Finding multiple local solutions to optimal control problems via saddle points and its application to the ascent trajectory of a winged rocket

Takahiro Fujikawa and Koichi Yonemoto

Department of Mechanical Engineering, Tokyo University of Science, Noda, Japan

ABSTRACT

Complex nonlinear optimal control problems can have more than a single local solution due to the lack of convexity, and widely-used gradient-based approaches for optimal control have a risk of producing an inferior local solution. Although global optimization methods for optimal control may overcome this difficulty, such existing methods suffer from high computational complexities. This paper presents a novel numerical algorithm for multi-modal optimal control problems based on a saddle-point search method. Starting from an already-found local solution, a saddle-point search algorithm finds a surrounding saddle point of the constrained optimization problem, from which a new local solution is obtained by applying a gradient-based optimizer. By repeating this procedure, multiple locally optimal solutions are explored exhaustively and systematically. Although the developed method is not rigorously guaranteed to produce the global solution, it is applicable to large-scale general optimal control problems with equality and inequality constraints, which makes it a practically beneficial technique. After numerical experiments with an example problem, the ascent trajectory optimization problem of a winged rocket for maximizing apogee altitude is solved by using the proposed method.

1. Introduction

Optimal operation of dynamic systems governed by differential equations is desired in diverse fields of engineering and natural/social sciences. This kind of problems has been actively studied under the category of optimal control, or equivalently trajectory optimization, and many matured numerical methods are available now [1]. State-of-the-art techniques for optimal control are categorized into direct methods [2] and indirect methods [3]. Direct methods transcribe the continuous-time optimal control problem into a nonlinear programming (NLP) problem as a finite-dimensional approximation, and the NLP problem is usually solved by a gradient-based optimizer. Direct methods are further divided into direct collocation methods [2] that discretize both state and control variables, and direct shooting methods [4] that parameterize control variables only. In the indirect methods, on the other hand, a two-point boundary value problem is derived as the optimality conditions of the optimal control problem, and its solution is explored with a gradient-based root finding technique. Since the above methods adopt some sort of gradient-based numerical technique, they give a locally optimal solution near a specified initial guess. Complex nonlinear problems, on the other hand, may have more than a single local solution due to the lack of convexity. Such multi-modal optimal control problems with major practical interests include (1) path planning of robot manipulators [5], (2) obstacle-avoidance control [6], and (3) low-thrust orbit transfer [7]. When objective functions corresponding to local solutions are significantly different, a solution obtained by gradient-based methods can have a risk of being trapped in a quite inferior solution. In addition, if the collection of locally optimal solutions can be revealed instead of a single solution, it brings diverse design options, from which users can make final decisions.

This has motivated recent research activities of applying global optimization techniques to the numerical solution of optimal control problems. They include (1) branch-and-bound [8] or branch-and-lift [9] algorithms, (2) semidefinite relaxation [10], and (3) evolutionary computations [11–13]. Whereas the above first and second methods may be successfully applied to simple problems, the scalability to complex and larger-scale problems is limited due to high computational complexity. A concern on evolutionary computations is the high computation cost, especially when multiple equality terminal conditions are involved, as observed in [12,13].

Under the above-mentioned background, the authors have recently focused on the combination of a direct collocation method and a numerical algorithm for locating saddle points of functions. Saddle-point search methods are studied mainly in computational
physics, in order to identify transition states of a molecular system by searching saddle points of its energy function [14]. At a saddle point with Morse index \( m \), the function becomes maximum along \( m \) orthogonal directions and minimum along the other directions. Figure 1 shows the landscape of Himmelblau’s function, which is a well-known test function for optimization, with its stationary points, as an example. Steepest-descent pathways converge to different local minima depending on the starting points, and it defines the basins of attraction shown in Figure 2. Since an index-one saddle point lies at the boundary of two basins of attractions, two neighbouring initial points near a saddle point leads to two different local minima. By utilizing this property, the systematic exploration of local solutions is realized in the following steps: (1) The NLP problem arising from an optimal control problem through the direct collocation method is solved by a conventional gradient-based optimizer, and a local minimum is obtained. (2) Surrounding index-one saddle points are found by using a saddle-point search method. (3) New local minima are acquired by running gradient-based optimization with new initial guesses chosen properly around the found saddle points. (4) By repeating this procedure until no additional local minimum is found, it is expected that local optimal solutions are exhaustively obtained.

In contrast to energy functions of molecular systems, NLP problems derived from optimal control problems have constraints that originate from dynamic, path, and boundary constraints. An enhanced saddle-point search algorithm for equality-constrained systems is developed in the previous research by the authors [15], based on an iterative minimization formulation proposed by Gao et al. [16,17]. Its limitation is that it cannot handle inequality path and boundary constraints that are common in practical optimal control problems. To overcome this issue, in this paper, the saddle-point search algorithm developed in [15] is extended so that the inequality constraints can be considered. In addition, its application to the ascent trajectory optimization of a winged rocket is performed.

The remainder of this paper is organized as follows: Section 2 describes the transcription of an optimal control problem into an NLP problem. Then, the procedures for locating saddle points around the current local minimum are explained in Section 3. Section 4 presents an algorithm for successively finding multiple local minima via saddle points. Validity of the developed method will be demonstrated via solving an illustrative example problem in Section 5. In Section 6, the ascent trajectory optimization problem of a winged rocket is solved as a practical application. Finally, Section 7 provides concluding remarks.

2. Transcription into static optimization problem

Let us consider the following general Bolza-form optimal control problem:

\[
\begin{align*}
\text{find} \quad & x(t), \ u(t), \ t_0, \ t_f, \\
\text{min.} \quad & J = \Phi[x(t_0), t_0, x(t_f), t_f] \\
& + \int_{t_0}^{t_f} g[x(t), u(t), t] \, dt, \\
\text{s.t.} \quad & \frac{dx}{dt} = f[x(t), u(t), t], \\
& C[x(t), u(t), t] = 0, \\
& C_i[x(t), u(t), t] \leq 0, \\
& \Psi[x(t_0), t_0, x(t_f), t_f] = 0, \\
& \Psi_i[x(t_0), t_0, x(t_f), t_f] \leq 0,
\end{align*}
\]  

where \( x(t) \) and \( u(t) \) are state and control variable vectors, respectively. Independent variable, \( t \in [t_0, t_f] \), typically denotes the time. It is worth noting that \( t_0 \) and
where \( t_f \) are free in this optimal control problem, while they can be fixed to specified values by using Equation (6) if necessary. One of the advantages of the proposed method is the applicability to a wide range of optimal control problems including free final time problems.

In order to handle inequality path and boundary constraints, the augmented objective function \( \tilde{J} \) is introduced as follows:

\[
\tilde{J} = J + w_c^T \int_{t_0}^{t_f} P(C_t) \, dt + w_{\Psi}^T P(\Psi_t),
\]

where \( P(\cdot) \) is a penalty function that gives large value for positive argument and is approximately zero for negative argument. \( w_c \) and \( w_{\Psi} \) are the penalty weights for the inequality path and boundary constraints, respectively. In this paper, the following hyperbolic smooth penalty function presented in [18] is used:

\[
P(y) = \lambda y + \sqrt{\lambda^2 y^2 + \tau^2},
\]

where the parameter \( \tau \) controls the smoothness, and the parameter \( \lambda \) is related to the intensity of the penalty.

Using the Legendre–Gauss (LG) pseudospectral method [19], which is one of the most efficient direct collocation methods, the optimal control problem given by Equations (1)–(7) can be transcribed into the following NLP problem by collocating it at \( N \) LG nodes:

\[
\begin{align*}
\text{find} & \quad \{X_i\}_{i=1}^N, X_0, X_f, \{U_i\}_{i=1}^N, f_0, t_f, \\
\text{min.} & \quad \Phi(X_0, t_0, X_f, t_f) \\
& \quad + \frac{t_f - t_0}{2} \sum_{k=1}^N w_k g(X_k, U_k, t_k), \\
\text{s.t.} & \quad \sum_{k=1}^N D_{i,k+1} X_k - \frac{t_f - t_0}{2} f(X_i, U_i, t_i) = 0, \\
& \quad i = 1 \ldots, N, \\
& \quad X_f - X_0 - \sum_{i=0}^N X_i \left( \sum_{j=0}^N w_j D_{i,j+1} \right) = 0, \\
& \quad C_i(X_i, U_i, t_i) = 0, \quad i = 1 \ldots, N, \\
& \quad C_f(X_f, U_f, t_f) \leq 0, \quad i = 1 \ldots, N, \\
& \quad \Psi_i(X_0, t_0, X_f, t_f) \leq 0,
\end{align*}
\]

where \( w_k \) is the \( k \)th LG weight, \( D_{i,k+1} \) is the \((i, k + 1)\)th entry of the LG pseudospectral differentiation matrix, and \( t_k \) represents the independent variable at the \( k \)th LG node. \( X_i \) and \( U_k \) are the discretized state and control variables at \( t_k \), respectively. \( X_0 \) and \( X_f \) denote the discretized state variables at \( t_0 \) and \( t_f \), respectively. The augmented objective function \( \tilde{J} \) is transformed into the following form:

\[
\tilde{J} \approx \Phi(X_0, t_0, X_f, t_f) + w_c^T P(\Psi_t) + \frac{t_f - t_0}{2} \sum_{k=1}^N w_k [g(X_k, U_k, t_k) + w_{\Psi}^T P(C_t)].
\]

It is noted that any direct collocation method (e.g. Hermite–Simpson method) can be employed instead of the LG pseudospectral method. Let this NLP problem of Equations (12)–(19) be rearranged for the ease of visibility as follows:

\[
z^* = \arg \min_{z \in \mathbb{R}^d} \left\{ V(z) \mid c(z) = 0 \in \mathbb{R}^p, c_i(z) \leq 0 \right\},
\]

where \( z \) indicates all the variables of the NLP problem shown in Equation (12). The transformed augmented objective function, Equation (20), is written in short by \( \tilde{V}(z) \).

The problem must be scaled so that the variables approximately have the range between zero and one, and that the constraint functions roughly have the magnitude of one. This makes the values of tuning parameters recommended in Sections 3 and 4 properly applicable to various problems in a robust manner. Refer to [2, Section 4.8] for a scaling method for NLP problems derived from optimal control problems.

3. Saddle-point search method

An index-one saddle point of a function is defined as a point at which the first derivatives (i.e. gradient vector) are zero and the second derivatives (i.e. Hessian matrix) have one negative eigenvalue. The function takes the maximum value at the saddle point along the direction given by the eigenvector corresponding to the negative eigenvalue. Eigenvector-following methods for saddle-point search are based on this property, and they explore a saddle point by gently ascending a function along the eigenvector corresponding to the minimum Hessian eigenvalue. Among many existing eigenvector-following methods, this paper focuses on the iterative minimization formulation (IMF) [16,17], because of its efficiency and applicability to equality-constrained systems. After the outline of the original IMF is described in Section 3.1, its enhancements are presented in Sections 3.2 and 3.3.

3.1. Iterative minimization formulation for index-one saddle points

The original IMF for the systems with equality constraints (without inequality constraints) consists of the following steps: (1) Given the current point, \( z^{(k)} \), the
rotation step calculates the eigenvector, $v^{(k+1)}$, corresponding to the minimum Hessian eigenvalue. (2) The translation step determines the next point, $z^{(k+1)}$, so that it moves towards an index-one saddle point. (3) The process is repeated while increasing the iteration counter, $k$, until $z^{(k)}$ converges to a saddle point.

The rotation step is expressed by the following optimization problem in [16]:

rotation problem :  
$$v^{(k+1)} = \arg\min_{u \in \mathbb{R}^d, \|u\|_1 = 1} \left\{ u^T \nabla^2 V[z^{(k)}] u \ | \ \nabla c[z^{(k)}] = 0 \right\}. \tag{22}$$

Equation (22) is equivalent to a constrained eigenvalue problem, and it can be transformed into an unconstrained problem [20]. Then, it can be solved efficiently without evaluating expensive Hessian $\nabla^2 V[z^{(k)}]$ explicitly, by using an appropriate eigenvalue solver [21]. As a result, $v^{(k+1)}$ gives the eigenvector associated with the minimum eigenvalue, $E(z)$, which is calculated by

$$E[z^{(k)}] = v^{(k+1)T} \nabla^2 V[z^{(k)}] v^{(k+1)}.$$ \tag{23}

The translation step is accomplished by solving the following optimization problem [16]:

translation problem :  
$$z^{(k+1)} = \arg\min_{y \in \mathbb{R}^d} \left\{ L[y; z^{(k)}, v^{(k+1)}] \ | \ c(y) = 0 \right\}, \tag{24}$$

where $L[y; z, v]$ is an auxiliary objective function constructed so that $z^{(k)}$ converges to a saddle point as $k$ increases. Since the required solution of Equation (24) is the local one, it must be solved by a gradient-based nonlinear optimizer with the initial guess given by $y = z^{(k)}$. In addition, the solution is sought only within a local neighbourhood of $z^{(k)}$ denoted by $\mathcal{U}[z^{(k)}]$, in order to exclude the unintended non-local solutions. One of the possible forms of $L$ discussed in [16,17] is

$$L[y; z, v] = -V(y) + 2V(\xi[y - vv^T(y-z)]). \tag{25}$$

Here, $\xi(y) : \mathbb{R}^d \to \mathbb{R}^d$ is the operator that projects a point, $y$, onto geodesics curves whose tangents are in the tangent space of $c(z)$ and orthogonal to $v$. For general nonlinear constraints, such an operator cannot be strictly written in a closed form, and this issue is left unsolved in [16,17]. The descent direction of $L$ in Equation (25) corresponds to the gentlest descent direction of $V$, and the solution of Equation (24) yields a new point closer to a saddle point than the old point. In this paper, let the local neighbourhood, $\mathcal{U}[z^{(k)}]$, be defined by

$$\mathcal{U}[z^{(k)}] = \left\{ y \in \mathbb{R}^d \ | \ \|y - z^{(k)}\|_\infty \leq r_{\text{nei}} \right\}, \tag{26}$$

which can be expressed as simple upper and lower bounds on each component of $y$.

Objective gradient along the tangent space of $c(z)$ must vanish at a saddle point, and the residual of this constrained stationary condition, $R^{(k)}$, is defined as

$$R^{(k)} = \|P[z^{(k)}] \nabla V[z^{(k)}]\|_2. \tag{27}$$

$P[z^{(k)}]$ is the orthogonal projection operator on the tangent space of $c(z)$ at $z^{(k)}$, which is calculated by

$$P[z^{(k)}] = I - \nabla c[z^{(k)}] \begin{bmatrix} \nabla c[z^{(k)}]^T \nabla c[z^{(k)}] \end{bmatrix}^{-1} \nabla c[z^{(k)}]^T. \tag{28}$$

IMF procedure is terminated when $z^{(k)}$ sufficiently converges to a saddle point, and its criterion with the tolerance value, $\epsilon_{\text{tol}}$, can be written as follows:

$$R^{(k)} \leq \epsilon_{\text{tol}}. \tag{29}$$

Under some conditions, $z^{(k)}$ obtained via IMF is guaranteed to locally converge to an index-one saddle point (see Theorem 3.1 of [16] or Theorem 1 of [17]). When $z^{(k)}$ is not near a saddle point and the corresponding minimum eigenvalue, $E[z^{(k)}]$, becomes positive, on the other hand, the auxiliary objective, $L$, has no lower bound locally [16,17]. Therefore, it is recommended that (1) an initial guess of $z$ is chosen so that the associated minimum eigenvalue is negative, and (2) the minimum eigenvalues at $z^{(k)}$’s during the IMF iterations are kept negative. This idea motivates the development of globalization strategies for IMF introduced in next subsections.

### 3.2. Modified iterative minimization formulation

In this subsection, four modifications to the IMF are proposed to improve its constraint handling and global convergence.

As the first modification, the rotation problem is revised for the effective handling of nonlinear constraints. The modified rotation problem is formulated as follows by using Riemannian Hessian containing second-derivative information of constraints [22], instead of using the combination of usual Hessian and linearized constraints:

modified rotation problem :  
$$v^{(k+1)} = \arg\min_{u \in \mathbb{R}^d, \|u\|_1 = 1} \left\{ \nabla^2 V[z^{(k)}] u, u \right\}. \tag{30}$$

where $\langle \cdot, \cdot \rangle$ means an inner product. It is noted that the modified rotation problem is formulated with the augmented objective function, $\tilde{V}$, containing the penalty term for inequality constraints. As a result, inequality constraints that are near active (i.e. $c_i[z^{(k)}] \approx 0$) have influences on the resulting $v^{(k+1)}$. $\nabla^2 \tilde{V}(z^{(k)})[u]$ is the
Riemannian Hessian defined by

\[ \text{Hess} \tilde{V}(z^{(k)}) [u] = P[z^{(k)}] \nabla^2 \tilde{V}(z^{(k)}) P[z^{(k)}] u - P[z^{(k)}] \nabla^2 c(z^{(k)}) P[z^{(k)}] u \]

\[ \approx \nabla c(z^{(k)})^T \nabla V(z^{(k)}) \left\{ \nabla c(z^{(k)})^T \nabla V(z^{(k)}) \right\}^{-1} \nabla c(z^{(k)})^T \nabla \tilde{V}(z^{(k)}). \]  

(31)

In spite of its lengthy expression, this modified rotation problem can still be handled efficiently by an eigenvalue solver [21]. Here, the explicit evaluations of second derivatives (i.e. \( \nabla^2 \tilde{V}(z^{(k)}) \) and \( \nabla^2 c(z^{(k)}) \)) can be avoided by exploiting the following finite-difference approximations:

\[ \nabla^2 \tilde{V}(z) P(z) u \approx \frac{1}{2\delta} \left\{ \nabla \tilde{V}(z + \delta P(z) u) - \nabla \tilde{V}(z - \delta P(z) u) \right\}, \]

(32)

\[ \nabla^2 c(z) P(z) u \approx \frac{1}{2\delta} \left\{ \nabla c(z + \delta P(z) u) - \nabla c(z - \delta P(z) u) \right\}, \]  

(33)

where \( \delta \) is a step size. After Equation (30) is solved, the minimum eigenvalue of the Riemannian Hessian, \( E(z) \), at \( z^{(k)} \) is calculated by

\[ E(z^{(k)}; v^{(k+1)}) = \left( \text{Hess} \tilde{V}(z^{(k)}) \right) [v^{(k+1)}], v^{(k+1)} \].  

(34)

and the corresponding eigenvector is given by \( v^{(k+1)} \).

The second modification is made on the translation problem as follows:

modified translation problem : \( z^{(k+1)} \)

\[ = \arg \min_{y \in \mathbb{R}^d} \left\{ L[y; z^{(k)}, v^{(k+1)}] \mid c(y) = 0, \right\} \]

\[ c_i(y) \leq 0, E_i(y; z^{(k)}, v^{(k+1)}) \leq \epsilon_{\text{eig}} \}. \]  

(35)

Equation (36) can be evaluated via finite differencing Equation (34), before starting the solution of the modified translation problem.

The modified translation problem is different from the original NLP problem derived from the optimal control problem, Equation (21), in that (1) the objective function, \( V \), is replaced by the auxiliary function, \( L \), and (2) a linear constraint related to the minimum eigenvalue is introduced. This means that the translation problems have the same numbers of variables and only one more linear constraint, compared to the original NLP problem. Although the computational complexity of \( L \) is higher than \( V \), its effect on the overall computational burden is minor, since the computational complexity of NLP for handling \( c(y) = 0 \) and \( c_i(y) \leq 0 \) is dominant. It is noted that the dimensions of \( c(y) \) and \( c_i(y) \) are typically more than dozens in practical problems (see Section 6). Therefore, the computational burden of the translation problems is comparable to that of the original NLP problem.

The third modification is the implementation of a trust-region algorithm for adjusting \( r_{\text{nei}} \) and \( \epsilon_{\text{eig}} \) as shown in Algorithm 1. \( \epsilon_{\text{eig}} \) and \( \epsilon_{\text{tol}} \) are negative and
positive tuning parameters, respectively, whose absolute values are recommended to be comparable or larger than the optimality and feasibility tolerances of the NLP solver. In this paper, $\varepsilon_{\text{ei}} = -10^{-4}$ and $\varepsilon_{\text{tol}} = 10^{-4}$. $a$, $b_1$, $b_2$, $c_1$, and $c_2$ are selected as follows based on typical values used in trust-region schemes: $a = 2$, $b_1 = 1.5$, $b_2 = 0.5$, $c_1 = 0.5$, and $c_2 = 1.5$. Although these tuning parameters vary significantly, the iteration results are consistent when $\varepsilon_{\text{ei}}$ is of the same order of magnitude.

In this subsection, an iterative procedure for finding such a point from a starting point near a local minimum is constructed. It is noted that, at the local minimum, the Hessian is positive definite, and the minimum eigenvalue is positive. By changing the translation problem and termination condition in Algorithm 1, a new method and Equation (35) is actually its single-iteration version. This operation maps a point to the proximity of the nonlinear manifold defined by the constraints.

### 3.3. Finding initial guess for iterative minimization formulation

As mentioned in Section 3.1, it is preferable that an initial guess of $z$ that initiates Algorithm 1 has a negative minimum eigenvalue. In this subsection, an iterative procedure for finding such a point from a starting point near a local minimum is constructed. It is noted that, at the local minimum, the Hessian is positive definite, and the minimum eigenvalue is positive. By changing the translation problem and termination condition in Algorithm 1, a new algorithm that accomplishes this goal is developed as shown in Algorithm 2. The translation problem below is used instead of Equation (35) during this initialization step:

$$z^{(k+1)} = \arg \min_{y \in \mathcal{L}(z^{(k)})} \left\{ L[y; z^{(k)}], V^{(k+1)} \right\},$$

where $L'$ is a new auxiliary objective function defined by

$$L'[y; z, v] = -V(y) + V[\xi [y - vv^T (y - z)]] \right\}.$$

This means that the iterations stop when the minimum eigenvalue at the current point, $z^{(k)}$, is substantially smaller than zero. It is not guaranteed that the termination condition is satisfied within a finite number of iterations by using Algorithm 2. The development of a method with the proved termination is the scope for future research.

### 4. Successive search of local optimal solutions

As mentioned in Section 1, an index-one saddle point is located on the boundary of two basins of attractions where steepest descent leads to different local minima. By exploiting this property, it is possible to find multiple local optimal solutions successively and systematically in the following way: (1) A saddle-point

$z^{(k+1)} = \arg \min_{y \in \mathcal{L}(z^{(k)})} \left\{ L'[y; z^{(k)}], V^{(k+1)} \right\},$
solution around the current minimum is searched by using Algorithms 1 and 2. (2) By choosing a proper initial guess near the found saddle point, a new local minimum may be obtained by using a gradient-based optimizer. (3) These two steps are repeated alternately until no new solution is found.

4.1. Initial guess for seeking new local minima

Let us describe how to calculate the proper initial guess near the found saddle point in the second step. Suppose that a saddle point, $\mathbf{z}_{\text{sad}}$, is obtained by running Algorithm 1 from a starting point, $\mathbf{z}_{\text{ini}}$. The Riemannian Hessian of the augmented objective function has one negative eigenvalue at the index-one saddle point. Along the direction defined by the corresponding eigenvector, the augmented objective function has the maximum value at the saddle point. Therefore, the point given by $\mathbf{z}_{\text{sad}} + \delta_2 (\mathbf{z}_{\text{sad}} - \mathbf{z}_{\text{ini}})$ with $\delta_2 > 0$ is in a basin of attraction that is different from the basin which $\mathbf{z}_{\text{ini}}$ belongs to. This means that a new local solution is expected to be acquired by a gradient-based optimizer when the above-mentioned point is used as its initial guess. A concern is that the theoretical steepest-descent pathway and complex gradient-based optimizers may have different basins of attractions [24]. It is noted that the first-order steepest-descent algorithm usually suffers from slow convergence, and it is not applicable to practical large-scale problems. Figure 3 shows the basins of attractions of a sequential quadratic programming (SQP) solver [25] with the default setting for the Himmelblau’s function. When it is compared with Figure 2, the SQP solver produces basins similar to those of the steepest descent around the saddle points, in this specific function. In order to realize robust search of new local solutions for various optimal control problems and gradient-based optimizers, it is recommended to try several initial guesses from different $\delta_2$’s.

4.2. Successive search algorithm

Algorithm 3 shows the whole procedures for the successive search of local optima. $\delta_1$ is a sufficiently small positive number (e.g. $\delta_1 = 10^{-4}$). $\Delta_2$ is the vector of positive numbers, and its dimension, $n$, is the number of trial initial guesses in solving the original NLP problem. In this paper, $\Delta_2 = [10^{-4}, 10^{-3}, 10^{-2}]$. $\delta_{\text{dup}}$ must be substantially larger than the optimality and feasibility tolerances of the NLP solver (e.g. $10^{-2}$). $\mathbf{z}_{\text{sols}}$ and $\mathbf{z}_{\text{sads}}$ are the sets of local optimal solutions and saddle-point solutions, respectively, which are the outputs of this algorithm. $\mathbf{z}_{\text{tmp}}$ indicates the set of local optimal solutions whose nearby saddle points are not explored yet.

In lines 2–4, the original NLP problem is solved with the user-specified initial guess, and the resulting solution is stored in $\mathbf{z}_{\text{sols}}$ and $\mathbf{z}_{\text{tmp}}$. Lines 6–10 represent the core procedures consisting of the saddle-point search and the local-minimum search, and they are repeated until $\mathbf{Z}_{\text{tmp}}$ becomes empty. In line 8, the eigenvector corresponding to the minimum eigenvalue at the current local minimum, $\mathbf{z}_{\text{cur}}$, is calculated and denoted by $\mathbf{v}$. Since $\mathbf{v}$ and $-\mathbf{v}$ correspond to the gentlest ascent directions of the augmented objective function, they are used as promising initial search directions towards surrounding saddle points in lines 9 and 10. After a saddle point is found through lines 16 and 17, the original NLP problem is solved in line 20, with several initial guesses as discussed in Section 4.1. If a newly found local minimum is not the duplication of any solutions obtained so far with the tolerance of $\delta_{\text{dup}}$ (see line 21), it is added to $\mathbf{z}_{\text{sols}}$ and $\mathbf{z}_{\text{tmp}}$. In addition, the associated saddle point is appended to $\mathbf{Z}_{\text{sads}}$. 

Algorithm 2 Initial-guess search for modified IMF.

| Input: $\mathbf{z}_{\text{ini}} \in \mathbb{R}^d, r_{\text{nei0}}$ |
| Output: $\mathbf{z}_{\text{ini}} \in \mathbb{R}^d$ |
| 1: $r_{\text{nei}} \leftarrow r_{\text{nei0}}, \mathbf{z}^{(1)} \leftarrow \mathbf{z}_{\text{ini0}}, k = 1$. |
| 2: while do |
| 3: Solve the modified rotation problem, Equation (30). |
| 4: Solve the translation problem for initialization, Equation (38), with $r_{\text{nei}}$. |
| 5: if Stopping condition, Equation (40), is satisfied. then |
| 6: break |
| 7: end if |
| 8: $k \leftarrow k + 1$ |
| 9: end while |
| 10: $\mathbf{z}_{\text{ini}} \leftarrow \mathbf{z}^{(k)}$ |

Figure 3. Basins of attraction obtained by an SQP solver [25] in Himmelblau’s function.
Algorithm 3 Successive search of local minima.

Input: $z_{ini,u} \in \mathbb{R}^d, \delta_1, \Delta_2 \in \mathbb{R}^n, \epsilon_{dup}$

Output: $Z_{sols}, Z_{sads}$

1: $Z_{sols} \leftarrow \{\}, Z_{tmp} \leftarrow \{\}, Z_{sads} \leftarrow \{\}$
2: Solve original NLP problem, Equation (21), with user-specified initial guess, $z_{ini,u}$, to obtain $z^*$.
3: $Z_{sols} \leftarrow Z_{sols} \cup \{z^*\}$
4: $Z_{tmp} \leftarrow Z_{tmp} \cup \{z^*\}$
5: while do
6: $z_{cur} \in \mathbb{R}^d \leftarrow$ any element of $Z_{tmp}$
7: $Z_{tmp} \leftarrow Z_{tmp} \setminus z_{cur}$
8: Solve modified rotation problem, Equation (30), with $z^{(k)} = z_{cur}$, to obtain $v$.
9: $Z_{sols} \leftarrow Z_{sols} \cup \{z^*\}$
10: $Z_{tmp} \leftarrow Z_{tmp} \cup \{z^*\}$
11: if $Z_{tmp} = \{\}$ then
12: break
13: end if
14: end while

function $\text{FindNewSol}(z_{ini0})$
15: Run Algorithm 2 to obtain $z_{ini}$.
16: Run Algorithm 1 to obtain $z_{sad}$.
17: for $i=1:n$ do
18: $\delta_2 \leftarrow \Delta_2(i)$
19: Solve original NLP problem, Equation (21), with initial guess, $z_{sad} + \delta_2(z_{sad} - z_{ini})$, to obtain $z^*$.
20: if $\exists z \in Z_{sols}$ such that $||z - z^*||_2 \leq \epsilon_{dup}$ then
21: $Z_{sols} \leftarrow Z_{sols} \cup \{z^*\}$
22: $Z_{tmp} \leftarrow Z_{tmp} \cup \{z^*\}$
23: $Z_{sads} \leftarrow Z_{sads} \cup \{z_{sad}\}$
24: end if
25: end for
26: end function

In Sections 5 and 6, the original NLP problems [Equation (21)], the modified translation problems [Equation (35)], and the translation problem during initialization [Equation (38)] are solved by using an SQP solver [25]. The optimality and feasibility tolerances are both $10^{-4}$. In order to calculate gradient information required in the SQP algorithm accurately and to improve numerical stability, complex-step derivative approximation [26] is used instead of usual finite differencing.

### 4.3. Characteristics of algorithm

One of the advantages of the developed successive search method over other global optimization techniques mentioned in Section 1 is its scalability to larger or higher-dimensional problems. When the rigorous global optimality is pursued via branch-and-bound methods, auxiliary problems must be solved repeatedly, and finer discretization of control variables leads to exponential growth of computation time [8]. As for the semidefinite relaxation, the dimension of the relaxed problem increases significantly when the original problem has strong nonlinearity [10]. The proposed method, on the other hand, explores multiple optimal solutions by repeating the local gradient-based searches for saddle points and local minima. As a result, the sets of problems to be solved are (1) the original NLP problem [Equation (21)] arising from the optimal control problem, (2) the modified rotation problems [Equation (30)], and (3) the modified translation problems [Equation (35)]. The rotation problems can be handled by eigenvalue solvers, and they are not computationally expensive. The computational burden of the translation problems is comparable to that of the original NLP problem as explained in Section 3.2. In addition, it is empirically found that the required number of saddle-point search iterations is around 10 regardless of the size of the problem. Therefore, it is expected that the global trajectory optimization by the developed method is feasible, as long as the local solution to the original problem can be obtained in a stable way.

Saddle-point solutions to optimal control problems are just the byproducts of the systematic search of local optimal solutions, in contrast to the saddle-point search in computational physics. It is noted that a saddle-point solution is not usually a desirable design solution, since
it is the most inferior point between two adjacent local optimal solutions. However, saddle points may be useful for discussing the relations between several local optimal solutions, because a saddle point is essentially a branch point between two local minima. Investigations on these issues are the subjects of future research.

5. Numerical example

In order to investigate capability and efficiency of the developed method, the following example problem inspired by [10] is considered:

\[
\begin{align*}
\text{find} & \quad x(t) = x(t), \quad u(t) = u(t), \quad t_0, \quad t_f, \\
\min & \quad J = \int_{t_0}^{t_f} 2\sin[2x(t)] - 0.1x(t)^2 + u(t)^2 \, dt, \\
\text{s.t.} & \quad \frac{dx}{dt} = 5u(t), \\
& \quad [x(t_0) - 0.5, \quad t_0, \quad t_f - 1] = 0, \\
& \quad [u(t) - 10, -u(t) - 10] \leq 0.
\end{align*}
\]

After the transcription into an NLP problem with \( N = 21 \), Algorithm 3 is applied with \( \text{r}_{\text{net}} = 1.0 \). The penalty weight for Equation (45) is \( w_C = 0.001 \). The initial guess \( z_{\text{ini},u} \) corresponds to \( x(t) = 0, \quad u(t) = 0, \quad t_0 = 0, \quad \text{and} \quad t_f = 1 \).

Due to highly nonlinear objective function, seven locally optimal trajectories connected by six saddle-point trajectories are obtained as shown in Figure 4. The local optimal solution obtained first at line 2 of Algorithm 3 is the fourth solution \( (J \approx -1.48) \). Other local optima and saddle points are found under the connectivity that is illustrated by arrows in Figure 4. Such an extensive exploration of local optima reveals that the best locally optimal trajectory is the fourth solution \( (J \approx -1.48) \).

Figure 5 presents the residuals of the constrained stationary condition, \( R(k) \), as the functions of iteration number, \( k \). During the initial-guess search using Algorithm 2, the residual increases since \( z(k) \) moves away from a local minimum. The average value of computational local order of convergence [27], \( \bar{\lambda} \), defined as follows is approximately 1.7:

\[
\bar{\lambda} = \frac{\log|R(k_f)|}{\log|R(k_f-1)|},
\]

where \( k_f \) denotes the iteration counter when the termination condition is satisfied. It is suggested in [16,17] that the original IMF has the quadratic convergence rate when the subproblem is solved exactly, and that it may achieve the superlinear convergence when neither an exact nor high-accuracy solution of the subproblem is pursued. The convergence result obtained in this numerical experiment demonstrates that superlinear convergence rate is successfully attained also in the modified IMF method developed in this paper.

In order to investigate the robustness of the proposed method, a saddle-point search method without a trust-region strategy (i.e. the original IMF [16,17]) is also applied to solve this problem. Then, the fifth saddle point cannot be reached from the fifth local minimum, when the fixed value of \( \text{r}_{\text{net}} = 1.0 \) is used. As a consequence, the sixth and seventh locally optimal trajectories are not found. This shows that the trust-region based adjustment of \( \text{r}_{\text{net}} \) is useful for improving the robustness of the saddle-point search algorithm.

Let us next examine the validity of the penalty term in the modified rotation problems, Equation (30), for handling inequality constraints. For this purpose, the modified rotation problem with the Riemannian Hessian of the original objective function \( V \) is tentatively
6. Application to ascent trajectory of winged rocket

In this section, the proposed method is applied to the optimization of ascent trajectory of a winged rocket. Reusable winged space transportation systems are expected to realize superior cost efficiency, safety, versatility, and operability, compared to conventional expendable systems. Among possible vehicle configurations, a vertically-launched suborbital vehicle developed under the industry-academia-government collaboration [28] is considered. The latest major specifications of the vehicle used in this section are shown in Table 1. The optimal ascent trajectory of conventional rockets on the one-dimensional vertical line for maximizing their attained altitude has been actively studied analytically [29] and numerically [30]. On the other hand, the optimal ascent trajectory of winged vehicles on the two-dimensional vertical plane considering the lift force requires more investigations.

### Table 1. Major specifications of a winged suborbital rocket.

| Parameter                              | Value     |
|----------------------------------------|-----------|
| Total length [m]                       | 13.4      |
| Wing span [m]                          | 8.4       |
| Lift-off mass, \(m_0\) [kg]            | 18290     |
| Landing mass [kg]                      | 6590      |
| Sea-level thrust, \(T_h\) [N]          | \(2.36 \times 10^5\) |
| Vacuum thrust, \(T_v\) [N]             | \(2.80 \times 10^5\) |
| Propellant consumption rate, \(\dot{m}_p\) [kg/s] | 90.4 |
| Aerodynamic reference area, \(S_{ref}\) [m^2] | 25.8 |

then, the saddle-point search near the lower and upper bounds of \(u\) fails, and it implies that the penalty method employed in this paper for the constraint handling is certainly effective.

### 6.1. Problem formulation

Figure 6 shows the schematics for the ascent trajectory optimization problem. In order to derive a numerically tractable but practically viable problem, the following assumptions and approximations are introduced.

- Two degree-of-freedom equations of motion of the vehicle as a point mass on the vertical plane is employed.
- The Cartesian coordinate system, \([x, h]\), whose origin is located at the launch point, is used. \(x\) and \(h\) denote downrange and altitude, respectively.
- Uniform gravity field with the magnitude of \(g = 9.8 \text{m/s}^2\) is considered.
- Approximate models of temperature and pressure of atmosphere as the smooth functions of altitude are constructed based on U.S. standard atmosphere 1976 [31]. Density and speed of sound are calculated from temperature and pressure.
- An aerodynamic model of the vehicle is built based on the results of computational fluid dynamics (CFD). Lift coefficient, \(C_L\), and drag coefficient, \(C_D\), at representative Mach numbers (0.3, 0.8, 1.0, 2.0, and 3.5) are fitted by the linear and quadratic functions of angle of attack, respectively, as shown in Figure 7. Lift-to-drag ratio, \(L/D\), is included in Figure 7 as well.
- The consumption rate of propellant is constant before the engine cut-off, and the resulting thrust force changes according to the ambient pressure.

The ascent flight is divided into the following three phases as shown in Figure 6.

- Lift-off phase:
  - The vehicle is launched with the vertical attitude, and it continues to ascend vertically for the first five
seconds after the lift-off, in order to leave the launch facility safely.

- Powered-ascent phase:

The vehicle continues to accelerate until the engine thrust is cut off at 128 s after the lift-off. Angle of attack during the powered ascent phase is designed between $-10 \text{deg}$ and $10 \text{deg}$, which is a reasonable range where the longitudinal trim is typically possible.

- Coasting phase:

After the engine cut-off, a coasting ascent flight continues until the vehicle reaches the highest point (apogee point). Aerodynamic forces in this phase are quite small, and they are ignored in the present trajectory optimization problem. In addition, it is imposed that the apogee point is 25 km away from the launch point along $x$ direction. This requirement comes from the ground safety considering the flight termination in abort scenarios and debris fallings in explosive accidents.

The maximization of the apogee altitude is the objective of the trajectory optimization. Based on the assumptions above, the trajectory optimization problem only during the powered-ascent phase is considered as illustrated in Figure 6. Initial conditions at $t = t_0$ are given by the pre-determined trajectory in the lift-off phase. Position of the apogee point can be calculated from the terminal position and velocity at $t = t_f$. As a result, the trajectory optimization problem is formulated as follows:

\begin{align}
\text{find} \quad & x(t) = [h(t), v_x(t), v_h(t)], \quad u(t) = \alpha(t), \quad t_0, t_f, \\
\text{min.} \quad & -h_{apo} = -\left[h(t_f) + \frac{1}{2g} v_h(t_f)^2\right], \\
\text{s.t.} \quad & \frac{dh}{dt} = v_h, \\
& \frac{dv_x}{dt} = \frac{Th}{m} \left[\frac{v_x \cos \alpha}{(v_x^2 + v_h^2)^{\frac{1}{2}}} - \frac{v_h \sin \alpha}{(v_x^2 + v_h^2)^{\frac{1}{2}}}\right] \\
& \frac{dv_h}{dt} = \frac{Th}{m} \left[\frac{v_x \sin \alpha}{(v_x^2 + v_h^2)^{\frac{1}{2}}} + \frac{v_h \cos \alpha}{(v_x^2 + v_h^2)^{\frac{1}{2}}}\right] - g,
\end{align}

where $h$, $v_x$, $v_h$, and $\alpha$ denote altitude, horizontal velocity, vertical velocity, and angle of attack, respectively.

Thrust force, $Th$, depends on the ambient pressure, $p$, in the following way:

$$Th = Th_v - 0.4308p,$$

where the second term represents the pressure loss of the thrust. $x_{apo}$ and $m$ mean the down range at the apogee point and the vehicle mass, respectively, and they are given by

$$x_{apo} = \int_{t_0}^{t_f} v_x(t)dt + v_x(t_f) \frac{v_h(t_f)}{g},$$

$$m = m_0 - \dot{m}_p t_f.$$

Lift, $L$, and drag, $D$, are expressed by

$$L = q C_L S_{ref},$$

$$D = q C_D S_{ref},$$

$$q = \frac{1}{2} \rho (v_x^2 + v_h^2),$$

$$C_L = C_{L,1}(Ma)\alpha + C_{L,0}(Ma).$$
\[ C_D = C_{D,2}(Ma)\alpha^2 + C_{D,1}(Ma)\alpha + C_{D,0}(Ma), \]  
\[ Ma = \left(\frac{v_x^2 + v_h^2}{c}\right)^{\frac{1}{2}}, \]  
where \( q, \rho, Ma, \) and \( c \) are dynamic pressure, air density, Mach number, and speed of sound, respectively. \( C_{L,1}(Ma), C_{L,0}(Ma), C_{D,2}(Ma), C_{D,1}(Ma), \) and \( C_{D,0}(Ma) \) are the nonlinear functions of Mach number such that approximate the CFD results. They are constructed by using the radial basis function network with multi-quadric basis [32]. The atmospheric model is as follows:

\[ \rho = \frac{P}{RT}, \]  
\[ \kappa = \sqrt{\kappa RT}, \]  
\[ T = 7.62 \times 10^{-22}h^5 - 1.12 \times 10^{-16}h^4 + 2.56 \times 10^{-12}h^3 + 2.14 \times 10^{-7}h^2 - 8.35 \times 10^{-3}h + 289.4, \]
\[ p = 10^{(\ln 10 \times 0.005 h^3 + 4.99)}, \]  
where \( \kappa = 1.402 \) and \( R = 287.04. \) Deviations of Equations (65) and (66) from U.S. standard atmosphere 1976 within \( h \in [0, 80 \times 10^3] \) m in terms of the coefficients of determination are 0.977 and 0.999, respectively.

### 6.2. Result and discussion

For the numerical stability, the original problem formulated in the previous subsection is scaled so that the magnitude of resulting variables approximately has the order of unity. Then, this properly scaled problem is transcribed into an NLP problem with \( N = 31. \) Although the constraint given by \( x_{apo} = 25 \times 10^3 \) in Equation (52) is not written in the form of a conventional terminal condition due to Equation (55), it can be converted into an equality constraint by approximating the integration by the Gaussian quadrature. The penalty weight for Equation (53) is \( w_c = 100. \) The resulting NLP problem is solved by using Algorithm 3 with \( r_{test} = 1.0. \)

Two locally optimal solutions and one saddle-point solution between them are obtained as shown in Figures 8–10, respectively. Total number of iterations of Algorithms 1 and 2 for the convergence to the saddle point is 10. The computational local order of convergence given by Equation (46) is around 1.4. They imply that realistic and complex optimal control problems can be solved by the proposed methodology in a scalable manner. The apogee altitudes of these three trajectories are 154.8 km, 151.8 km, and 146.1 km, respectively. Since it is confirmed that the dynamic pressure during the coasting phase is actually small, the assumption of zero aerodynamic force introduced in Section 6.1 is valid.

The left part of Figures 8–10 presents an ascent trajectory consisting of powered-ascent and coasting phases, and its enlarged view around the launch point. Two optimal trajectories are different from each other until around 50 s, during which angle of attack changes between its upper and lower bounds. It may be counterintuitive that the second optimal trajectory is better than the saddle-point trajectory even though it flies towards the negative direction of the down range in the beginning of flight. In a previous design study of a winged suborbital vehicle by the authors [33], a path constraint to make the time derivative of flight path angle negative is intentionally imposed in order to obtain a seemingly suitable ascent trajectory. However,
the present result reveals that such a trajectory design approach gives rise to an inferior ascent trajectory which is close to a saddle point rather than optima. The attainable altitude of the saddle-point trajectory is lower than that of the first optimal trajectory by 8.7 km, which is not huge but practically meaningful.

The reason for different apogee altitudes of three trajectories can be examined quantitatively in terms of vertical velocity loss. The ideal vertical velocity increment realized in the absence of thrust loss, aerodynamic forces, attitude control, and gravity is given by

\[ L_{\text{i}} = \int_{t_0}^{t_f} \frac{Th}{m} - \frac{Th}{m} \, dt, \]  

(67)

The vertical velocity loss is defined as the difference between this ideal velocity increment and the increment actually obtained. The losses concerning the thrust, \( L_t \), the aerodynamic forces, \( L_a \), the control, \( L_c \), and the gravity, \( L_g \), are calculated as follows:

\[ L_{\text{t}} = \int_{t_0}^{t_f} \frac{Th}{m} - \frac{Th}{m} \, dt, \]  

(68)

\[ L_{\text{a}} = \int_{t_0}^{t_f} - \frac{1}{m} \left[ - \frac{v_h D (v_x^2 + v_h^2)^{1/2}}{(v_x^2 + v_h^2)^{1/2}} + \frac{v_x L}{(v_x^2 + v_h^2)^{1/2}} \right] \, dt, \]  

(69)

\[ L_{\text{c}} = \int_{t_0}^{t_f} \frac{Th}{m} \left[ 1 - \frac{v_x \sin \alpha}{(v_x^2 + v_h^2)^{1/2}} - \frac{v_h \cos \alpha}{(v_x^2 + v_h^2)^{1/2}} \right] \, dt, \]  

(70)
Figure 11. Vertical velocity loss of optimal and saddle-point ascent trajectories of a winged rocket.

\[ L_g = \int_{t_0}^{t_f} g \, dt = g(t_f - t_0). \]  

Note that all the feasible trajectories have the same gravity loss in this case, because the duration of powered flight is fixed. Integrands in \( L_t \), \( L_a \), and \( L_c \) represent the instantaneous contribution of thrust loss, aerodynamic forces, and control loss, respectively; and they are shown in Figure 11, for two optimal trajectories and one saddle-point trajectory. Two optimal trajectories have large control losses before 40 s because their thrust vectors are not aligned to the vertical direction. However, these losses are overcome by the positive gains (i.e. negative losses) generated by aerodynamic forces, more specifically lift. As a result, the first optimal trajectory has the smallest total loss, followed by the second optimal trajectory.

Let us explain why the first optimal trajectory stops flying to the down-range direction and finishes exploiting the lift force at around 80 s. When the lift utilization is not beneficial, a more vertical trajectory with lower maximum dynamic pressure (like the present saddle-point trajectory) becomes better, since it can reduce drag. This means that the dynamic pressure can be regarded as the degree of lift utilization. In both the first and second optimal trajectories, dynamic pressure reaches its maximum value at 45 s, where Mach number is around 0.8. By referring to the fact that lift-to-drag ratio deteriorates above Mach 0.8 (see Figure 7), it can be inferred that the lift-assisted ascent is profitable only in the flight regime where lift-to-drag ratio is sufficiently high.

7. Conclusion

This paper presented an enhanced iterative minimization formulation for saddle-point search and its application to multi-modal optimal control. After a local minimum is found, surrounding two saddle points are searched, from which new local minima are expected to be obtained by using a gradient-based optimizer. By repeating this procedure, multiple locally optimal solutions connected by saddle-point solutions are explored systematically. The developed method is applicable to general optimal control problems with equality and inequality constraints. It is especially suitable for finding better local solutions around a currently-obtained solution in complex practical problems, when the pursuit of rigorous global optimality is prohibitively expensive. Numerical experiments with an example problem demonstrated that the proposed algorithm with globalization strategies and efficient constraint handling has robust search performance and superlinear convergence.

Then, the two-dimensional ascent trajectory optimization of a vertically-launched winged rocket for the maximization of the apogee altitude was performed using the present method. Two locally optimal trajectories exploit lift force before the lift-to-drag ratio deteriorates in the transonic flight regime, and it was revealed that a vertically-ascending trajectory that is common in conventional non-winged rockets is actually an inferior saddle-point solution.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

**Koichi Yonemoto** received his B.E. degree from Tohoku University, M.E. degree, and Ph.D. degree in Engineering from The University of Tokyo, Japan, in 1977, 1980, and 1988, respectively. He worked as an aerospace engineer at Kawasaki Heavy Industries, Ltd. from 1980 to 2005, then as a professor at Kyushu Institute of Technology from 2005 to 2019. He is currently a commissioned professor at Tokyo University of Science since 2019. His research interests include future space transportation system, guidance and control, and composite tanks.
Takahiro Fujikawa received his B.E., M.E., and Ph.D. degrees from The University of Tokyo, Japan, in 2011, 2013, and 2016, respectively. He was an Assistant Professor at Kyushu Institute of Technology from 2016 to 2019. In 2019, he joined Tokyo University of Science, where he is currently a Junior Associate Professor (Lecturer). His research interests include future space transportation system, system optimization, and guidance and control.

ORCID

Takahiro Fujikawa http://orcid.org/0000-0001-7709-4090

References

[1] Rao AV. A survey of numerical methods for optimal control. Adv Astronaut Sci. 2009;135(1):497–528.
[2] Betts JT. Practical methods for optimal control and estimation using nonlinear programming. 2nd ed. Philadelphia: SIAM; 2010.
[3] Bohme TJ, Frank B. Hybrid systems, optimal control and hybrid vehicles. Cham: Springer; 2017. Chapter 7, Indirect methods for optimal control: p. 215–231.
[4] Brauer GL, Cornick DE, Stevenson R. Capabilities and applications of the program to optimize simulated trajectories (POST). 1977. (NASA contract report; CR-2770).
[5] Osa T. Multimodal trajectory optimization for motion planning. Int J Rob Res. 2020;39(8):983–1001.
[6] Feng X, Villanueva ME, Chachuat B, et al. Branch-and-lift algorithm for obstacle avoidance control. In: Proceedings of 2017 IEEE 56th Annual Conference on Decision and Control; Dec 12–15; Melbourne, Australia: IEEE. 2017.
[7] Zeng X, Gong S, Li J. Fast solar sail rendezvous mission to near earth asteroids. Acta Astronaut. 2014;105:40–56.
[8] Diedam H, Sager S. Global optimal control with the direct multiple shooting method. Optim Control Appl Methods. 2018;39(2):449–470.
[9] Houck B, Chachuat B. Branch-and-lift algorithm for deterministic global optimization in nonlinear optimal control. J Optim Theory Appl. 2014;162:208–248.
[10] Tsuchiya T. Global optimization of polynomial-expressed nonlinear optimal control problems with semidefinite programming relaxation. J Glob Optim. 2012;54:831–854.
[11] Cruz ILL, Van Willigenburg LG, Van Straten G. Efficient differential evolution algorithms for multimodal optimal control problems. Appl Soft Comput. 2003;3(2):97–122.
[12] Ghosh P, Conway BA. Numerical trajectory optimization with swarm intelligence and dynamic assignment of solution structure. J Guid Control Dyn. 2012;35(4):1178–1191.
[13] Yokoyama N, Suzuki S. Modified genetic algorithm for constrained trajectory optimization. J Guid Control Dyn. 2005;28(1):139–144.
[14] Yin J, Wang Y, Chen JZY, et al. Construction of a pathway map on a complicated energy landscape. Phys Rev Lett. 2020;124(9):090601.
[15] Fujikawa T, Yonemoto K. Finding multiple local solutions to trajectory optimization problems using a saddle-point search method. In: Proceedings of the SICE Annual Conference; 2021 Sep 8–10; Tokyo, Japan: SICE. 2021.
[16] Gao W, Leng J, Zhou X. An iterative minimization formulation for saddle point search. SIAM J Numer Anal. 2015;53(4):1786–1805.
[17] Gao W, Leng J, Zhou X. Iterative minimization algorithm for efficient calculations of transition states. J Comput Phys. 2016;309:69–87.
[18] Xavier AE. Hyperbolic penalty: a new method for non-linear programming with inequalities. Intl Trans Oper Res. 2001;8:659–671.
[19] Garg D, Patterson M, Hager WW, et al. A unified framework for the numerical solution of optimal control problems using pseudospectral methods. Automatica. 2010;46(11):1843–1851.
[20] Golub GH. Some modified matrix eigenvalue problem. SIAM Rev. 1973;15(2):318–334.
[21] Lehoucq RB, Sorensen DC, Yang C. ARPACK users’ guide. Philadelphia: SIAM; 1998.
[22] Yin J, Huang Z, Zhang L. Constrained high-index saddle dynamics for the solution landscape with equality constraints. 2021. p. 32. Available from: https://arxiv.org/abs/2011.13173v2
[23] Zhang R. Newton retraction as approximate geodesics on submanifolds. 2020. p. 9. Available from: https://arxiv.org/abs/2006.14751
[24] Asenjo D, Stevenson JD, Wales DJ, et al. Visualizing basins of attraction for different minimization algorithms. J Phys Chem B. 2013;117:12717–12723.
[25] Gill PE, Murray W, Saunders MA. SNOPT: an SQP algorithm for large-scale constrained optimization. SIAM Rev. 2005;47(1):99–131.
[26] Martins JRRA, Sturdza P, Alonso JJ. The complex-step derivative approximation. ACM Trans Math Softw. 2003;29:245–262.
[27] Grau-Sanchez M, Noguera M, Grau A, et al. On new computational local orders of convergence. Appl Math Lett. 2012;25:2023–2030.
[28] Yonemoto K, Fujikawa T, Yamasaki Y, et al. Preliminary design status of unmanned suborbital spaceplane with LOX/methane engines by Tokyo University of Science’s start-up with the partnership of industries: part 2. In: Proceedings of the 71th International Astronautical Congress; Oct 12–24. IAC-20, D2, 4, 9, x59725; IAF. 2020.
[29] Rugescu RD. Goddard’s 85 years optimal ascent problem finally solved. In: Proceedings of the 57th International Astronautical Congress; Oct 2–6; Valencia, Spain: IAF. IAC-06-E4.3.05. 2006.
[30] Seywald H, Clift EM. Goddard problem in presence of a dynamic pressure limit. J Guid Control Dyn. 1993;14:776–781.
[31] NOAA, NASA, U.S. AirForce. U.S. standard atmosphere. 1976. (NASA technical memorandum; TM-X-74355).
[32] Forrester A, Sobester A, Keane A. Engineering design via surrogate modelling: a practical guide. Chichester: Wiley; 2008. p. 45–49.
[33] Fujiwara T, Yonemoto K. Multidisciplinary design optimization of winged suborbital vehicles considering multiple mission trajectories. Trans JSASS Aerospace Tech Japan. 2021;19:874–883.