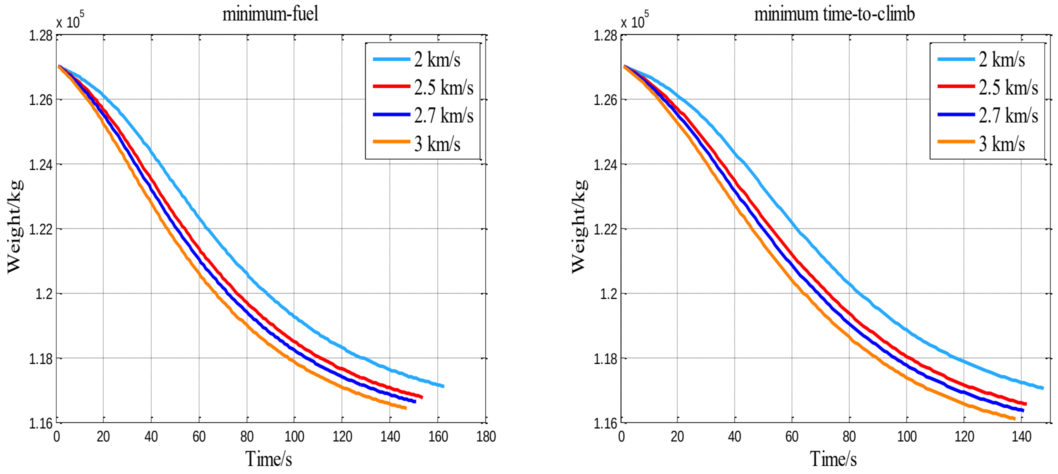


Fig.13 Weight

Results of varied initial velocity with respect to minimum fuel and minimum time-to-climb trajectory are presented in Table 8. As the initial velocity increases, fuel usage increases and the final time decreases. Larger velocity can provide larger thrust as shown in Fig. 12 consequently allowing the vehicle to climb up faster as shown in Fig. 11. Again, since thrust is associated with mass flow, the weight profiles of larger velocity in Fig. 13 will descend faster and lower.

6.3 Simulation with varied initial weight

The conditions of third set of solutions are: varied initial weight of vehicle m_i, initial altitude and velocity are held constant, $v_i = 2500\,\mathrm{m/s}$, $H_i = 25\,\mathrm{km}$, and other terminal conditions are regulated as previous.

Table 9 Initial weight comparison with constant $h_i = 25\,\mathrm{km}$, $h_f = 40\,\mathrm{km}$

Initial weight, kg	Minimum-fuel		Minimum time-to-climb	
	m_f ,kg	t_f ,s	m_f ,kg	t_f ,s
80000	72222	107	71830	105
100000	91169	127	90829	121
120000	110125	147	109865	137
140000	129060	167	109870	152

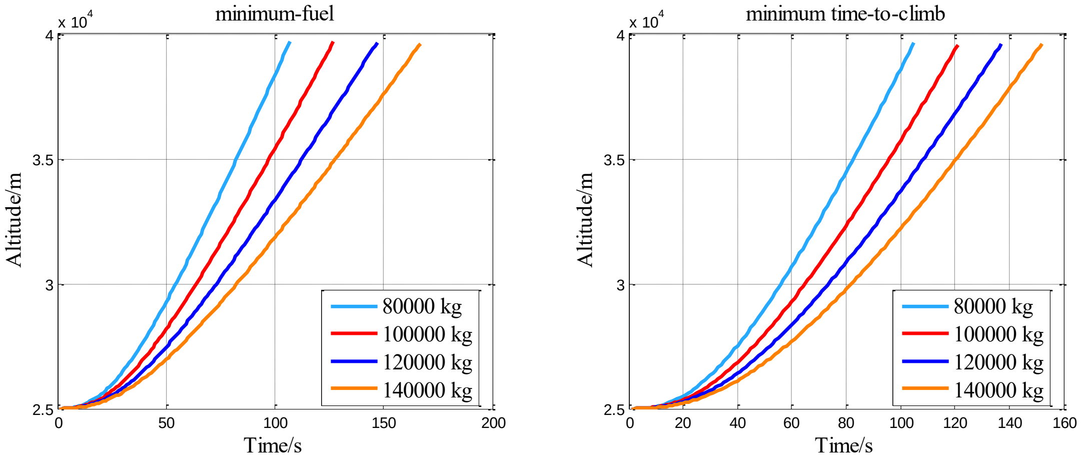


Fig.14 Altitude

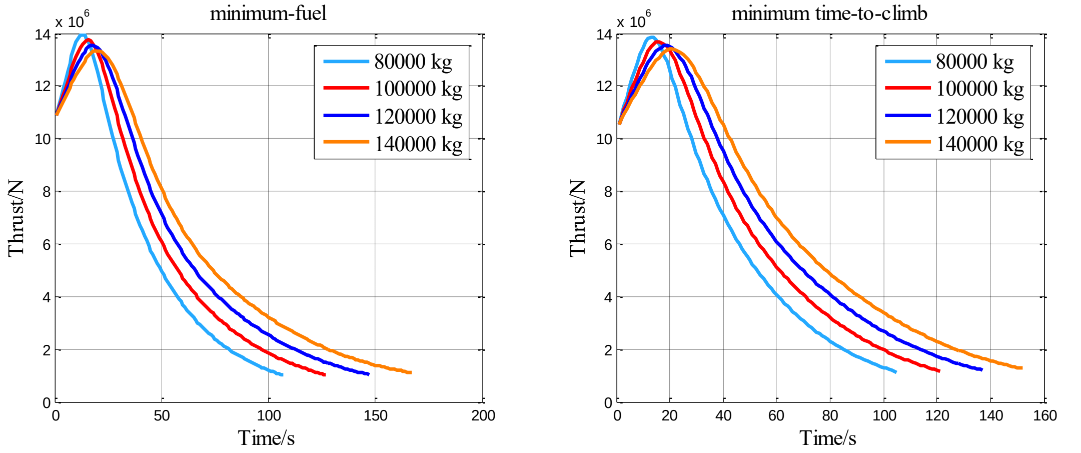


Fig. 15 Thrust

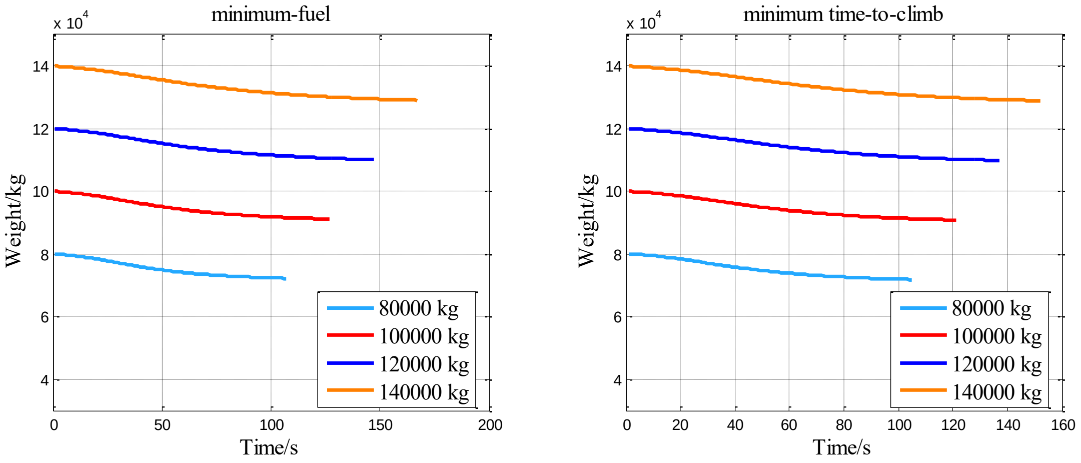


Fig.16 Weight

Optimal solutions with varied initial weight are determined via IPSO. Data in Table 9 indicates that heavier vehicle require more time and more fuel to climb up to the same altitude, which is quite intuitive. Fig.14 shows that, for a lighter vehicle, it's much easier to gain enough acceleration to maneuver to climb up when given the same initial velocity and altitude and as a consequence, saving time and fuel. However, what is counterintuitive is that during the early stage of climbing, thrust of the lighter vehicle is beyond heavier ones, as shown in Fig. 15. In the early stage of climbing, light vehicle will obtain bigger velocity due to larger acceleration, and consequently larger thrust due to bigger velocity. Whereas in the later stage, increase of altitude results in decrease of thrust, which counteracts the influence caused by increase of velocity. Ultimately, thrust begins to decrease after about 20 seconds. What is intuitive is that the weight profiles of the heavier one in Fig. 16 will descend lower, consuming more fuel.

7 Conclusions

This paper investigeted the relationship of trajectory performance and propulsion system of GHAME hypersonic airbreathing vehicle. The kinetic model of 3DOF point-mass system is established. Multi constraints and control parameter are considered in this optimization model, and the optimal objectives are minimum fuel and minimum time-to-climb trajectories. The optimal control problem is transformed into parameters optimization problem through expression for control parameter.

In order to treat shortcoming of traditional PSO, namely premature that leads to local optimum, a novel mechnism based on clustering idea is proposed. Sets of penalty functions are added to handle constraints. The idea underlying IPSO is increasing particles' diversity by learning from neighbors, which has been testified to be valid and efficient.

A trajectory optimization example of identical vehicle model is solved by finite difference method[21] , IPSO and other two PSO variations. Optimal solutions of fuel consumption, velosity and flight time between finite difference method and IPSO are nearly the same. Convergence and diversity of IPSO and other two PSO variations are contrasted, that global optimizing ability and effectiveness of IPSO precedes other PSO variations. Lastly, Three sets of simulations are presented regarding to varied initial altitude, velocity and weight. Numerical simulations demonstrate that the optimization method of IPSO is valid and efficient.

8. Contact Author Email Address

Zhiyu Wang: zhiyu_wang@sina.com

Acknowledgements

This research was supported by "Equipment Pre-Research Fund(Grant61400020303)".

9. Copyright Issues

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

[1] Hicks J W, Trippensee G, NASA Hypersonic X-Plane Flight Development of Technologies and Capabilities for the 21st Century Access to Space, Delta, 1997, pp. 3960.

[2] Paris S W, Joosten B K, Fink L E, Application of an advanced trajectory optimization method to ramjet propelled missiles, Optimal Control Applications and Methods, Vol.1, No.4, 1980, pp. 319-334.

[3] Hargraves C R, Paris S W, Direct trajectory optimization using nonlinear programming and collocation, Journal of guidance, control, and dynamics, Vol.10, No.4, 1987, pp. 338-342.

[4] Calise A J, Corban J E, Flandro G A. Trajectory optimization and guidance law development for national aerospace plane applications, 1988 American Control Conference, 1988, pp.1406-1411.

[5] Corban J E, Calise A J, Flandro G A. Rapid near-optimal aerospace plane trajectory generation and guidance. Journal of guidance, control, and dynamics, Vol.14, No. 6, 1991, pp.1181-1190.

[6] Corban J E, Calise A J, Flandro G A，Trajectory optimization and guidance law development for transatmospheric vehicles, ICCON IEEE International Conference on Control and Applications，1989, pp. 460-465.

[7] Corban J, Calise A, Flandro G. Optimal guidance and propulsion control for transatmospheric vehicles, Guidance, Navigation and Control Conference, 1989, pp.3617.

[8] Lu P. Inverse dynamics approach to trajectory optimization for an aerospace plane, Journal of Guidance, Control, and Dynamics, Vol.16, No.4, 1993, pp. 726-732.

[9] Lu P. Entry guidance and trajectory control for reusable launch vehicle, Journal of Guidance, Control, and Dynamics, Vol.20, No.1, 1997, pp.143-149.

[10] Lu P, Trajectory Optimization for the National Aerospace Plane, 1992.

[11] Benson D A, Huntington G T, Thorvaldsen T P. Direct trajectory optimization and costate estimation via an orthogonal collocation method. Journal of Guidance, Control, and Dynamics, Vol.29, No.6, 2006, pp.1435-1440.

[12] Gong Q, Kang W, Bedrossian N S, Pseudospectral optimal control for military and industrial applications, 2007 46th IEEE Conference on Decision and Control, 2007, pp.4128-4142.

[13] Yang S, Cui T, Hao X. Trajectory optimization for a ramjet-powered vehicle in ascent phase via the Gauss pseudospectral method, Aerospace Science and Technology, 2017, pp.88-95.

[14] Zheng J, Chang J, Yang S. Trajectory optimization for a TBCC-powered supersonic vehicle with transition thrust pinch, Aerospace Science and Technology, 2019, pp. 214-222.

[15] Sun H, Luo S, Sun Q, Trajectory optimization for parafoil delivery system considering complicated dynamic constraints in high-order model, Aerospace Science and Technology, 2020, 98: 105631.

[16] Wang M, Luo J, Walter U, Trajectory planning of free-floating space robot using particle swarm optimization (PSO), Acta Astronautica, 2015, 112: 77-88.

[17] Wang M, Luo J, Yuan J, Coordinated trajectory planning of dual-arm space robot using constrained particle swarm optimization, Acta Astronautica, 2018, 146: 259-272.

[18] Zhou H, Wang X, Cui N. Glide trajectory optimization for hypersonic vehicles via dynamic pressure control, Acta Astronautica, 2019, 164: 376-386.

[19] Zhou H, Wang X, Bai Y, Ascent phase trajectory optimization for vehicle with multi-combined cycle engine based on improved particle swarm optimization, Acta Astronautica, 2017, 140: 156-165.

[20] Lu P, Sun H, Tsai B. Closed-loop endoatmospheric ascent guidance, Journal of Guidance, Control, and Dynamics, Vol.26, No.2, 2003, pp. 283-294.

[21] Murillo O J. A fast ascent trajectory optimization method for hypersonic air-breathing vehicles, 2010.

[22] Eberhart R, Kennedy J, A new optimizer using particle swarm theory, Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995: 39-43.

[23] Kennedy J, Eberhart R, Particle swarm optimization, Proceedings of ICNN'95-International Conference on Neural Networks, 1995, 4: 1942-1948.

[24] Eberhart R C, Shi Y, Comparison between genetic algorithms and particle swarm optimization, International conference on evolutionary programming, 1998, pp.611-616.

[25] Angeline P J, Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences, International Conference on Evolutionary Programming, 1998, pp. 601-610.

[26] Pontani M, Conway B A, Particle swarm optimization applied to space trajectories, Journal of Guidance, Control, and Dynamics, Vol.33, No.5, 2010, pp.1429-1441.

[27] Kennedy J, Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance, Proceedings of the 1999 congress on evolutionary computation-CEC99, 1999, 3: 1931-1938.

[28] Suganthan P N, Particle swarm optimiser with neighbourhood operator, Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, 1999, 3: 1958-1962.

[29] Veeramachaneni K, Peram T, Mohan C, Optimization using particle swarms with near neighbor interactions, Genetic and evolutionary computation conference, 2003, pp.110-121.

[30] Likas A, Vlassis N, Verbeek J J, The global k-means clustering algorithm, Pattern recognition, Vol.36, No. 2, 2003, pp.451-461.

[31] Hartigan J A, Wong M A, Algorithm AS 136: A k-means clustering algorithm, Journal of the Royal Statistical Society, Vol.28, No.1, 1979, pp.100-108.

[32] Wang X, Ma J J, Wang S, Distributed particle swarm optimization and simulated annealing for energy-efficient coverage in wireless sensor networks, Sensors, Vol.7, No.5, 2007, pp.628-648.