Direct Solution of Multi-Objective Optimal Control Problems Applied to Spaceplane Mission Design

Lorenzo A. Ricciardi[*], Christie Alisa Maddock[†] and Massimiliano Vasile[‡]
University of Strathclyde, Glasgow, United Kingdom, G1 1XJ

This paper presents a novel approach to the solution of multi-phase multi-objective optimal control problems. The proposed solution strategy is based on the transcription of the optimal control problem with Finite Elements in Time and the solution of the resulting Multi-Objective Non-Linear Programming (MONLP) problem with a memetic strategy that extends the Multi Agent Collaborative Search algorithm. The MONLP problem is reformulated as two non-linear programming problems: a bi-level and a single level problem. The bi-level formulation is used to globally explore the search space and generate a well spread set of non-dominated decision vectors while the single level formulation is used to locally converge to Pareto efficient solutions. Within the bi-level formulation, the outer level selects trial decision vectors that satisfy an improvement condition based on Chebyshev weighted norm, while the inner level restores the feasibility of the trial vectors generated by the outer level. The single level refinement implements a Pascoletti-Serafini scalarisation of the MONLP problem to optimise the objectives while satisfying the constraints. The approach is applied to the solution of three test cases of increasing complexity: an atmospheric re-entry problem, an ascent and abort trajectory scenario and a three-objective system and trajectory optimisation problem for spaceplanes.

Nomenclature

$\mathbf{b}$	=	static parameter vector
$\mathbf{C}$	=	constraint vector
C_D	=	drag coefficient
C_L	=	lift coefficient
D	=	aerodynamic drag force (N)
E	=	energy function
$f_{s,j}$	=	Bernstein polynomials

[*] Ph.D. Candidate, Aerospace Centre of Excellence, Mechanical & Aerospace Engineering, 75 Montrose Street, G1 1XJ, Glasgow, AIAA Student Member.

[†] Lecturer, Aerospace Centre of Excellence, Mechanical & Aerospace Engineering, 75 Montrose Street, G1 1XJ, Glasgow, AIAA Member.

[‡] Professor, Aerospace Centre of Excellence, Mechanical & Aerospace Engineering, 75 Montrose Street, G1 1XJ, Glasgow, AIAA Member.

$\mathbf{g}$	=	algebraic constraints
g_0	=	magnitude of gravitational acceleration at sea level (m/s^2)
h	=	altitude (m)
I_{sp}	=	specific impulse (s)
$\mathbf{J}$	=	objective vector
L	=	aerodynamic lift force (N)
M	=	mach number
m	=	vehicle mass (kg)
m_p	=	mass of propellant (kg)
$\mathbf{p}$	=	decision vector
p_a	=	atmospheric pressure (Pa)
q_{flux}	=	heat flux (W/m^2)
R_E	=	radius of the Earth (m)
$\mathbf{r}$	=	position vector in inertial frame (m)
S_{ref}	=	aerodynamic surface reference area (m^2)
$\mathcal{T}$	=	time domain
T	=	magnitude of thrust (N)
$\mathbf{t}$	=	time (s)
$\mathbf{u}$	=	control vector
$\mathbf{v}$	=	velocity in inertial frame (m/s)
$\mathbf{w}$	=	weight functions
$\mathbf{x}$	=	state vector
$\mathbf{y}$	=	vector of state weights
$\mathbf{z}$	=	utopia point
α	=	angle of attack (rad)
β_k	=	Gauss integration weights
γ	=	flight path angle (rad)
δ_T	=	throttle
θ	=	latitude (rad)
λ	=	longitude (rad)
μ_E	=	gravitational parameter of Earth (m^3/s^2)
ρ_a	=	atmospheric density (kg/m^3)

σ	=	bank angle (rad)
τ	=	normalised time
ϕ_i	=	scalar objective function
χ	=	heading angle (rad)
$\boldsymbol{\psi}$	=	boundary constraints
$\boldsymbol{\omega}$	=	weight of descent directions
ω_E	=	magnitude of angular velocity of the Earth (rad/s)
Subscripts		
0	=	initial
f	=	final

I. Introduction

This paper proposes a method for the solution of multi-objective, multi-phase optimal control problems. In the past three decades, a considerable body of work has been dedicated to the direct solution of single objective optimal control problems, see e.g., [1–8] and references therein, and translated into a number of commercial and open source software[*],[†]. However, while conditions of optimality and related theoretical aspects of multi-objective optimal control problems have been studied in a number of papers, see [9–13] and references therein, less research has been dedicated to the solution of multi-objective optimal control problems.

Ober-Blöbaum et al. [14] coupled a direct transcription approach with an approach that scalarised the multi-objective vector along directions pointing at predefined unreachable points in the criteria space. Each scalar problem was then solved with a standard NLP solver. This approach was employed to solve an interplanetary trajectory optimisation. Kaya and Maurer [15] proposed a similar approach but used the Pascoletti-Serafini scalarisation [16] to transform the multi-objective optimisation problem in a set of single-objective optimisation problems, and employed the resulting approach to solve chemical reaction engineering and drug dosage problems. Pagano and Mooij [17] optimised the mass of the payload for a launch vehicle and minimised the violation of the constraints as a second objective, Bairstow et al. [18] performed a multiobjective optimisation of a two stage launcher minimising cost and maximising the payload and Roshanian et al. [19] performed robust design optimisation of a two stage launch vehicle by means of multiobjective optimisation, minimising both the mean and the variance of the gross take-off mass when several design and operative parameters where subject to uncertainty. In these last three works the control laws had a simple parametric shape. The parameters describing those shapes were optimisation variables, and stochastic multiobjective optimisation algorithms were employed to find the set of Pareto optimal solutions. Coverstone-Carroll et al. [20] combined Genetic Algorithms

[*] https://en.wikipedia.org/wiki/Trajectory_optimization

[†] https://en.wikipedia.org/wiki/Optimal_control

and optimal control theory in a dual loop algorithm. In the outer loop, a Multi-Objective Genetic Algorithm (MOGA) was generating vectors of co-states and times of flight. For each set, the inner loop was solving a single objective optimal control problem with given time of flight, minimising the propellant consumption. Englander et al. [21] proposed a dual loop algorithm in which the outer loop solves a multi-objective problem handling a set of categorical variables through a multi-objective genetic algorithm and the inner loop solves a set of single objective constrained optimal control problems using Monotonic Basin Hopping [22].

The method proposed in this work is based on a direct transcription with Finite Elements in Time [6] (DFET) and a solution of the resulting Multi-Objective Nonlinear Programming (MONLP) problem with a version of Multi Agent Collaborative Search [23] (MACS) called MACSoc. DFET have been successfully used to solve many difficult single-objective trajectory optimisation problems [3, 4, 24, 25]. Similarly, MACS has been tested and validated on a number of benchmarks of difficult multi-objective optimisation problems[23][26]. Previous work by the authors validated the pairing of DFET and MACS on a set of optimal control problems with known solutions [27][28]. Here, this pairing is extended to treat more complex multi-objective optimal control problems, with the inclusion of static system design parameters and multiple phases. The method proposed in this paper differentiates from Ober-Blöbaum et al. [14] and Kaya and Maurer [15] in that it combines a global exploration and local convergence with a smooth transition between Chebyshev and Pascoletti-Serafini scalarisation [16] and incorporates an automatic and unsupervised procedure to generate feasible first guesses. It also differentiates from [17–21] in that it does not use a generic MOGA but proposes a more efficient memetic approach and implements a more general direct transcription method.

The method is applied to three realistic test cases for spaceplane-based launch vehicles, optimising the ascent, descent and abort trajectories and some key vehicle and mission design variables. A considerable amount of work has been devoted to the off-line optimisation of ascent and re-entry trajectories for launch systems with the inclusion of progressively more sophisticated physics and constraints examining the launch system design and performance [29–33], and the fast and robust generation of a guidance law with the ultimate goal to have an on-line closed loop guidance update [34–39]. This paper departs from these two streams of research, and focuses on the off-line generation of Pareto optimal trade-off solutions for a generic multi-objective optimal control problem. The three test cases were selected to show the benefits of this method to the trade-off and feasibility studies conducted during the initial design phases.

The paper is structured as follows: after formulating the multi-objective optimal control problem, the paper briefly introduces the transcription method and then presents the approach to the solution of the resulting MONLP problem. The application to the three case studies follows with a demonstration of the effectiveness of the proposed approach and the added value of the multi-objective formulation.

II. Direct Transcription of Multi-Objective Optimal Control Problems

Multi-, or more generally many, objective optimal control problems can be formulated as follows:

$$
\begin{aligned}
&\min_{\mathbf{u}\in U,\mathbf{b}\in B} \mathbf{J} \\
&s.t. \\
&\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x},\mathbf{u},\mathbf{b},t) \\
&\mathbf{g}(\mathbf{x},\mathbf{u},\mathbf{b},t) \geq 0 \\
&\boldsymbol{\psi}(\mathbf{x}_0,\mathbf{x}_f,t_0,t_f,\mathbf{b}) \geq 0 \\
&t \in [t_0,t_f]
\end{aligned}
\tag{1}
$$

where $\mathbf{J} = [J_1, J_2, ..., J_i..., J_m]^T$ is, in general, a vector of objectives J_i that are functions of the state vector $\mathbf{x} : [t_0, t_f] \to \mathbb{R}^n$, control variables $\mathbf{u} \in L^\infty(U \subseteq \mathbb{R}^{n_u})$, static parameters $\mathbf{b} \in B \subseteq \mathbb{R}^{n_b}$ and time t. The functions $\mathbf{x}(t)$ belong to the Sobolev space $\mathcal{W}^{1,\infty}$ while the objective functions are $J_i : \mathbb{R}^{3n+2} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_b} \longrightarrow \mathbb{R}$. The objective vector is subject to a set of dynamic constraints with $\mathbf{F} : \mathbb{R}^n \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_b} \times [t_0, t_f] \longrightarrow \mathbb{R}^n$, algebraic constraints $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_b} \times [t_0, t_f] \longrightarrow \mathbb{R}^{n_g}$, and boundary conditions $\boldsymbol{\psi} : \mathbb{R}^{2n+2} \times \mathbb{R}^{n_b} \longrightarrow \mathbb{R}^{n_\psi}$. In the following we will consider only Mayer's types of optimal control problems, whereby each objective function is expressed as a scalar function of the boundary conditions, boundary times and static parameters:

$$J_i = \phi(\mathbf{x}_0, \mathbf{x}_f, t_0, t_f, \mathbf{b}) \tag{2}$$

Note that, if boundary states and times, $\mathbf{x}_0, \mathbf{x}_f, t_0, t_f$ are free decision variables, they can be included in the static parameter vector $\mathbf{b}$ and the union of controls and static parameters is called a decision vector. Thus, without loss of generality, one can say that the solution of problem (1) is a subset of $U \times B$ that satisfies the constraints and contains Pareto efficient decision vectors. This leads to the following definition.

Definition II.1 *Given the subset $\Omega_U \subset U \times B$ of feasible decision vectors, a decision vector $[\mathbf{u}^*, \mathbf{b}^*] \in \Omega_U$ is said to be Pareto efficient if $[\mathbf{u}^*, \mathbf{b}^*] \not\succ [\mathbf{u}, \mathbf{b}]$, $\forall [\mathbf{u}, \mathbf{b}] \in \Omega_U$.*

The symbol of dominance $\succ$ is introduced to indicate that if $[\mathbf{u}, \mathbf{b}]_1 \succ [\mathbf{u}, \mathbf{b}]_2$ then $J_i([\mathbf{u}, \mathbf{b}]_2) \leq J_i([\mathbf{u}, \mathbf{b}]_1)$ for $i = 1, \ldots, m$ and $\exists j$ such that $J_j([\mathbf{u}, \mathbf{b}]_2) < J_j([\mathbf{u}, \mathbf{b}]_1)$.

A. Direct Transcription with Finite Elements in Time

In this paper it is proposed to transcribe problem (1) into a many-objective, non-linear programming problem via DFET [6]. DFET was initially proposed by Vasile [24] in 2000 and uses finite elements in time on spectral bases to transcribe the differential equations into a set of algebraic equations. Finite Elements in Time (FET) for the indirect solution of optimal control problems were initially proposed by Hodges and Bless [40], and during the late 1990s evolved to the discontinuous version. As pointed out by Bottasso and Ragazzi [41], FET for the forward integration of ordinary differential equations are equivalent to some classes of implicit Runge-Kutta integration schemes, can be extended to arbitrary high-order, are very robust and allow full h-p adaptivity. In the past decade, direct transcription with FET on spectral bases has been successfully used to solve a range of difficult problems: from the design of low-thrust multi-gravity assist trajectories to Mercury [4] and to the Sun [25], to the design of weak stability boundary transfers to the Moon, low-thrust transfers in the restricted three body problem and optimal landing trajectories to the Moon [24].

Following the standard procedure for DFET transcription (see [6] for more details), the time domain $\mathcal{T}$ is decomposed into N finite elements such that:

$$\mathcal{T} = \bigcup_{j=1}^{N} \mathcal{T}_j(t_{j-1}, t_j) = \left[t_0,\, t_f\right] \tag{3}$$

with $t_N = t_f$. On each time element, the differential constraints in (1) are first recast in weak form and integrated by parts leading to:

$$\int_{\mathcal{T}_j} \dot{\mathbf{w}}^T \mathbf{x} + \mathbf{w}^T \mathbf{F}(\mathbf{x}, \mathbf{u}, \mathbf{b}, t)\; dt - \mathbf{w}^T(t_j)\mathbf{x}_j^b + \mathbf{w}^T(t_{j-1})\mathbf{x}_{j-1}^b = 0 \tag{4}$$

where $\mathbf{w}$ are generalised weight functions, and $\mathbf{x}_j^b$ and $\mathbf{x}_{j-1}^b$ are the values of the states at the boundaries of each element. Then, states, controls and weight functions are transcribed in polynomial form as follows:

$$\mathbf{x}_j(t) = \sum_{s=0}^{l_x} f_{s,j}(t)\, \mathbf{x}_{s,j} \tag{5a}$$

$$\mathbf{u}_j(t) = \sum_{s=0}^{l_u} f_{s,j}(t)\, \mathbf{u}_{s,j} \tag{5b}$$

$$\mathbf{w}_j(t) = \sum_{s=0}^{l_x+1} f_{s,j}(t)\, \mathbf{w}_{s,j} \tag{5c}$$

where functions $f_{s,j}$ are chosen among the space of Bernstein polynomials. It is practical to redefine Eq. (4) and basis functions (5) over the normalised interval $[-1,\ 1]$ through the transformation:

$$\tau = 2\frac{t - \frac{t_j - t_{j-1}}{2}}{t_j - t_{j-1}} \qquad t_{j-1} \leq t \leq t_j \tag{6}$$

where τ is the normalised time. This way the domain of the basis function is constant and irrespective of the size of the element. By substituting Eqs. (5) into (4) and solving the integral with a Gauss quadrature formula on l_u Gauss nodes, one gets:

$$\sum_{k=0}^{l_u} \beta_k \left[\dot{\mathbf{w}}_j(\tau_k)^T \mathbf{x}_j(\tau_k) + \mathbf{w}_j(\tau_k)^T \mathbf{F}_j(\tau_k)\frac{\Delta t_j}{2}\right] - \mathbf{w}^T(1)\mathbf{x}_j^b + \mathbf{w}^T(-1)\mathbf{x}_{j-1}^b = 0 \tag{7}$$

where τ_k and β_k are Gauss nodes and weights, $\Delta t_j = (t_j - t_{j-1})$ and $\mathbf{F}_j(\tau_k)$ is the shorthand notation for $\mathbf{F}\left(\mathbf{x}_j(\tau_k), \mathbf{u}_j(\tau_k), \mathbf{b}, t(\tau_k)\right)$. Since Eq. (7) must be valid for every arbitrary $\mathbf{w}_{s,j}$, Eq. (7) gives rise to a system of $(l_x + 1)$ vector equations for each element:

$$\begin{aligned}
&\textstyle\sum_{k=0}^{l_u} \beta_k \left[\dot{f}_{1,j}(\tau_k)\,\mathbf{x}_j(\tau_k) + f_{1,j}(\tau_k)\mathbf{F}_j(\tau_k)\frac{\Delta t_j}{2}\right] + \mathbf{x}^b_{j-1} = 0 \\
&\vdots \\
&\textstyle\sum_{k=0}^{l_u} \beta_k \left[\dot{f}_{s,j}(\tau_k)\mathbf{x}_j(\tau_k) + f_{s,j}(\tau_k)\mathbf{F}_j(\tau_k)\frac{\Delta t_j}{2}\right] = 0 \\
&\vdots \\
&\textstyle\sum_{k=0}^{l_u} \beta_k \left[\dot{f}_{l_x+1,j}(\tau_k)\mathbf{x}_j(\tau_k) + f_{l_x+1,j}(\tau_k)\mathbf{F}_j(\tau_k)\frac{\Delta t_j}{2}\right] - \mathbf{x}^b_j = 0
\end{aligned} \tag{8}$$

Path constraints are evaluated at Gauss nodes for each element:

$$\mathbf{g}\left(\mathbf{x}_j(\tau_k), \mathbf{u}_j(\tau_k), \mathbf{b}, t(\tau_k)\right) \geq 0 \tag{9}$$

Continuity conditions are then imposed on the boundary states of adjacent elements, such that all boundary values $\mathbf{x}^b_j$ cancel out except for the initial boundary term of the first element $\mathbf{x}^b_0$ and last boundary element of the last element $\mathbf{x}^b_f$. Thus, once all the elements are assembled together, the only parameters (or decision variables) that remain to be defined are $\mathbf{x}_{s,j}$ for the states, $\mathbf{u}_{s,j}$ for the controls, the boundary states $\mathbf{x}^b_0$ and $\mathbf{x}^b_f$, the time variables t_0 and t_f and the static parameters $\mathbf{b}$. Each transcribed objective function in Mayer's form (2) is calculated simply as:

$$\tilde{J}_i = \phi_i(\mathbf{x}^b_0, \mathbf{x}^b_f, t_0, t_f, \mathbf{b}) \tag{10}$$

The time domain $\mathcal{T}$ corresponds to a single time phase, or timeline. However, a general problem can have multiple phases either in series or in parallel. For example, a multi-stage vehicle can have one phase per vehicle stage with all phases connected in series for the ascent, and/or branching parallel phases for the upper stage ascent and first stage descent and return. These branching phases can also be seen in abort scenarios, which comprises one of the test cases in this paper. When N_p phases are present, dynamic constraints (7), path and boundary constraints, $\mathbf{g}$ and ψ, and objective functions (10) are defined on each timeline. In order to connect different timelines, a set of N_{ip} inter-phase constraints are introduced:

$$\psi_{s_p}\left(\mathbf{x}^b_{0,\mathbf{I}_{s_p}}, \mathbf{x}^b_{f,\mathbf{I}_{s_p}}, t_{0,\mathbf{I}_{s_p}}, t_{f,\mathbf{I}_{s_p}}\right) \geq 0 \quad s_p = 1, ..., N_{ip} \tag{11}$$

where the index vector $\mathbf{I}_{s_p}$ collects all the indexes of the phases that are connected by constraint ψ_{s_p}. Note that the number of phases is fixed, but their temporal order is actually defined by the inter-phase constraints (11). Section IV will show one example with two sequential phases and another one with branching parallel phases.

The resulting MNLP problem coming from the transcription of problem (1), with the inclusion of interphase constraints (11), can be written, in vector form, as:

$$\begin{aligned}&\min_{\mathbf{y}\in Y,\mathbf{p}\in\Pi} \tilde{\mathbf{J}}\\&s.t.\\&\mathbf{C}(\mathbf{y},\mathbf{p}) \geq 0\end{aligned} \tag{12}$$

where $\mathbf{y} = [\mathbf{x}_{0,1}, .., \mathbf{x}_{s,j}, ..., \mathbf{x}_{l_x,N}]^T$, Y is a box in $\mathbb{R}^{n_Y}$ with $n_Y = n(l_x + 1)N$, $\mathbf{p} = [\mathbf{u}_{0,1}, .., \mathbf{u}_{s,j}, ..., \mathbf{u}_{l_u,N}, \mathbf{b}^*]^T$ collects all the static and discretised dynamic control variables, $\mathbf{b}^* = [\mathbf{b}, \mathbf{x}_0^b, \mathbf{x}_f^b, t_0, t_f]^T$, $\Pi \subseteq \mathbb{R}^{n_s} \times \mathbb{R}^{n_b^*}$, with $n_s = n_u(l_u + 1)N$ (assuming that each element has the same number of control parameters) and $n_b^* = n_b + 2n + 2$, and $\mathbf{C}$ collects all constraints, including boundary and interphase ones.

Similar to problem (1), the solution of problem (12) is a subset of $\Omega_\Pi \subset \Pi$ that satisfies the constraints and contains vectors $\mathbf{p}$ that are Pareto efficient. For continuous functions, the subset Ω_Π is a manifold in $\mathbb{R}^{n_s+n_b^*}$ with dimension $\leq (m-1)$ [42]. In the following, the goal will be to identify a pre-defined countable number of Pareto efficient solutions contained in Ω_Π.

III. Solution of the Transcribed Problem

Problem (12) is solved with a memetic many-objective optimisation algorithm, adapted from MACS (Multi-Agent Collaborative Search [23, 26]) and called MACSoc, that combines a stochastic agent-based global search with a local (gradient-based in this case) refinement of the solutions [27, 28, 43]. The overall solution process implemented in MACSoc is summarised in Algorithm 1.

At the start of MACSoc, N_a candidate solutions are generated with a Latin Hypercube sampling, associated to a population $\mathcal{P}_0$ of N_a agents, and an attempt is made to make each candidate solution feasible before the optimisation process starts (line 1 in Algorithm 1). Both the global search and local refinement strategies implemented in MACSoc require the definition of a set of descent directions in criteria space, thus, after initialising the agents, the algorithm generates N_w uniformly spread weight vectors $\boldsymbol{\omega}$ (line 2 in Algorithm 1) that define the components of N_w descent vectors; this is explained further in Subsection E. Each agent will be associated to a different weight vector, allowing each agent to converge to a different part of the Pareto front.

The global search generates candidate solutions, for the decision vector, using a combination of social and individualistic actions (lines 4 and 7 in Algorithm 1). Each action generates a candidate decision vector, starting from the current solution allocated to a given agent j, and submits it to a bi-level optimisation problem, where the inner level makes the candidate decision vector feasible, with respect to differential, path, and boundary constraints, and the outer level assesses whether the solution of the inner level represents an improvement with respect to the current solution allocated to agent j. All feasible and non-dominated solutions update the current population $\mathcal{P}_k$ and are saved in an

Algorithm 1 MACS optimal control (MACSoc) framework

```
1: Initialise population $\mathcal{P}_0$ and global archive $\mathcal{A}_0$, $k = 0$, $\rho_{\mathcal{B}} = 1$
2: Initialise weight vectors $\boldsymbol{\omega}$
3: while $n\_fun\_eval < max\_fun\_eval$ do
4:     Run individualistic heuristics on $\mathcal{P}_k$ using bi-level formulation
5:     $\mathcal{P}_k \rightarrow \mathcal{P}_k^+$
6:     Update archive $\mathcal{A}_k$ with potential field filter
7:     Run social heuristics combining $\mathcal{P}_k^+$ and $\mathcal{A}_k$ using bilevel formulation
8:     Update archive $\mathcal{A}_k$ with potential field filter
9:     $\mathcal{P}_k^+ \rightarrow \mathcal{P}_k^\dagger$
10:    if local search triggered then
11:        Run gradient based refinement using single level formulation
12:        $\mathcal{P}_k^\dagger \rightarrow \mathcal{P}_k^*$
13:        Update archive $\mathcal{A}_k$ with potential field filter
14:        $\mathcal{P}_k^* \rightarrow \mathcal{P}_{k+1}$
15:    else
16:        $\mathcal{P}_k^\dagger \rightarrow \mathcal{P}_{k+1}$
17:    end if
18:    $k = k + 1$
19:    Update $\rho_{\mathcal{B}}$
20: end while
```

archive $\mathcal{A}_k$ (lines 5, 6, 8, 9 and 13 in Algorithm 1). After every user specified number of iterations, and as a last step before the algorithm ends, the local refinement is triggered, and the archive and population are updated with the refined solutions (lines 10 to 17 in Algorithm 1). The local refinement solves a single level scalarised version of problem (12). The process proceeds alternating social and individualistic actions, with periodic local refinement, until a maximum number of calls to the objective vector *max_fun_eval* is reached. The overall algorithmic complexity is dominated by the NLP solver used in the bi-level problem and for the local refinement.

In the following both the bi-level and single level problems are explained in more detail together with the heuristics used to generate new candidate solutions. Note that the combined use of the bi-level formulation, for global exploration, and single level formulation for local convergence, within MACSoc is one of the distinctive features of the proposed approach compared to previous works such as [14], [15] and [21].

A. Bi-level Global Optimisation Problem

The global search part of the algorithm solves the following two-level problem:

$$
\begin{aligned}
&\min_{\mathbf{p}^*} \tilde{\mathbf{J}}(\mathbf{y}^*, \mathbf{p}^*) \\
&s.t. \\
&(\mathbf{y}^*, \mathbf{p}^*) = \text{argmin}_{(\mathbf{y},\mathbf{p})} \left\{\delta_p(\mathbf{y}, \mathbf{p}) \,|\, \mathbf{C}(\mathbf{y}, \mathbf{p}) \geq 0\right\}
\end{aligned}
\tag{13}
$$

Problem (13) defines two different optimisation sub-problems at two different levels. The outer level handles the objective vector $\tilde{\mathbf{J}}$ and generates tentative decision vectors $\mathbf{p}$. Tentative solutions are then submitted to the inner level, whose goal is to find the state and control vectors $\mathbf{y}^*$ and $\mathbf{p}^*$ that satisfy constraints $\mathbf{C}$ and minimise an inner cost function $\delta_p = \|\mathbf{p}^* - \mathbf{p}\|$. Thus, the inner level will look for the closest feasible solution to the tentative one generated by the outer level. The outer level then receives the solution $(\mathbf{y}^*, \mathbf{p}^*)$ and proceeds by evaluating the objective functions associated to $\mathbf{p}^*$. The inner level problem is solved with a generic NLP solver (Matlab `fmincon` in this case).

In order to reduce the number of iterations required by the inner level to converge, the outer level stores the feasible states $\mathbf{y}^*$ at iteration k to be used as a warm start for the inner level at iteration $k+1$. As illustrated in Fig. 1, the feasible states $\mathbf{y}_k^*$ are preserved from iteration k to iteration $(k+1)$ and the outer level only generates a new tentative vector $\mathbf{p}_{k+1}$. The inner level at iteration $(k+1)$ will thus use $\mathbf{y}_k^*$ and $\mathbf{p}_{k+1}$ as initial guesses for states and controls. Because of the way $\mathbf{p}_{k+1}$ is generated, even if $\mathbf{y}_k^*$ is associated to $\mathbf{p}_k^*$, it works well as an initial guess also when associated to $\mathbf{p}_{k+1}$.

When individualistic actions are applied, each agent generates one or more tentative vectors through three mechanisms that are triggered one after the other in this order: Inertia → Pattern Search → Differential Evolution. If any of these mechanisms produces an improved solution, the following ones are not triggered and the process proceeds by updating the population and archive (line 5 and 6 of Algorithm 1). The three mechanisms operate as follows:

- **Inertia** is triggered by agent j only if, in the previous iteration, agent j generated an improved solution. In this case a step with random length is taken in the direction defined by the vector $\mathbf{p}_k^* - \mathbf{p}_{k-1}^*$.
- **Pattern Search** consists of changing one optimisation parameter at a time by a random amount in each direction within a given neighbourhood $\mathcal{B}_j$ of agent j. The order by which the parameters are changed is a random permutation of the number of decision parameters. The process is repeated until either an improvement is registered or the maximum number of trials has been reached. As in [23], the maximum number of trials is dynamically adjusted during the optimisation process: when the Archive is empty, the maximum number of parameters scanned is equal to the total number of optimisaton parameters. This maximum value is decreased linearly as the Archive fills up, until only one optimisation parameter is changed when the Archive is full. The neighbourhood $\mathcal{B}_j$ is a box centred in the position of the agent in parameter space and with the edges equal to the edges of the search space Π multiplied by the scaling parameter $\rho_{\mathcal{B}_j}$.
- **Differential Evolution** generates a sample with the simple heuristic:

$$\mathbf{p}_{trial,j} = \mathbf{p}_j + \xi_1 \mathbf{e}\left(\left(\mathbf{p}_j - \mathbf{p}_{j_1}\right) + c_F\left(\mathbf{p}_{j_2} - \mathbf{p}_{j_3}\right)\right) \tag{14}$$

  where $\mathbf{p}_j$ is the current solution, $\mathbf{p}_{j_1}, \mathbf{p}_{j_2}, \mathbf{p}_{j_3}$ are three randomly chosen solutions from the current population $\mathcal{P}_k$, ξ_1 is a uniformly distributed random number in the unit interval, c_F is a constant and $\mathbf{e}$ is a mask vector

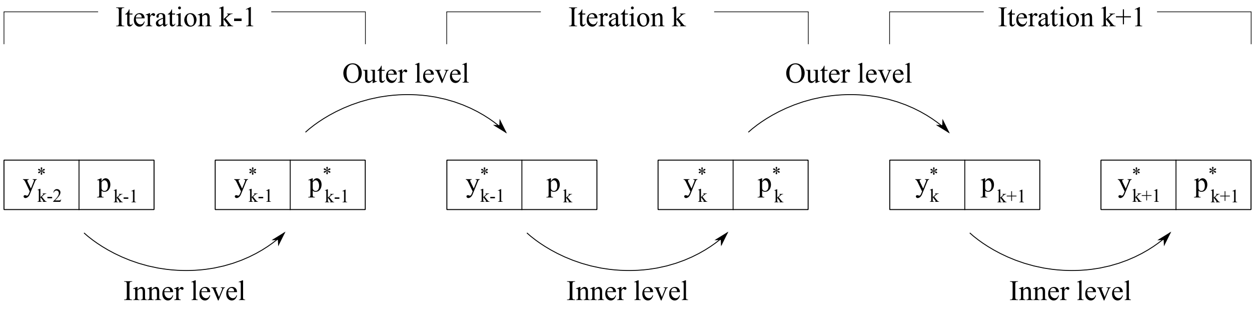


Fig. 1 Schematic representation of the bilevel approach acting on a single solution.

constructed as follows:

$$e_j = \begin{cases} 1, & \text{if } \xi_2 < CR \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where ξ_2 is another uniformly distributed random number in the unit interval and CR is a constant, known as crossover rate. In this work, $c_F = 0.9$ and $CR = 1$.

If no improvement is made after trying all the three heuristics, $\rho_{\mathcal{B}}$ is halved, while if an improvement is made, it is doubled until the initial value $\rho_{\mathcal{B}} = 1$ is reached again.

When social actions are applied, the outer level uses the entire populations of agents to generate a tentative solution using heuristic (14), but the parent solutions $\mathbf{p}_{j_1}, \mathbf{p}_{j_2}, \mathbf{p}_{j_3}$ are chosen from the union of the current population $\mathcal{P}_k$ and the current archive $\mathcal{A}_k$.

A candidate solution $(\mathbf{y}^*, \mathbf{p}^*)$, generated by the inner level, is evaluated, in the outer level, by computing the weighted Chebychev norm:

$$\Phi_i = \max_i \omega_i \left[\tilde{J}_i(\mathbf{y}^*, \mathbf{p}^*) - z_i \right] \tag{16}$$

where ω_i are the components of a weight vector in objective space and $z_i = \min_{\mathcal{P}_k \cup \mathcal{A}_k} \tilde{J}_i$ are the components of the current utopia point, or the point whose coordinates are the minimum value of each objective function $\tilde{J}_i$ over all the elements of the population and the archive combined. Norm (16) measures the distance, along each coordinate in objective space, between the current objective vector $\tilde{\mathbf{J}}(\mathbf{y}^*, \mathbf{p}^*)$ and the current utopia point $\mathbf{z}$, weights the different distances via $\boldsymbol{\omega}$, and takes the worst, or maximum, weighted distance. Therefore, given the value of Φ_i at step k, or Φ_i^k, an improvement corresponds to $\Phi_i^{k+1} < \Phi_i^k$. This improvement criterion has two very important properties: first, it allows one to reach even non-convex parts of the Pareto front, and second, if the weights are chosen appropriately, it allows one to efficiently converge to the minimum possible values of each objective function. The generation of the weights $\boldsymbol{\omega}$ will be explained in Subsection E.

Note that, if the inner level does not converge to the required tolerance, the objective functions of the outer level are recalculated to be the infinity norm of the constraint violation plus the maximum values of each objective functions in the archive and population. This creates an adaptive rejection mechanism: if none of the agents are feasible, the ones that best satisfies the feasibility is temporarily entered in the archive with the next iterations trying to further improve their feasibility. Once an agent finds a feasible solution, it will explore the search space through the global bi-level approach, generating several feasible and non dominated solutions. These solutions will enter in the archive because they will dominate many of the existing non-feasible ones, and thanks to the social actions some agents will be directly moved onto those solutions, allowing the whole population to converge to feasible solutions in a handful of iterations. Finally if any tentative vector for **y** and **p** is outside the boundaries of the search space $Y \times \Pi$, the vector is shrunk till it is back into the search space (for more details please refer to [23]).

B. Single Level Local Search

The local refinement solves the following scalarised problem for each agent j:

$$\begin{aligned} &\min_{\epsilon \geq 0} \ \epsilon \\ &s.t. \\ &\omega_{i,j}\ \vartheta_{i,j}(\mathbf{y},\mathbf{p}) \leq \epsilon \qquad \text{for } i = 1,\dots,m \\ &\mathbf{C}(\mathbf{y},\mathbf{p}) \geq 0 \end{aligned} \tag{17}$$

where $\boldsymbol{\omega}_j$ is a vector of weights, $\vartheta_{i,j}$ is the i^{th} component of the rescaled objective vector of the j^{th} agent, and ϵ is a slack variable. This reformulation of the problem, known as Pascoletti-Serafini scalarisation [16], is constraining the agent's movement, in criteria space, within the descent cone defined by the point $(\epsilon\mathbf{d}_j + \zeta_j)$ along the direction $\mathbf{d}_j = (1/\omega_{1,j},\dots,1/\omega_{i,j},\dots,1/\omega_{m,j})$. The rescaled objective vector is defined as:

$$\boldsymbol{\vartheta}_j(\mathbf{y},\mathbf{p}) = \frac{\tilde{J}_{i,j}(\mathbf{y},\mathbf{p}) - \tilde{z}_i}{z^*_{i,j} - \tilde{z}_i} \qquad \text{for } i = 1,\dots,m \tag{18}$$

where $\mathbf{z}^*_j$ is equal to $\tilde{\mathbf{J}}_j(\overline{\mathbf{y}},\overline{\mathbf{p}})$, $(\overline{\mathbf{y}},\overline{\mathbf{p}})$ is the initial guess for the solution of (17) and $\tilde{\mathbf{z}} = \mathbf{z} - \mathbf{z}_A$ with $\mathbf{z}_A$ the nadir of the archive, or point whose components are the maximum values of all the components of the objective vectors in the archive. From the normalisation one can derive the components of the vector $\boldsymbol{\zeta}_j$:

$$\zeta_{i,j} = \frac{z_i}{z^*_{i,j} - \tilde{z}_i} \qquad \text{for } i = 1,\dots,m \tag{19}$$

This allows the components of $\vartheta_j(\mathbf{y},\mathbf{p})$ to have values of 1 at the beginning of the local search, and value 0 if the agent converges to the target point $\tilde{\mathbf{z}}$. With this normalisation the single level approach avoids biases when the objectives have significantly different scales. The construction of the descent directions will be explained in Subsection E. It is important to remark that Problem (16) and (17) are equivalent and lead to the same optimal solution if the target point for the Pascoletti-Serafini scalarisation coincides with the utopia point and the weight vectors are the same. As such, the following theorem holds true [44]:

Theorem III.1 *A point* $(\epsilon, \mathbf{y}, \mathbf{p}) \in \mathbb{R} \times Y \times \Pi$ *is a minimal solution of* (17) *with* $\mathbf{z} \in \mathbb{R}^m$, $z_j < \min_{\mathbf{y} \in Y, \mathbf{p} \in \Pi} J_j(\mathbf{y}, \mathbf{p})$, $j = 1, \ldots, m$, *and* $\omega \in int(\mathbb{R}^m_+)$ *if and only if* $\mathbf{y}$ *and* $\mathbf{p}$ *are a solution of* (16).

Therefore, by combining (16) in the global search phase with (17) in the refinement phase, the algorithm realises a smooth transition from global exploration of the Pareto set, to local convergence.

C. Archiving Strategy

MACS employs a unique archiving strategy proposed by Ricciardi and Vasile in [23] that is also applied in the context of MACSoc. When the elements in the archive $\mathcal{A}$ are less than the maximum allowed cardinality of $\mathcal{A}$, every new feasible and non-dominated solution is stored in the archive. Once the maximum size is reached, a retention-rejection policy is implemented. The retention-rejection policy is based on a minimum energy principle. New elements are added to $\mathcal{A}$ only if they minimise the potential function:

$$E\left(\tilde{\mathbf{J}}_1, \cdots, \tilde{\mathbf{J}}_{N_{\mathcal{A}}}\right) = \sum_{i=1}^{N_{\mathcal{A}}} \sum_{j=i+1}^{N_{\mathcal{A}}} \frac{1}{(\tilde{\mathbf{J}}_i - \tilde{\mathbf{J}}_j)^T (\tilde{\mathbf{J}}_i - \tilde{\mathbf{J}}_j)} \tag{20}$$

where $N_{\mathcal{A}}$ is the number of elements in $\mathcal{A}$. To avoid biasing the in the rejection-retention process when the objectives have different scales, the objective values of the set of non dominated solutions are all normalised between 0 and 1. This leads to a combinatorial problem that can be solved approximately but efficiently using the approach described by Ricciardi and Vasile [23], and returns a uniformly spread set of points. This minimum energy criterion is also used for the generation of uniform descent directions, as will be explained in Subsection E.

D. Generation of the Initial Feasible Population

Before the optimisation starts, MACSoc generates an initial population of agents $\mathcal{P}_0$ (see line 1 of Algorithm 1) with the following four-step automatic and unsupervised procedure:

1) A first guess for the decision vectors is generated with a Latin Hypercube sampling within the prescribed boundaries. State variables for each phase are initialised with a simple linear interpolation between initial and final conditions.
2) For each phase, each integral equation (7) is made feasible by solving only the inner level subproblem of problem (13) with an NLP solver.
3) Starting from the solution at step 2, for each phase, all constraints related to that phase are then included and the resulting problem is satisfied again applying the same NLP solver to the subproblem in (13).

4) All phases are connected together and the inter-phase constraints are satisfied applying again the same NLP solver to the subproblem in (13).

If, at the end of the initialisation phase, an agent is associated to a solution that is not feasible within the prescribed tolerance, that solution is still included in the initial population P_0 and submitted to the subsequent optimisation cycle. In the following, the feasibility level required to the initial population is the same required for the rest of the algorithm, which is 10^{-6}, so if the NLP converges, the solution generated by this approach is a fully feasible solution. By default, the NLP solver is allowed to use a maximum number of calls to the constraint function that is equal to $10(n_b^* + n_s + n_Y)$. For this procedure, an Interior Point NLP algorithm was used because it delivered a more robust and consistent convergence to feasible solutions.

E. Definition of the descent directions and target points

The weight vectors for the bi-level global search are generated as follows: first, a simplex in objective space is generated through simplex lattice design [45]. Then, the points of this simplex lattice are projected on the unit sphere by dividing their position vectors by their distance to the origin. This gives a fairly uniform distribution of weight vectors (and thus descent directions) in any N_w dimensional space. In order to generate a more uniform distribution, however, these weight vectors are used as an initial guess for the following optimisation problem:

$$\begin{aligned} &\min\ E(\omega_1, \ldots, \omega_{N_w}) \\ &s.t. \\ &\omega_i^T \omega_i = 1 \end{aligned} \tag{21}$$

where $E(\omega_1, \cdots, \omega_{N_w})$ is calculated using Eq. (20). This optimisation problem can be quickly solved with a standard NLP solver. As a reference, from the initial lattice to the final optimised distribution the generation of 106 uniformly spread weight vectors for the three objective problem in Section C takes 22 iterations of the NLP solver with an SQP algorithm, which translates into approximately half a second in Matlab on an i7 laptop. While this approach is valid for general m-objective problems, for two objective problems it is simpler and faster to generate uniformly angularly spaced weight vectors. In the following $N_a = N_w$ and each agent is associated to the closest descent direction in criteria space, at the initialisation stage, with the constraint that no two agents can have the same descent direction.

For the single level approach, the weight vectors $\omega_j = [\sqrt{2}, \ldots, \sqrt{2}]^T$ are allocated to all agents except to those m agents that minimise each individual objective function. For these m agents the weight vectors are $\omega_j = [0, \ldots, j, \ldots, 0]^T$ with $j = 1, ..., m$. These weights are called orthogonal because correspond to the m orthogonal directions in criteria space. If agent j associated to weight ω_j does not generate any improvement after two iterations, a new random orthogonal weight is associated to j and problem (17) is solved with the added constraints:

$$\tilde{J}_i \leq z_i \quad \forall i \neq j \tag{22}$$

The reason for the different choice of weight vectors between the bi-level and the single level formulation can be explained as follows: the bi-level formulation explores globally the search space with a population of agents, thus there is the need to maximise the spreading of the solutions, on the contrary, the single-level is used to improve the local convergence of each agent in a normalised criteria space. Thus the goal of the single level is to return dominating solutions without altering too much their spreading in criteria space.

IV. Case Studies

The approach to the solution of multi-objective optimal control problems described in previous sections is applied to three problems of increasing complexity: a known re-entry benchmark problem with 2 objectives, a 2 phase ascent problem with 3 objectives, and 3 phase branched ascent and abort problem with 2 objectives. The code and all the test cases in this paper were implemented in Matlab 2017b and run on a laptop with an i7 processor under Windows 10. The gradient based refinement was run every 10 iterations in all cases, and the NLP solver used in both the bi-level and single level formulations employed an SQP algorithm. The maximum number of calls to the constraint function for the NLP solver in the bi-level formulation was set equal to $n_b^* + n_s + n_Y$, while it was increased to $10(n_b^* + n_s + n_Y)$ in the single level refinement.

A. Physical models

Before describing the test cases, we present the physical models that are common to all three problems.

1. Dynamical model

The vehicle dynamics are modelled as a 3DOF point mass moving in an Earth Centred Inertial reference frame (adapted from [5]),

$$\dot{r} = \dot{h} = v \sin\gamma \tag{23a}$$

$$\dot{\theta} = \frac{v}{r} \cos\gamma \cos\chi \tag{23b}$$

$$\dot{\lambda} = \frac{v}{r\cos\theta} \cos\gamma \sin\chi \tag{23c}$$

$$\dot{v} = \frac{T(\delta_T)\cos\alpha - D}{m} - g\sin\gamma \tag{23d}$$

$$\dot{\gamma} = \frac{T(\delta_T)\sin\alpha + L}{mv}\cos\sigma + \left(\frac{v}{r} - \frac{g}{v}\right)\cos\gamma \tag{23e}$$

$$\dot{\chi} = \frac{T(\delta_T)\sin\alpha + L}{mv\cos\gamma}\sin\sigma + \frac{v}{r\cos\theta}\cos\gamma\sin\chi\sin\theta \tag{23f}$$

$$\dot{m} = -\dot{m}_p(\delta_T) \tag{23g}$$

where $r = \|\mathbf{r}\|$ is the modulus of the position vector $\mathbf{r}$, $h = r - R_E$ is the altitude, λ and θ are longitude and latitude, $v = \|\mathbf{v}\|$ is the magnitude of velocity vector, γ and χ are the flight path and heading angles, m is the mass of the vehicle, L and D are the aerodynamic lift and drag forces, $g = \mu_E/r^2$ is the gravitational acceleration, T is the thrust produced by the engine and $\dot{m}_p$ is the propellent mass flow rate. The control variables are the angle of attack α, bank angle σ, and the throttling δ_T of the engine.

2. Atmospheric and aerodynamic models

The International Standard Atmosphere model was used to model the atmospheric temperature, pressure p_a, and density ρ_a as a function of the radius r assuming a spherical Earth model with a radius $R_E = 6371\,\mathrm{km}$ and constant angular rotational velocity $\omega_E = 7.292\,118 \times 10^{-5}\,\mathrm{rad\,s^{-1}}$. The atmosphere is assumed to rotate with the same angular velocity as the Earth, so the aerodynamic forces were computed using the velocity of the vehicle relative to the air with no wind (or other disturbances),

$$L = \frac{C_L \rho_a S_{ref} v_{rel}^2}{2} \tag{24}$$

$$D = \frac{C_D \rho_a S_{ref} v_{rel}^2}{2} \tag{25}$$

where $\mathbf{v}_{rel} = \mathbf{v} - \boldsymbol{\omega_E}\mathbf{r}$ and given the aerodynamic coefficients for lift C_L and drag C_D, and the reference area of the vehicle S_{ref}.

3. Propulsion model

The thrust vector is assumed to be always aligned with the vehicle longitudinal body axis and is proportional to the vacuum thrust T_{vac} of the engine and modulated by the throttle control $\delta_T \in [0,\ 1]$. An additional term is added to account for the losses due to the difference between the nozzle's exit pressure and the external atmospheric pressure p_a. The resulting model is,

$$T = \delta_T \left(T_{vac} - A_e p_a\right) \tag{26}$$

where A_e is the nozzle exit area. The mass flow rate of the propellent $\dot{m}_p$ is given by

$$\dot{m}_p = \frac{\delta_T T_{vac}}{I_{sp} g_0} \tag{27}$$

where I_{sp} is the specific impulse of the engine and g_0 is the Earth gravitational acceleration at sea level.

B. Optimal Descent Trajectory

The first test case is based on a benchmark problem proposed by Betts [5] who analysed the unpowered re-entry of a Space Shuttle-like vehicle controlled by changing the angle of attack α and bank angle σ. To be consistent with Betts [5], the throttle was set to $\delta_T = 0$ for all t, $\omega_E = 0$ and units are in the Imperial system. Furthermore, the following models and reference values were taken from Betts [5]: the exponential model for the atmosphere, the linear model for C_L, the parabolic model for C_D, the aerodynamic reference area of $S_{ref} = 2690\ \text{ft}^2$ ($249.9\ \text{m}^2$) and the vehicle mass of 6309 sl (92 t). Bounds on the rate of change of the flight path and heading angles were imposed: $|\dot{\gamma}| \leq 0.035\ \text{rad}\ \text{s}^{-1}$ and $|\dot{\chi}| \leq 0.035\ \text{rad}\ \text{s}^{-1}\text{s}$. A semi-empirical correlation for the heat flux at the nose was given as

$$q_{flux} = \left(c_0 + c_1\alpha + c_2\alpha^2 + c_3\alpha^3\right) c_4 \sqrt{\rho_a}\, (c_5 v)^{c_6} \leq 70\ \text{btu ft}^{-2}\ \text{s}^{-1} \tag{28}$$

with the coefficients $c_0, c_1, c_2, c_3, c_4, c_5, c_6$ as reported in [5]. Boundary conditions, lower bounds and upper bounds for the optimisation variables are listed in Table 1.

1. Objectives

In [5] two different single objective problems were proposed: a maximum cross-range (equivalent to maximising the final latitude θ_f since $\theta_0 = 0$) and a minimum peak heat flux $\max q_{flux}$ problem. Here, we propose, instead, the

Table 1 Lower bounds, upper bounds and boundary conditions for the Optimal Descent Trajectory case

Variable [units]	Lower bound	Upper Bound	Initial value	Final value
Radius r [kft]	20900	21300	21163	20983
Latitude θ [rad N]	$-\pi/2$	$\pi/2$	0	≥ 0.2618
Longitude λ [rad E]	$-\pi$	π	0	Free
Velocity v [ft s^{-1}]	1	30000	25600	2500
Flight path angle γ [rad]	$-\pi/2$	$\pi/2$	-0.0175	-0.0873
Heading angle χ [rad]	$-\pi$	π	$\pi/2$	Free
Angle of attack α [rad]	$-\pi/2$	$\pi/2$	Free	Free
Bank angle σ [rad]	$-\pi/2$	0	Free	Free

following single multi-objective optimisation problem:

$$\begin{aligned}
&\min_{t_f,\mathbf{u}} [J_1,\ J_2]^T = \left[-\theta_f,\ q_u\right]^T \\
&s.t \\
&q_{flux} \leq q_u
\end{aligned} \tag{29}$$

2. Numerical settings

The problem was formulated as a single time phase, with boundary conditions defined in Table 1, discretised using 6 finite elements and order 9 Bernstein polynomials for both states and controls, resulting in 121 optimisation parameters for the outer level (120 for the control variables, and 1 for the free final time), and 484 total variables for the single level and inner level NLP. A limit of 20000 calls to the objective vector was given to MACSoc, a population of 10 agents was deployed in the search space and the size of the archive $\mathcal{A}$ was limited to 10 elements. The initialisation of the population required approximately 1 second for 8 of the 10 agents, while for 2 agents it took approximately 3 minutes because the NLP solver did not converge to a feasible solution in the maximum number of iterations. However, as soon as the optimisation loop started, all solutions immediately became feasible thanks to the bi-level approach. The total runtime was approximately 1 hr.

3. Results

Figure 2 shows the 10 Pareto optimal solutions in the archive $\mathcal{A}$ at the end of the optimisation process, while Figs. 3–6 show altitude, velocity, lift-to-drag ratio L/D, heat flux, angle of attack, bank angle and angular velocities for the flight path and heading angles for each of the 10 solutions in $\mathcal{A}$. Figure 2 shows that the method is able to find an even spread of elements belonging to the Pareto front including the two single objective solutions given in [5], here labelled as Betts A and Betts B, corresponding to Solutions 1 and 10, without any externally supplied guesses. Solution 1 has objective values $(-0.2618, 28.0016)$ and Solution 10 has objective values $(-0.5345, 69.9997)$, while the corresponding

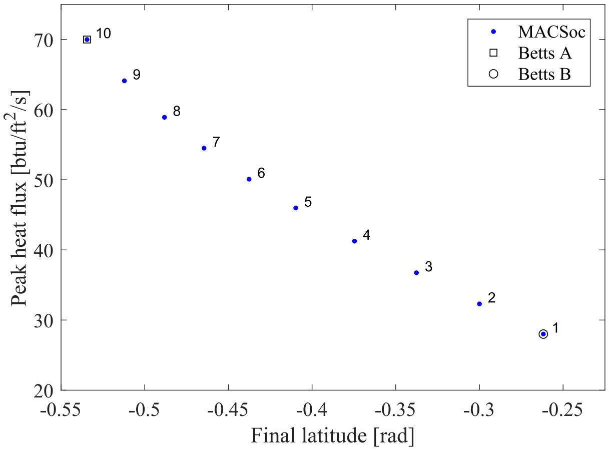


Fig. 2 Optimal descent trajectory: Pareto optimal solutions stored in the archive $\mathcal{A}$ at the end of the optimisation process and the two published single objective solutions from Betts [5].

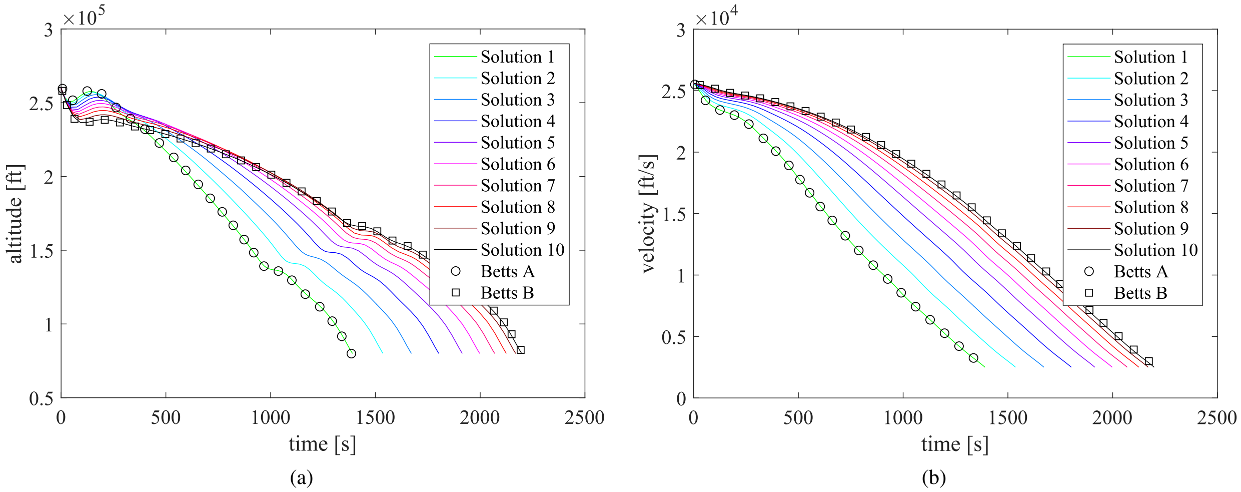
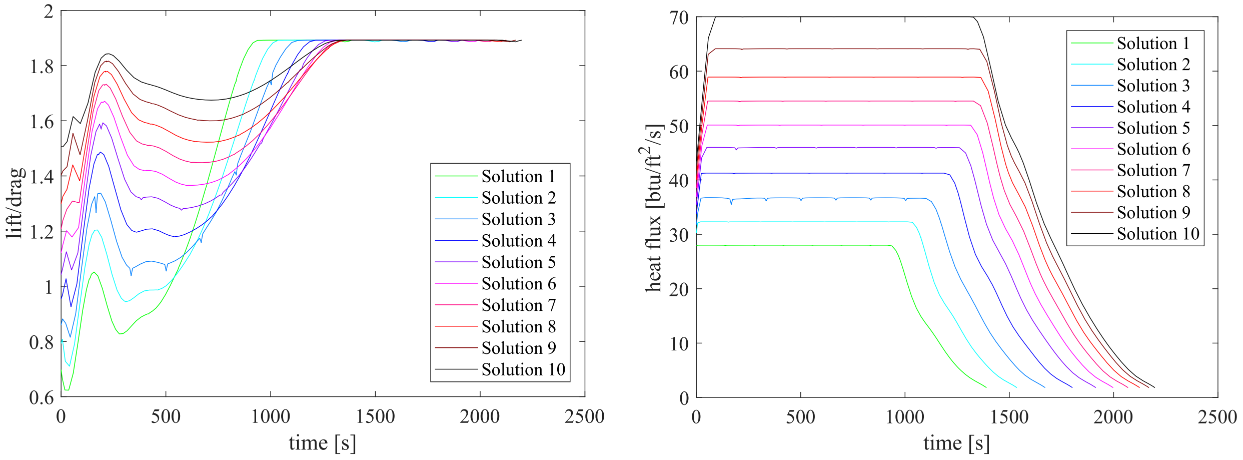


Fig. 3 Optimal descent trajectory: time-history of a) the altitude and b) the velocity for each of the 10 solutions in the Pareto front in Fig. 2 plus the two published single objective solutions from Betts [5].

solutions from the reference are (−0.2618, 27.9982) and (−0.5345, 70) respectively. The relative difference in the second objectives is below 10^{-4} and can be attributed to the different NLP solvers and settings employed, and the presence of an iterative mesh refinement procedure in the reference solutions. In Figs. 3 and 5 the circles and squares represent the solutions from [5] for the two individual objective functions. The figures show a very good match between the result generated with MACSoc.

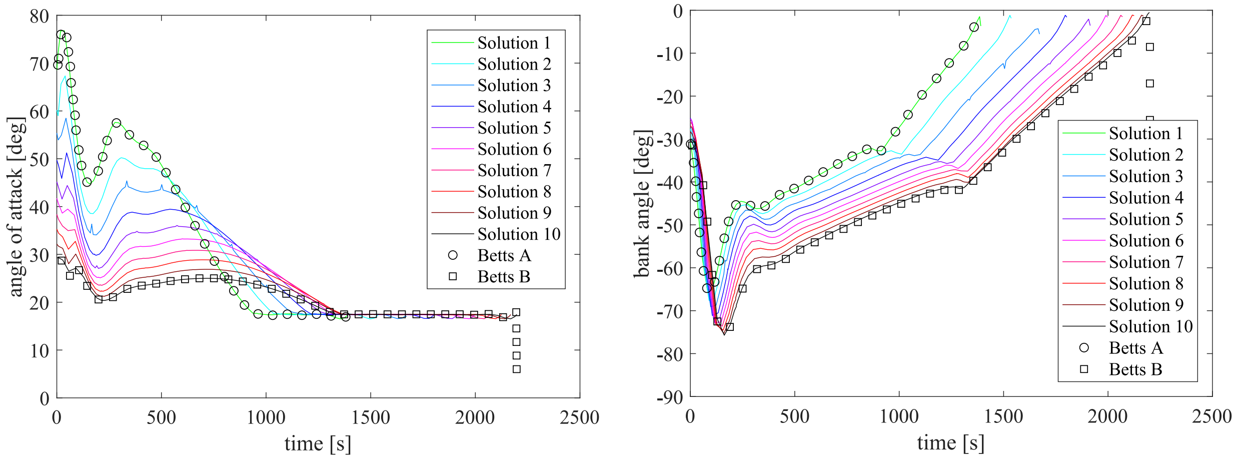
A clear trend emerges from the solutions of the multi-objective problem: the minimum peak heat flux solution is also the shortest in time with longer re-entries imposing higher peak heat fluxes. This is due to the initial altitude skip in the trajectory (see Fig. 3a in the first 200 s), which is more pronounced for the shorter time re-entries and causes the velocity to decrease faster but at the price of having less energy and time to maximise the cross-range. Since the initial



(a)

(b)

Fig. 4 Optimal descent trajectory: time-history of a) the lift to drag ratio and b) the heat flux for each of the 10 solutions in the Pareto front in Fig. 2.

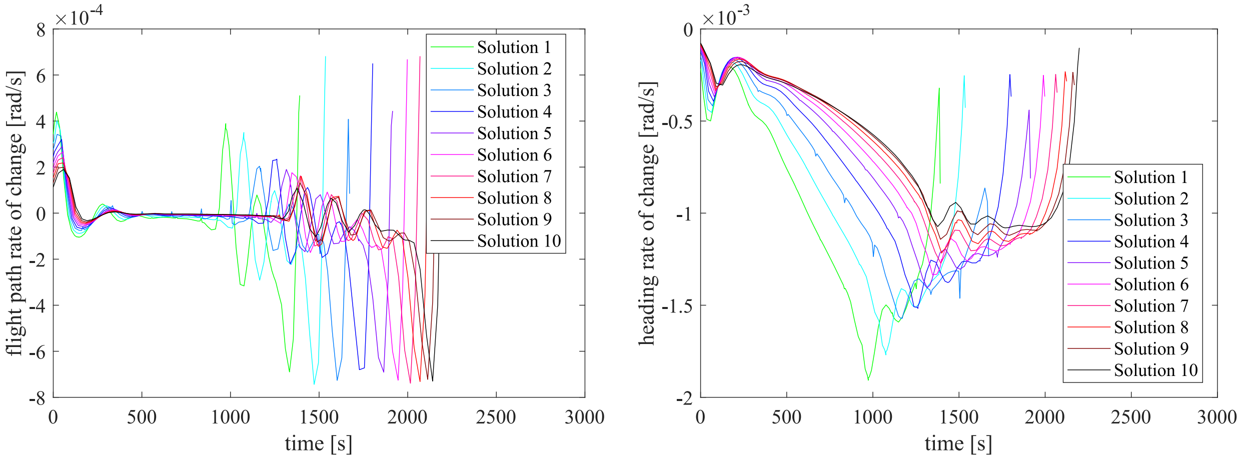


(a)

(b)

Fig. 5 Optimal descent trajectory: time-history of a) the angle of attack and b) the bank angle for each of the 10 solutions in the Pareto front in Fig. 2 plus the two published single objective solutions from Betts [5].

conditions are fixed, the corresponding heat flux is also fixed and this gives the minimum possible peak heat flux. All the trajectories present a similar control profile: an initial peak heat flux containment phase, with high bank and angle of attack, followed by a maximum aerodynamic efficiency phase to maximise the cross-range when heat flux is no longer a concern. The angle of attack during the initial descent is limited by the objective to minimise the heat flux, which is a function of the velocity, altitude and angle of attack, while a high bank angle is used to maximise the cross-range with no penalty to the heat flux. Angular rates for the flight path and heading angles are well below the imposed constraints along all the trajectories.



(a)

(b)

Fig. 6 Optimal descent trajectory: time-history of a) the rate of change of the flight path angle $\dot{\gamma}$ and b) the heading angle $\dot{\chi}$ for each of the 10 solutions in the Pareto front in Fig. 2.

C. Three-objective Ascent Problem

This test case is the multi-objective, multidisciplinary design of a rocket-powered, two-stage launch vehicle optimised for the ascent to orbit. The vehicle is air dropped from a carrier aeroplane flying at 200 m s^{-1} at an altitude of 10 km eastbound along the equator, with an initial flight path angle of 10°. It has to deliver a 500 kg payload to a 650 km altitude circular equatorial orbit. The aim of this test case is to minimise the initial gross mass of the vehicle, examining the trade-off between the engine sizing and dry masses of each of the two stages. The vacuum thrust ratings of the two rocket engines are set as optimisation variables, which through the mass model directly affect the dry masses of the two vehicle stages. Similarly, the mass of propellant used in each stage also affects the dry mass of each stage by altering the mass of the tanks. The vehicle design assumes a recoverable first stage using a winged spaceplane design, with an expendable upper stage with no lifting surfaces. As the focus here is on the vehicle design of the mass and propulsion systems, a simple aerodynamic model was used for both stages: for the first $C_L = 0$, $C_D = 0.1$ and $S_{ref} = 73.73\,\mathrm{m}^2$, while for the second $C_L = 0$, $C_D = 0.01$ and $S_{ref} = 1\,\mathrm{m}^2$.

The ascent trajectory was divided into two phases: Phase 1 is the ascent of the integrated vehicle (combined first and second stage vehicles), and Phase 2 is the ascent of only the second stage vehicle. The initial and final conditions and bounds for optimisation variables are listed in Table 2.

Table 2 Lower bounds, upper bounds and boundary conditions for the Three Objective Ascent case

Variable [units]	Lower bound	Upper Bound	Initial value (Phase 1)	Final value (Phase 2)
Radius r [km]	6371	7171	6381	7021
Latitude θ [rad E]	$-\pi/2$	$\pi/2$	0	Free
Longitude λ [rad N]	0	π	0	Free
Velocity v [m s^{-1}]	465	10000	665	7535
Flight path angle γ [rad]	$-\pi/2$	$\pi/2$	0.1745	0
Heading angle χ [rad]	0	π	$\pi/2$	$\pi/2$
Vehicle mass m [t]	0	100	Free	Free
Angle of attack α [rad]	0	0.3491	Free	Free
Bank angle σ [rad]	-0.1745	0.1745	Free	Free
Throttle δ_T	0	1	Free	Free
$T_{vac,1}$ [MN]	0	2	N.A.	N.A.
$T_{vac,2}$ [kN]	0	200	N.A.	N.A.
$m_{p,1}$ [t]	0	100	N.A.	N.A.
$m_{p,2}$ [t]	0	1	N.A.	N.A.

1. Structural mass models

For the first stage, the dry mass is a function of the engine mass $m_{eng,1}$ and propellent mass required for the first phase $m_{p,1}$,

$$m_{dry,1} = -l_3\tilde{m}_{p,1}^3 + l_2\tilde{m}_{p,1}^2 + l_1\tilde{m}_{p,1} + l_0 + m_{eng,1} \tag{30}$$

$$\tilde{m}_{p,1} = \frac{m_{p,1} - l_4}{l_5} \tag{31}$$

For the second stage vehicle, the dry mass was assumed to be

$$m_{dry,2} = \frac{0.1}{0.9}m_{p,2} + m_{eng,2} \tag{32}$$

The gross vehicle masses for each phase are then given by,

$$m_{0,1} = m_{dry,1} + m_{p,1} + m_{dry,2} + m_{p,2} \tag{33}$$

$$m_{0,2} = m_{dry,2} + m_{p,2} \tag{34}$$

The vacuum thrust of the engines was used to estimate their structural mass based on an empirical linear relationship of existing commercial engines,

$$m_{eng} = l_6T_{vac} + l_7 \tag{35}$$

where l_6, l_7 are constants. The mass model was developed in parallel for an industrial vehicle and cannot be released publicly [46]. For the first stage engine, $0 \leq T_{vac} \leq 2\,\mathrm{MN}$ and $I_{sp} = 332\,\mathrm{s}$, while for the second stage engine $0 \leq T_{vac} \leq 200\,\mathrm{kN}$ and $I_{sp} = 352\,\mathrm{s}$. Propellent masses were limited to 100 t for the first stage and 20 t for the second.

2. Objectives

The aim of the optimisation is to study the trade off between propellent efficient designs and designs that require relatively small engines. The objective functions were to minimise the gross vehicle mass $m_{0,1}$ and the two ratios between the vacuum thrust of the stage engine, and the gross weight at the beginning of each phase. The thrust-to-weight metric also gives an indication of the vehicle loads or induced accelerations the vehicle experiences during flight. The higher the ratio between thrust and mass, the higher the loads imposed on the vehicle, thus one option is to minimise loading by minimising the thrust to weight ratio. Reducing the vacuum thrust reduces the engine performance however, which requires often longer duration trajectories and more propellant, which in turn increase the vehicle mass.

$$\min_{t_f,\mathbf{u},T_{vac,1},T_{vac,2},m_{p,1},m_{p,2}} [J_1,\ J_2,\ J_3]^T = \left[m_{0,1},\ \frac{T_{vac,1}}{g_0 m_{0,1}},\ \frac{T_{vac,2}}{g_0 m_{0,2}}\right]^T \tag{36}$$

3. Numerical settings

The problem was discretised using 4 DFET elements of order 7 for both states and controls, and both phases, resulting in a total of 207 optimisation variables for the outer level and 666 optimisation variables for the single level and inner level NLP. A limit of 80000 calls to the objective vector was given to the optimiser, 106 agents were deployed in the search space and the same maximum number of solutions were kept in the Archive. The initialisation of the population required between 5 seconds and 5 minutes per agent. Matching conditions between the phases were imposed on all state variables except for the mass, for which the following instantaneous drop was imposed at the stage separation:

$$m_{0,2} = m_{f,1} - m_{dry,1} \tag{37}$$

4. Results

Figure 7 shows the 106 Pareto optimal solutions in the archive at the last iteration, with an additional colorbar indicating gross take-off mass. The shape of this 3D Pareto front resembles a smooth half cup. The figure shows the 3D surface in the middle, and the three orthogonal projections. As can be seen, the algorithm found a very good spread set of solutions, all of which are feasible and locally Pareto optimal up to the requested 10^{-6} threshold.

Figures 8 and 9 show the altitude and velocity profiles plus the flight path angle and throttle time histories of the three extreme solutions of the Pareto front. The altitude, velocity and throttle profiles of the minimum gross mass and minimum first stage ($T_{vac,1}/m_{0,1}g_0$) solutions are similar, while their flight path angles differ substantially during the initial ascent: in both cases the first stage engine is working at full throttle and for a comparable time, but given the

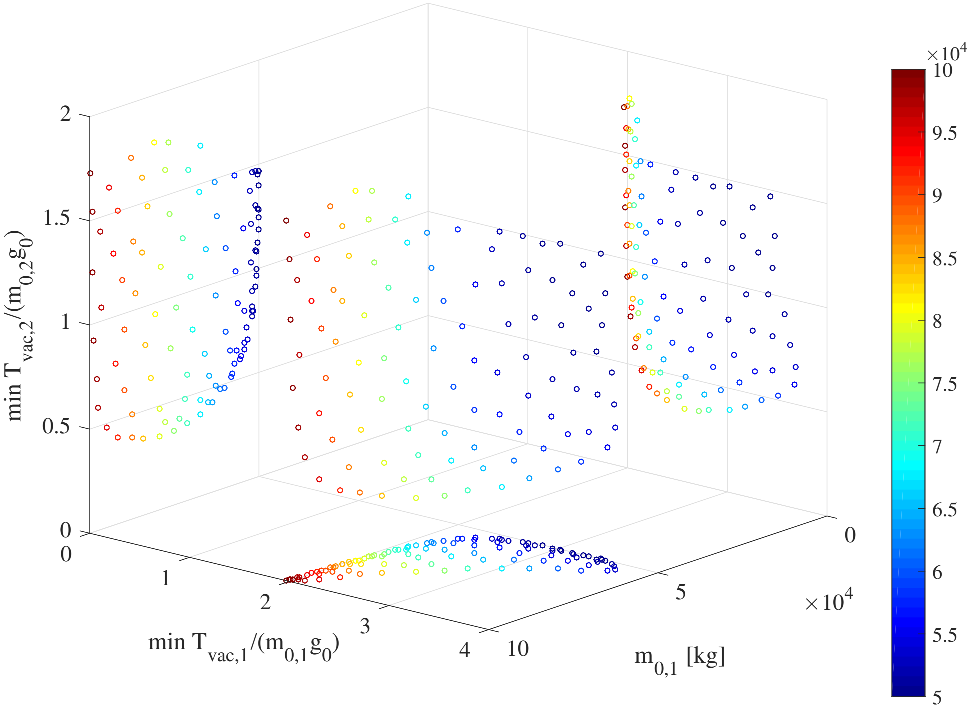


Fig. 7 Three-Objective Ascent: set of Pareto-optimal solutions, colorbar indicates gross mass of the vehicle. The 3D Pareto front is in the middle, with orthographic projections shown on each coordinate plane.

relatively lower thrust engine of the minimum ($T_{vac,1}/m_{0,1}g_0$) case, the resulting flight path angle dips and becomes negative causing the vehicle to briefly lose altitude. The minimum second stage ($T_{vac,2}/m_{0,2}g_0$) solution is instead quite different: the first stage engine has to compensate for the relatively small second stage engine by pushing the vehicle to a higher altitude, velocity and flight path angle at the separation point. The second stage engine has to operate at maximum throttle for a comparatively longer length of time after the separation, and has a higher throttle setting during the final circularisation burn in order to compensate for its lower thrust, as shown in Fig. 9b. The total flight duration is also slightly longer than the other two.

Table 3 Design parameters for the three extreme cases of the Three-Objective Ascent case

Solution	Stage	Initial mass [t]	Propellant mass [t]	Dry mass [t]	Vacuum thrust [kN]	Thrust weight ratio	Δv [km s^{-1}]
min($m_{0,1}$)	1	49.995	29.632 (59.27%)	20.363 (40.73%)	1682.611	3.432	2.836
	2	8.765	7.063 (80.59%)	1.699 (19.38%)	126.100	1.467	5.664
min($T_{vac,1}/m_{0,1}g_0$)	1	100.000	72.830 (72.83%)	27.170 (27.17%)	1930.137	1.968	4.115
	2	11.789	9.717 (82.42%)	2.071 (17.57%)	200.000	1.730	6.003
min($T_{vac,2}/m_{0,2}g_0$)	1	79.318	60.629 (76.44%)	18.689 (23.56%)	2000.000	2.571	4.565
	2	4.390	3.234 (73.66%)	1.156 (26.34%)	15.124	0.351	4.606

Table 3 reports the vehicle design parameters for the Pareto extrema (i.e., the solutions that minimise each objective individually) including a breakdown of the vehicle masses with the relative percentage values with respect to the stage's initial mass, engine vacuum thrust, thrust to weight ratio, and resulting Δv contribution. The solution with minimum initial mass requires high ratios of vacuum thrust to initial weight, though the vacuum thrust of the engines does not reach the maximum allowed values. Propellent mass is approximately 60% of the total mass of the first stage and approximately 80% of the total of the second stage. Total Δv is of $8.5\,\mathrm{km\,s^{-1}}$, with the first stage contributing approximately for $2.8\,\mathrm{km\,s^{-1}}$ or 33% of the total, and the rest coming from the second stage. The ratio between the payload and gross vehicle mass is approximately 1%.

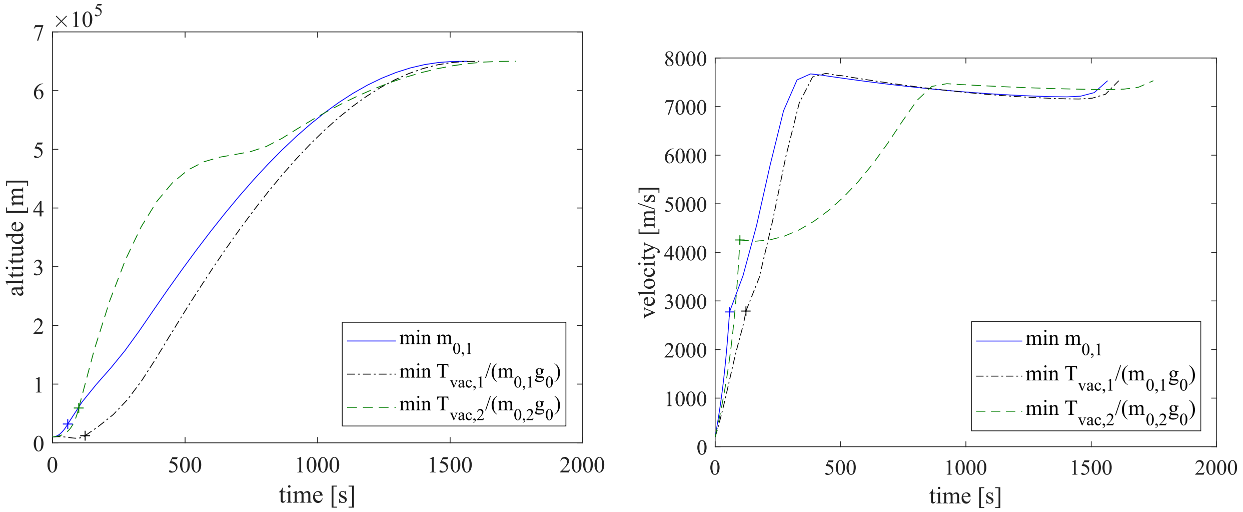
The solution corresponding to the minimum thrust to weight ratio of the first stage requires a larger vehicle with a substantially higher amount of propellant: its initial mass reaches the maximum allowed value for the mass of the vehicle (see Table 2), and is double the value of the previous case. Of this gross mass, approximately 70% is propellant for the first stage. The ratio between the payload mass and the initial mass is 0.5%. The total required Δv is $10.1\,\mathrm{km\,s^{-1}}$, with $6\,\mathrm{km\,s^{-1}}$ coming from the second stage. The second stage engine also has the maximum possible vacuum thrust and consumes more propellant than the previous case leading to a high $(T_{vac,2}/m_{0,2}g_0)$ at the cost of a minimised first stage $(T_{vac,1}/m_{0,1}g_0)$.

The solution corresponding to the minimum thrust to weight ratio of the second stage requires an intermediate initial mass, approximately 60% more than the minimum initial mass case. The ratio between the payload mass and the initial mass is 0.63%, and the required Δv totals $9.1\,\mathrm{km\,s^{-1}}$, evenly spread between the two stages. This is true also for the propellant mass, representing approximately 75% of the total of each stage and totalling twice as much as the minimum gross take-off mass case. The first stage engine has to compensate by taking the maximum allowed value of vacuum thrust, with the resulting thrust to weight ratio being higher than in the previous case, leading to higher induced accelerations. However, the second stage is significantly lighter than the other solutions both in terms of dry mass and propellent mass, and its engine has a vacuum thrust one order of magnitude smaller than the previous solutions.

D. Optimal Ascent and Abort Scenarios

The third test case is the multidisciplinary design of a spaceplane accounting for abort scenarios. Other studies [47] have looked at optimising the abort descent for an independently determined optimal ascent trajectory. Here, the design of the vehicle and trajectory is formulated as a multi-objective optimisation to account for the vehicle design and performance during the ascent to orbit under nominal conditions and the descent under abort conditions, for different abort scenarios.

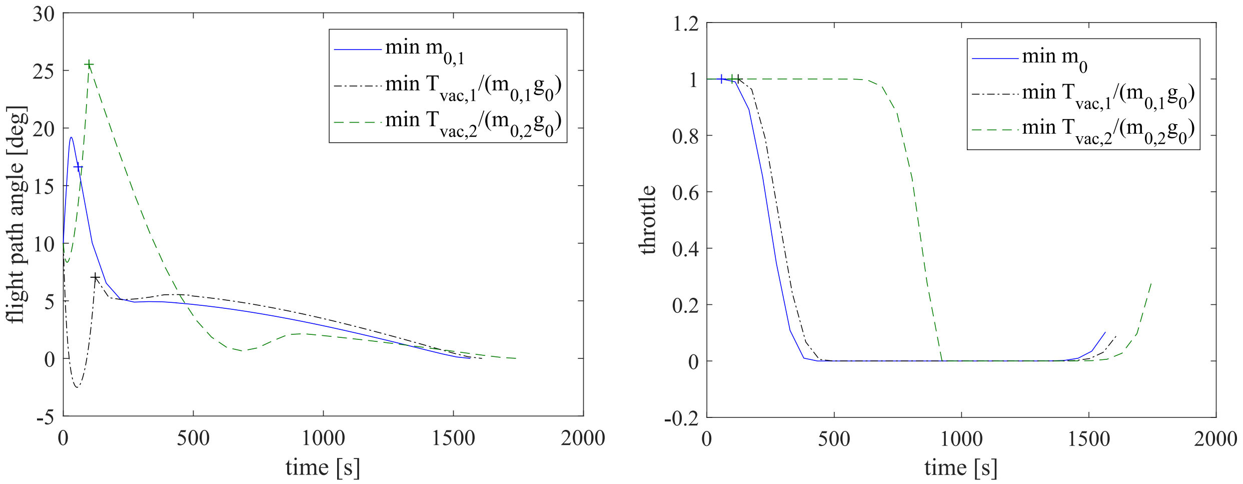
The vehicle is a single stage to orbit spaceplane designed to be air-dropped from a carrier aircraft, similar in design to the X-34. As in the three-objective ascent case, the drop point from the carrier aircraft is set at 10 km altitude and 200 m/s. The mission is to reach a 100 km altitude, circular equatorial parking orbit. The propulsion system is rocket-based with T_{vac} = 1340 kN and I_{sp} = 450 s. An abort is designed to occur after complete engine failure at a



(a)

(b)

Fig. 8 Three-Objective Ascent: time-history of a) the altitude and b) the velocity for the three extreme solutions of the Pareto front in Fig. 7. The + indicates the stage separation point.



(a)

(b)

Fig. 9 Three-Objective Ascent: time-history of a) the flight path angle and b) the throttle for the three extreme solutions of the Pareto front in Fig. 7. The + indicates the stage separation point.

specific time t_{fail}. To study the worst case, no propellent dumping was allowed.

The optimisation problem is configured to find the optimal control for the trajectories, and the optimal sizing of the wing area, which affects the downrange performance during the abort and the dry mass of the vehicle. The initial flight path angle is also set as an optimisable design parameter.

The timeline was divided into three phases: Phase 1 considers the normal ascent trajectory before the failure occurs, from t_0 to t_{fail}. At t_{fail}, the timeline branches into two parallel phases: in Phase 2 the engine is assumed to be working normally and the spaceplane ascends to its target orbit, while instead in Phase 3 the engine has failed and the spaceplane attempts an emergency landing. The boundary conditions for the states and controls are given in Table 4.

Inter-phase constraints were introduced at the branching point to impose the continuity of states and controls between Phase 1 and Phase 2 and continuity of the states only between Phases 1 and 3. The throttle in Phase 3 was imposed to be zero.

Table 4 Lower bounds, upper bounds and boundary conditions for the Optimal Ascent and Abort Scenarios case

			Initial value	Final value	
Variable [units]	Lower bound	Upper Bound	Phase 1	Phase 2	Phase 3
Radius r [km]	6371	6496	6381	6471	6373
Latitude θ [rad N]	$-\pi/2$	$\pi/2$	0	0	0
Longitude λ [rad E]	0	π	0	Free	Free
Velocity v [m s^{-1}]	465	10000	665	7848	Free
Flight path angle γ [rad]	$-\pi/2$	$\pi/2$	$\lvert\gamma_{0,1}\rvert \leq 0.1745$	0	$\gamma_{f,3} \geq 0.1745$
Heading angle χ [rad]	0	π	$\pi/2$	$\pi/2$	Free
Vehicle mass m [t]	0	100	Free	Free	Free
Angle of attack α [rad]	0	0.6109	Free	Free	Free
Bank angle σ [rad]	−0.1745	0.1745	Free	Free	Free
Throttle δ_T	0	1	Free	Free	N.A.
S_{ref} [m^2]	20	400	N.A.	N.A.	N.A.

1. Aerodynamic model

Polynomial models of the lift and drag coefficients, as functions of angle of attack α and Mach number M, were built with a non-linear least square best fit of the aerodynamic data of the X-34 vehicle [48, 49]. The polynomial models are in the form:

$$\begin{aligned} C_L(\alpha, M) &= P_{2,1}(\alpha) + P_{2,2}(\alpha)W_1(M) + P_{2,3}(\alpha)W_2(M) \\ C_D(\alpha, M) &= P_{3,1}(\alpha) + P_{3,2}(\alpha)W_3(M) + P_{3,3}(\alpha)W_3(M) \end{aligned} \tag{38}$$

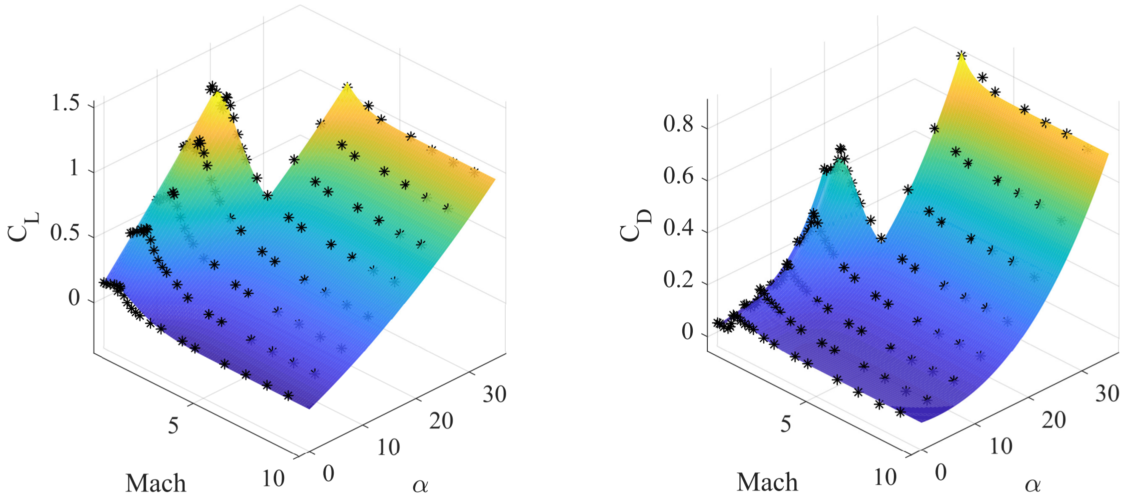
where $P_{i,j}$ is the j^{th} polynomial of degree i of α with monomial coefficients $(a_{j,0}, \cdots, a_{j,i+1})$ and W_i are Weibull distributions over Mach with parameters (ς_j, κ_j) shifted by ϱ_j, i.e.,

$$W_i = \frac{\kappa_i}{\varsigma_i} \left(\frac{M - \varrho_i}{\varsigma_i} \right)^{\kappa_i - 1} e^{-\left(\frac{M-\varrho_i}{\varsigma_i}\right)^{\kappa_i}} \tag{39}$$

The upper and lower limits on the coefficient of the Weibull functions were chosen so that they had a maximum at around $M = 1$, went to zero for $M = 0$ and for $M \to \infty$ went to zero with first derivative equal to zero. Coefficients for this aerodynamic model are given in Table 5 in the Appendix. Figure 10 shows the overall agreement between the model and the data points. The R^2 value of the non-linear fit is over 0.99 for both models. As no data was provided in the rectangular area defined by $\alpha \geq 20°$ and $M \leq 3$, the smooth constraint

$$\left(\frac{\alpha - 35}{15}\right)^8 + \left(\frac{M - 30}{27}\right)^8 - 1 \leq 0 \tag{40}$$

was imposed to exclude that area. The additional path constraint $M \leq 0.3$ was imposed on all trajectories to exclude Mach numbers for which the models extrapolated poorly. As the vehicle is not expected to fly in either of these conditions, these constraints do not affect the optimality of the results.



(a)

(b)

Fig. 10 Data points, black dots, and non-linear fit surfaces of the aerodynamic coefficients: a) C_L as a function of M and α and b) C_D as a function of M and α.

2. Wing mass model

In order to account for the change in mass due to a change of wing surface, the following mass relationship from Rohrschneider [50] was used:

$$m_{wing} = \left(N_z m_{land} \frac{1}{1 + \eta \frac{S_{body}}{S_{exp}}}\right)^{0.386} \left(\frac{S_{exp}}{t_{root}}\right)^{0.572} \left(K_{wing} b_{str}^{0.572} + K_{ct} b_{body}^{0.572}\right) \tag{41}$$

where N_z is the ultimate load factor, m_{land} is the landed mass, S_{body} is the surface of the fuselage, S_{exp} is the surface of the exposed part of the wing, t_{root} is the wing thickness at the root, b_{str} is the structural wing span at half chord line, b_{body} is the fuselage width and η, K_{wing}, K_{ct} are constants depending on the wing structure. The landed mass m_{land} is set equal to the unknown initial mass of the vehicle $m_{0,1}$, while the final mass was set equal to $m_{f,2} = m_{wing} + m_{str}$, where m_{str} is the structural mass of the vehicle plus the payload, without the wings, and has a fixed value of 10 t. Given the options available in [50], the following values of the constants in Eq. (41) were used: $N_z = 4.5$ corresponding to a maximum loading of 3 multiplied by the safety factor of 1.5 for the ultimate load, $\eta = 0.15$ corresponding to a control configured vehicle, $K_{wing} = 0.214$ for organic composite honeycomb wing without TPS, and $K_{ct} = 0.0267$ for integral dry carry-through of the wing beam, the lightest option available. The geometric parameters were derived from the geometry of the X-34 [48, 49]: $S_{exp} = 0.7283 S_{ref}$, $S_{body} = 0.6970 S_{ref}$, $b_{str} = 1.486\sqrt{S_{ref}}$, $b_{body} = 0.2785\sqrt{S_{ref}}$, $t_{root} = 0.087\sqrt{S_{ref}}$, where S_{ref} is a static optimisation variable.

3. Loading Constraints

As an optimisation of the gross initial mass could result in a reduction of the wing area, the following constraint on the wing loading was imposed:

$$\frac{m_{0,1}}{S_{ref}} \leq 700\,\mathrm{kg/m^2} \tag{42}$$

together with the limit on the dynamic pressure experienced by the vehicle in all three phases:

$$\frac{1}{2}\rho_a v_{rel}^2 \leq 60\,\mathrm{kPa} \tag{43}$$

In order to limit the maximum static structural stresses, the total acceleration in all three phases was constrained to be:

$$\dot{v}^2 + v^2\left(\dot{\gamma}^2 + \dot{\theta}^2\right) \leq (3g_0)^2 \tag{44}$$

The final flight path angle for the abort phase was required to be greater than −10°, and the final velocity of the abort phase had to be such that $M = 0.4$.

4. Objectives

The optimisation criteria are the minimisation of the spaceplane initial mass and the maximisation of the downrange in case of an abort. As the vehicle is flying along the equator, the downrange can be measured in terms of the angular difference in longitude. The problem can thus be formulated as

$$\min_{t_f, \mathbf{u}, S_{ref}, \gamma_0} [J_1,\ J_2]^T = \left[m_{0,1},\ -(\lambda_{f,3} - \lambda_{0,3})\right]^T \tag{45}$$

where $m_{0,1}$ indicates the mass of the vehicle at the beginning of phase 1, while $\lambda_{0,3}$ and $\lambda_{f,3}$ indicate the longitude of the vehicle at the beginning and end of phase 3 respectively.

This coupled multi-objective, multi-phase formulation allows for a robust optimisation of the ascent trajectory because the abort scenario is included in the sizing process and the expected result is an optimal trade-off between ascent and landing performance in the case of a failure.

5. Numerical settings

The first analysis performed was for $t_{fail} = 0$ s. This is the worst case scenario in which the engine does not start. In this case there are only two phases as phase 1 has 0 length. The problem was discretised with 6 elements in the first phase and 3 in the second, using polynomials of order 7 for all states and controls. MACSoc was run for 20000 calls to the objective vector, 10 agents and a maximum archive size of 10. The initialisation took between 10 seconds and 40 seconds for each agent, all of which managed to find a feasible solution directly at the initialisation stage. The whole process ran for approximately 2h.

6. Results

The set of Pareto-optimal solutions is shown in Fig. 11a. The solutions are well spread and problems (17) are solved down to an accuracy of 10^{-6} in both optimality and feasibility. The associated wing loading and downrange for all the 10 Pareto-optimal solutions are shown in Fig. 11b.

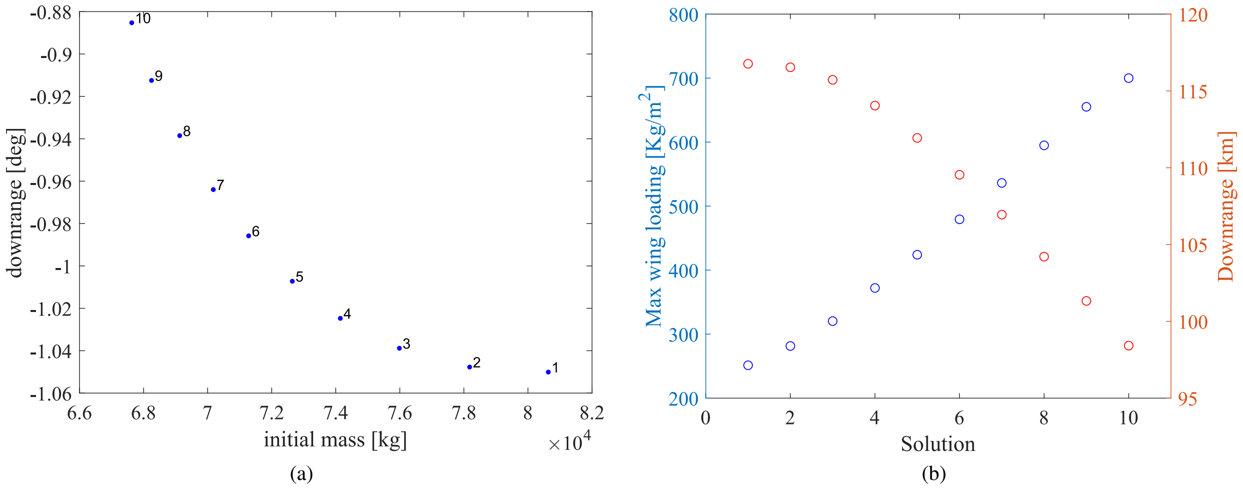
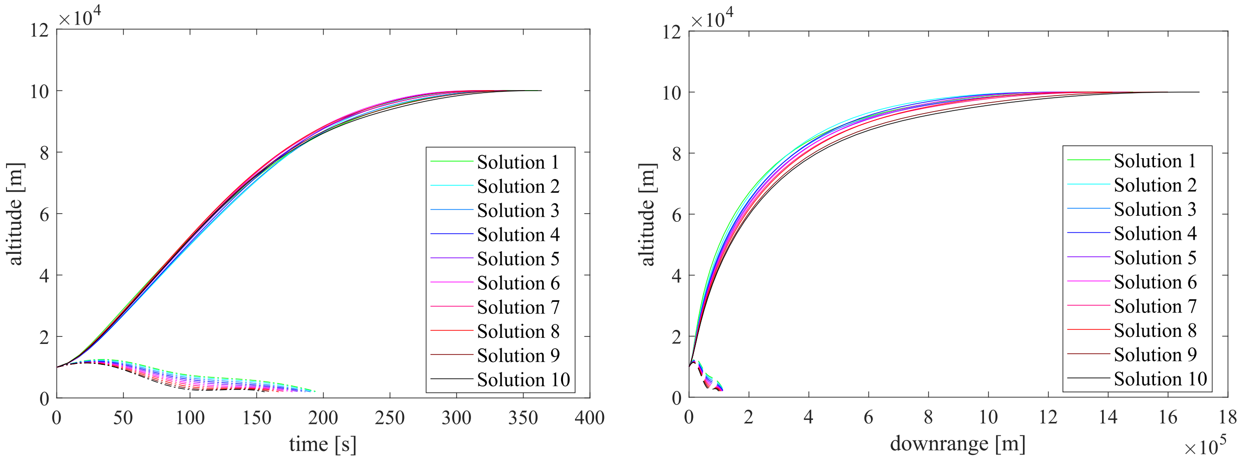


Fig. 11 Abort at 0 seconds: a) Pareto front and b) wing loading vs downrange.

As the ascent and abort trajectories start concurrently, the main trade-off is on the vehicle design and aerodynamic performance. The wing loading varies inversely to the downrange, as expected. This has an effect on the flight path angle and the throttle, which in turn affect the heat flux and the total acceleration. The result is that all solutions reach the maximum possible initial flight path angle.

High downrange solutions favour larger wing areas, which translates into a higher initial mass, due to the increased drag and higher mass of the wings themselves. However, the increase in wing area, is such that the wing loading actually decreases. Thus, solutions with larger wings have a lower wing loading and are able to generate longer downrange abort trajectories.

Figure 12 shows the ascent and abort trajectory profiles over time and downrange. Figure 13a shows the time history



(a)

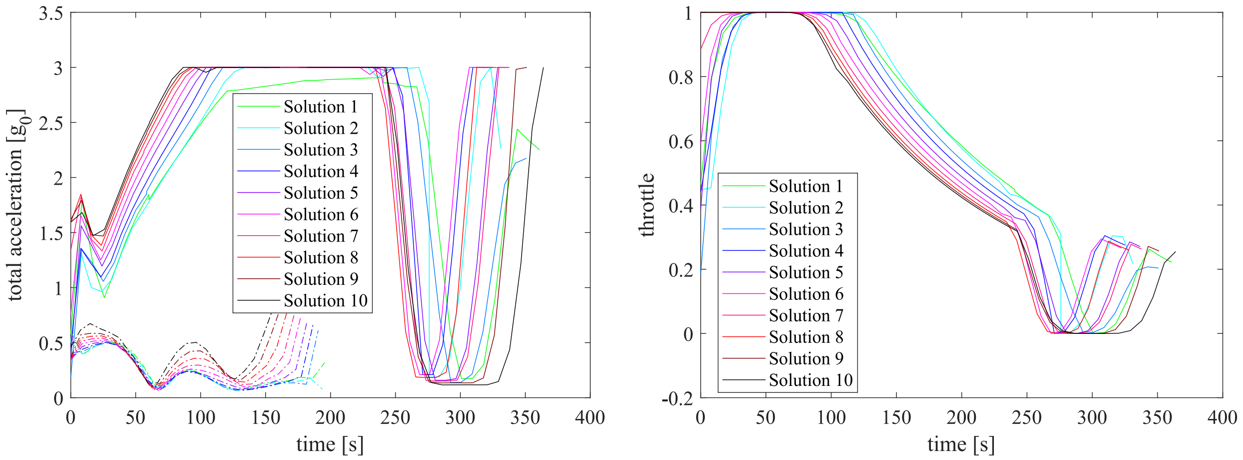
(b)

Fig. 12 Abort at 0 seconds: a) altitude vs time and b) altitude vs downrange for ascent (solid lines) and descent (dashed lines).

of the total acceleration experienced by the vehicle. All ascent phases are characterised by a maximum acceleration region in approximately the same time interval. The throttle profile, shown in Fig. 13b, differs from minimum mass solutions to maximum downrange solutions. For minimum mass solutions (small wing area) the throttle starts at maximum and remains there for a period of time, then progressively reduces down to zero, to comply with the limits on the accelerations, before finally increasing again to inject the spaceplane into orbit. Maximum downrange solutions, instead, favour a more gradual increase of the thrust and a better use of the aerodynamics.

The same analysis was then repeated for abort points at 5, 10, 15, 20, 25, 30, 35 and 40 seconds, changing the discretisation to 3 elements for each of the 3 phases. Figure 15a shows the corresponding approximations of the Pareto fronts. Interestingly, the different abort cases generate Pareto fronts which progressively converge to a single point at t_{fail} = 30 s. For t_{fail} between 30 s and 40 s, only one design solution exists for increasing values of the downrange. The increase of downrange is due to the increase of velocity and altitude while the size of the wings remain unchanged to keep the initial mass to the minimum. Thus, if the vehicle operates properly for at least 30 seconds, it gains so much velocity and altitude that, in the case of an abort, the downrange does not benefit from large wings. The same conclusion can be drawn from Fig. 15b that shows the wing loading of all solutions for all abort points.

The collapse of the Pareto front to a single point implies that there exists a single design solution that minimises the mass and simultaneously maximises the downrange. This design solution is both the most reliable and the most mass efficient if the abort happens between 30 s and 40 s from the drop point.



(a)

(b)

Fig. 13 Abort at 0 seconds: time histories for a) total acceleration and b) engine throttling for ascent (solid lines) and descent (dashed lines).

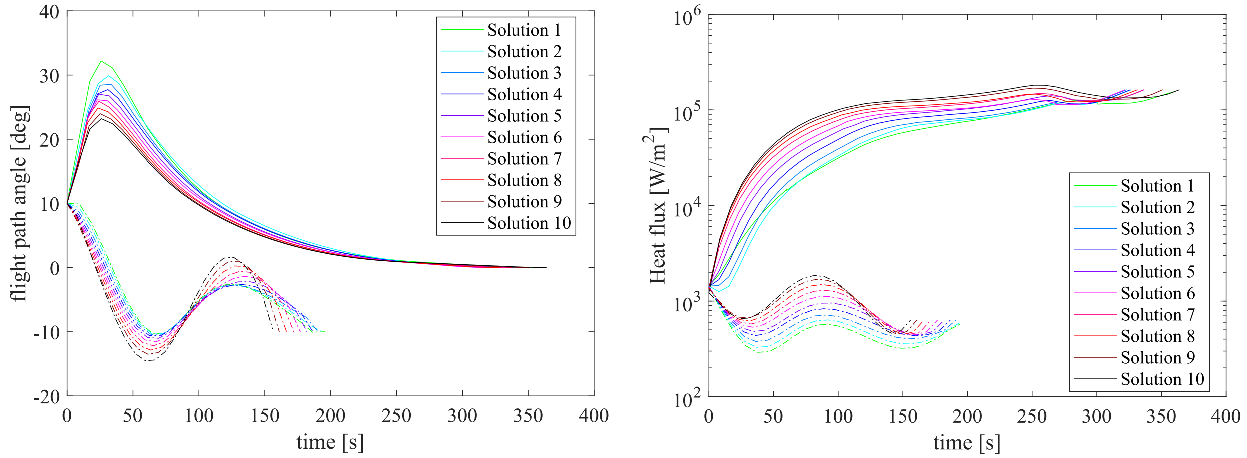
V. Conclusions

This paper presented a novel approach to the solution of multi-phase, multi-objective optimal control problems applied to the ascent, re-entry and abort scenarios for different hypothetical vehicles. The approach, combining Direct Finite Elements in Time transcription with Multi-Agent Collaborative Search, provided a robust and accurate method to compute sets of Pareto optimal solutions. In particular the smooth transition from Chebyshev to Pascoletti-Serafini scalarisation allows for a balanced and effective local refinement of the solutions and a global exploration of the search space.

The approach was first validated on a known case from the literature confirming the ability to accurately identify the optimal values for each individual objective function and to reconstruct a well spread set of locally Pareto optimal solutions. The application to the three objective case demonstrated the ability of the algorithm to generate a well spread Pareto front even in the case of more than two objectives. Furthermore, the solutions in the Pareto set give a unique insight on the impact of system design choices and an optimal trade-off between system sizing and control law, allowing the decision maker to make entirely different strategic choices depending on what is considered more important.

The abort scenario case provided an unexpected result that could not be derived from a single objective optimal control formulation of the problem: the trade-off on the wing loading affects the ascent trajectory as the aerodynamics of the vehicle changes due to the need to improve flight performance during descent, however this is true up to a limit abort time of about 30 s. Beyond that point there is no trade-off and only one optimal configuration exists because a high wing surface solution is also heavier, and the increase in gliding capabilities does not compensate for the increase of mass and thus the lower starting velocity and altitude of the abort phase.

Future work will consider more complex scenarios and a wider range of problems. For the way the approach to the



(a) (b)

Fig. 14 Abort at 0 seconds: time histories of a) flight path angle and b) heat flux for ascent (solid lines) and descent (dashed lines).

solution of multi-objective optimal control problems is devised any direct transcription method is applicable and can be paired to the MACSoc solver. Future implementations will consider these additional pairings with other transcription methods.

Appendix

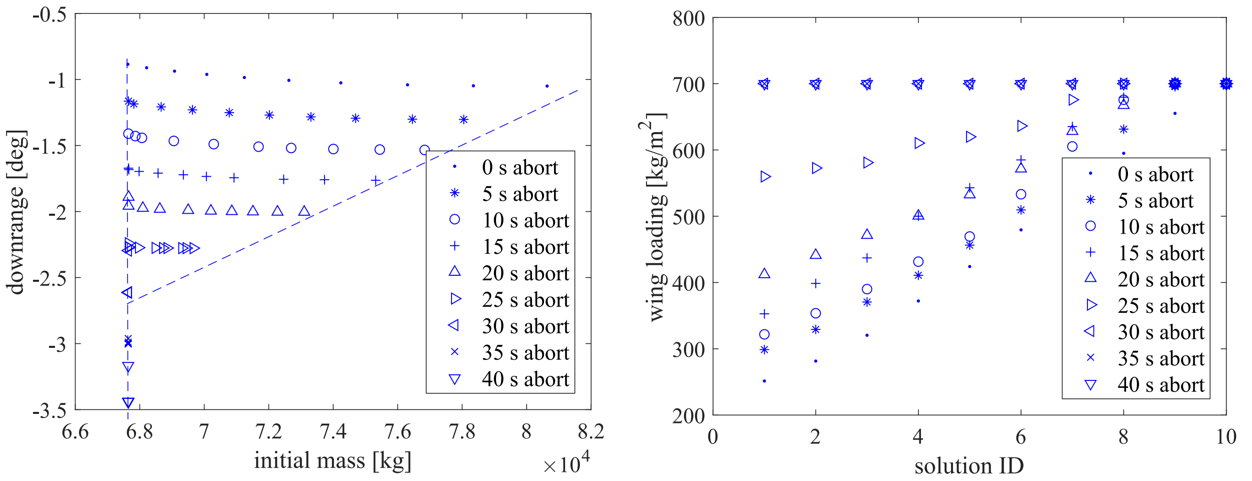
Table 5 reports the coefficients used for the aerodynamic model for the Optimal Ascent and Abort Scenarios case.

Table 5 Coefficients for the aerodynamic model

Coeff.	Value	Coeff.	Value	Coeff.	Value
$a_{1,0}$	-6.378335936032101e-02	ς_1	1.344774373794846e+00	$a_{5,3}$	-1.683702441539749e-04
$a_{1,1}$	1.976923220591986e-02	ς_2	1.542614382500486e+00	$a_{6,0}$	8.869412111210242e+00
$a_{1,2}$	2.973963976446931e-04	ϱ_1	-3.100574615791786e+01	$a_{6,1}$	7.482402265848023e-01
$a_{2,0}$	-3.349264465313049e+05	ϱ_2	-9.123166455265215e-02	$a_{6,2}$	2.604898066115986e-02
$a_{2,1}$	6.001178833841350e+05	$a_{4,0}$	4.941304084167961e-02	$a_{6,3}$	2.689898581212407e-04
$a_{2,2}$	-2.862064269172794e+04	$a_{4,1}$	-2.771101541363621e-03	κ_3	1.348770573128532e+00
$a_{3,0}$	4.398289382706495e-01	$a_{4,2}$	4.195518696508440e-04	κ_4	1.376496163657956e+00
$a_{3,1}$	6.555024080116524e-02	$a_{4,3}$	6.144719928912810e-06	ς_3	1.004292397340135e+00
$a_{3,2}$	-4.816546419307238e-04	$a_{5,0}$	-8.408713897701869e+00	ς_4	1.053245533842992e+00
κ_1	1.088199979257513e+00	$a_{5,1}$	-6.942588805953275e-01	ϱ_3	-2.715562764314760e-01
κ_2	1.523372394108941e+00	$a_{5,2}$	-2.602381527566895e-02	ϱ_4	-3.464086848960685e-01

Acknowledgments

The research was partially funded by an ESA NPI grant (ref TEC-ECN-SoW-20140806) and Airbus Defence & Space. The authors would like to thank Orbital Access Limited and BAE Systems for the mass and structural models.



(a)

(b)

Fig. 15 Comparison of Pareto optimal solutions for different abort times: a) downrange versus initial mass, b) wing loading for each solution ID.

References

[1] Stryk, O. V., "Numerical Solution of Optimal Control Problems by Direct Collocation," *Optimal Control Calculus of Variation, Optimal Control Theory and Numerical Methods*, Birkhauser Verlag, 1993.

[2] Herman, A. L., and Conway, B. A., "Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 592–599. doi:10.2514/3.21662.

[3] Vasile, M., "Robust Optimization of Trajectory Intercepting Dangerous NEO," *Proceedings of the AAS/AIAA Aerodynamics Specialist Conference*, Monterey, California, Aug 2002.

[4] Vasile, M., and Bernelli-Zazzera, F., "Optimizing Low-Thrust and Gravity Assist Maneuvres to Design Interplanetary Trajectories," *The Journal of the Astronautical Sciences*, Vol. 51, No. 1, 2003. January-March 2003.

[5] Betts, J. T., *Practical methods for optimal control and estimation using nonlinear programming*, Advances in Design and Control, SIAM, 2010.

[6] Vasile, M., "Finite Elements in Time: A Direct Transcription Method for Optimal Control Problems," *AIAA/AAS Astrodynamics Specialist Conference, Guidance, Navigation, and Control and Co-located Conferences*, Toronto, Canada, 2-5 Aug 2010. doi:0.2514/6.2010-8275.

[7] Garg, D. A., Patterson, M., Hagger, W., Rao, A. V., Benson, D. A., and Huntington, G. T., "A Unified Framework for the Numerical Solution of Optimal Control Problem Using Pseudospectral Methods," *Automatica*, 2010.

[8] Ross, I. M., and Karpenko, M., "A review of pseudospectral optimal control: From theory to flight," *Annual Reviews in Control*, Vol. 36, No. 2, 2012, pp. 182–197. doi:10.1016/j.arcontrol.2012.09.002.

[9] Zhu, Q. J., “Hamiltonian Necessary Conditions for a Multiobjective Optimal Control Problem with Endpoint Constraints,” *SIAM Journal on Control and Optimization*, Vol. 39, No. 1, 2000, p. 97–112. doi:10.1137/S0363012999350821.

[10] de Oliveira, V., Silva, G., and Rojas-Medar, M., “A class of multiobjective control problems,” *Optimal Control Applications and Methods*, Vol. 30, 2009, pp. 77–86. doi:10.1002/oca.863.

[11] Kien, B., and N.C. Wong, J. Y., “Necessary conditions for multiobjective optimal control problems with free end-time,” *SIAM Journal on Control and Optimization*, Vol. 47, No. 5, 2010, pp. 2251–2274. doi:10.1137/080714683.

[12] Oliveira, V. A., and Silva, G. N., “On Sufficient Optimality Conditions for Multiobjective Control Problems,” *Journal of Global Optimization*, Vol. 64, No. 4, 2016, pp. 721–744. doi:10.1007/s10898-015-0351-y.

[13] Ngo, T.-N., and Hayek, N., “Necessary conditions of Pareto optimality for multiobjective optimal control problems under constraints,” *Optimization*, Vol. 66, No. 2, 2017, pp. 149–177. doi:10.1080/02331934.2016.1261349.

[14] Ober-Blöbaum, S., Ringkamp, M., and zum Felde, G., “Solving multiobjective optimal control problems in space mission design using discrete mechanics and reference point techniques,” *51st IEEE Annual Conference on Decision and Control*, Maui, Hawaii, 10-13 Dec 2012, pp. 5711–5716. doi:10.1109/CDC.2012.6426285.

[15] Kaya, C. Y., and Maurer, H., “A numerical method for nonconvex multi-objective optimal control problems,” *Computational Optimization and Applications*, Vol. 57, No. 3, 2014, pp. 685–702. doi:10.1007/s10589-013-9603-2.

[16] Pascoletti, A., and Serafini, P., “Scalarizing vector optimization problems,” *Journal of Optimization Theory and Applications*, Vol. 42, No. 4, 1984, pp. 499–524. doi:10.1007/BF00934564.

[17] Pagano, A., and Mooij, E., “Global launcher trajectory optimization for lunar base settlement,” *AIAA/AAS Astrodynamics Specialist Conference, Guidance, Navigation, and Control and Co-located Conferences*, Toronto, Canada, 2-5 Aug 2010. doi:10.2514/6.2010-8387.

[18] Bairstow, B., de Weck, O., and Sobieszczanski-Sobieski, J., “Multiobjective Optimization of Two-Stage Rockets for Earth-To-Orbit Launch,” *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2006, p. 1720.

[19] Roshanian, J., Bataleblu, A. A., and Ebrahimi, M., “Robust ascent trajectory design and optimization of a typical launch vehicle,” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 2018. doi:10.1177/0954406217753460.

[20] Coverstone-Carroll, V., Hartmann, J. W., and Mason, W. J., “Optimal multi-objective low-thrust spacecraft trajectories,” *Computer methods in applied mechanics and engineering*, Vol. 186, No. 2, 2000, pp. 387–402. doi:10.1016/S0045-7825(99)00393-X.

[21] Englander, J. A., Vavrina, M. A., and Ghosh, A. R., “Multi-Objective Hybrid Optimal Control for Multiple-Flyby Low-Thrust Mission Design,” *AAS/AIAA Space Flight Mechanics Meeting*, Williamsburg, Virginia, 11-15 Jan 2015.

[22] Leary, R. H., “Global Optimization on Funneling Landscapes,” *Journal of Global Optimization*, Vol. 18, No. 4, 2000, pp. 367–383. doi:10.1023/A:1026500301312.

[23] Ricciardi, L. A., and Vasile, M., “Improved archiving and search strategies for Multi Agent Collaborative Search,” *International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems (EUROGEN)*, Glasgow, UK, 14-16 Sep 2015.

[24] Vasile, M., and Finzi, A. E., “Direct lunar descent optimisation by finite elements in time approach,” *International Journal of Mechanics and Control*, Vol. 1, No. 1, 2000.

[25] Vasile, M., and Bernelli-Zazzera, F., “Targeting a heliocentric orbit combining low-thrust propulsion and gravity assist manoeuvres,” *Operational Research in Space & Air*, Vol. 79, 2003.

[26] Zuiani, F., Kawakatsu, Y., and Vasile, M., “Multi-objective optimisation of many-revolution, low-thrust orbit raising for Destiny mission,” *23rd AAS/AIAA Space Flight Mechanics Conference*, Kauai, Hawaii, 10-14 Feb 2013.

[27] Ricciardi, L. A., Vasile, M., Toso, F., and Maddock, C. A., “Multi-Objective Optimal Control of the Ascent Trajectories of Launch Vehicles,” *AIAA/AAS Astrodynamics Specialist Conference*, Long Beach, California, 2016. doi:10.2514/6.2016-5669.

[28] Vasile, M., and Ricciardi, L., “A direct memetic approach to the solution of Multi-Objective Optimal Control Problems,” *IEEE Symposium Series on Computational Intelligence (SSCI)*, Athens, Greece, 6-9 Dec 2016. doi:10.1109/SSCI.2016.7850103.

[29] Graichen, K., and Petit, N., “Solving the Goddard problem with thrust and dynamic pressure constraints using saturation functions,” *17th International Federation of Automatic Control (IFAC) World Congress*, July 2008. doi:10.3182/20080706-5-KR-1001.02423.

[30] Preller, D., and Smart, M. K., “Reusable Launch of Small Satellites Using Scramjets,” *Journal of Spacecraft and Rockets*, 2017, pp. 1–13.

[31] Tsuchiya, T., and Mori, T., “Optimal design of two-stage-to-orbit space planes with airbreathing engines,” *Journal of Spacecraft and Rockets*, Vol. 42, No. 1, 2005, pp. 90–97.

[32] Yang, S., Cui, T., Hao, X., and Yu, D., “Trajectory optimization for a ramjet-powered vehicle in ascent phase via the Gauss pseudospectral method,” *Aerospace Science and Technology*, Vol. 67, 2017, pp. 88–95.

[33] Bayley, D. J., Hartfield, R. J., Burkhalter, J. E., and Jenkins, R. M., “Design optimization of a space launch vehicle using a genetic algorithm,” *Journal of Spacecraft and Rockets*, Vol. 45, No. 4, 2008, pp. 733–740.

[34] Gath, P. F., and Calise, A. J., “Optimization of Launch Vehicle Ascent Trajectories with Path Constraints and Coast Arcs,” *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 296–304. doi:10.2514/2.4712.

[35] Dukeman, G., “Atmospheric Ascent Guidance for Rocket-Powered Launch Vehicles,” *AIAA Guidance, Navigation, and Control Conference and Exhibit, Guidance, Navigation, and Control and Co-located Conferences*, Monterey, USA, 5-8 August 2002. doi:10.2514/6.2002-4559.

[36] Lu, P., Sun, H., and Tsai, B., "Closed-Loop Endoatmospheric Ascent Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, 2003, pp. 283–294. doi:10.2514/2.5045.

[37] Calise, A. J., and Brandt, N., "Generation of Launch Vehicle Abort Trajectories Using a Hybrid Optimization Method," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 6, 2004, pp. 930–937. doi:10.2514/1.7989.

[38] Dukeman, G., and Hill, A., "Rapid Trajectory Optimization for the ARES I Launch Vehicle," *AIAA Guidance, Navigation, and Control Conference and Exhibit, Guidance, Navigation, and Control and Co-located Conferences*, Honolulu, USA, 18-21 August 2008. doi:10.2514/6.2008-6288.

[39] Lu, P., and Pan, B., "Highly Constrained Optimal Launch Ascent Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, 2010, pp. 404–414. doi:10.2514/1.45632.

[40] Hodges, D. H., and Bless, R. R., "Weak Hamiltonian finite element method for optimal control problems," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 1, 1991, pp. 148–156. doi:10.2514/3.20616.

[41] Bottasso, C. L., and Ragazzi, A., "Finite element and Runge-Kutta methods for boundary-value and optimal control problems," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 4, 2000, pp. 749–751. doi:10.2514/2.4595.

[42] Hillermeier, C., *Nonlinear Multiobjective Optimization*, International Series of Numerical Mathematics, Birkhäuser Basel, 2001. doi:10.1007/978-3-0348-8280-4.

[43] Ricciardi, L. A., and Vasile, M., "Global Solution of Multi-objective Optimal Control Problems with Multi Agent Collaborative Search and Direct Finite Elements Transcription," *IEEE World Congress on Computational Intelligence, IEEE Congress on Evolutionary Computation*, Vancouver, Canada, 24-29 Jul 2016. doi:10.1109/CEC.2016.7743882.

[44] Eichfelder, G., *Adaptive Scalarization Methods in Multiobjective Optimization*, Springer-Verlag Berlin Heidelberg, 2008. doi:10.1007/978-3-540-79159-1.

[45] Chasalow, S. D., and Brand, R. J., "Algorithm AS 299: Generation of Simplex Lattice Points," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Vol. 44, No. 4, 1995, pp. 534–545. doi:10.2307/2986144.

[46] Maddock, C., Toso, F., Ricciardi, L., Mogavero, A., Lo, K. H., Rengarajan, S., Kontis, K., Milne, A., Evans, D., West, M., et al., "Vehicle and mission design of a future small payload launcher," *21st AIAA International Space Planes and Hypersonics Technologies Conference*, Xiamen, China, 6-9 Mar 2017. doi:10.2514/6.2017-2224.

[47] Toso, F., and Maddock, C., "Return and abort trajectory optimisation for reusable launch vehicles," *21st AIAA/IAA International Space Planes and Hypersonic Systems and Technologies Conferences*, Xiamen, China, 6-9 Mar 2017. doi:10.2514/6.2017-2250.

[48] Pamadi, B. N., and Brauckmann, G. J., "Aerodynamic characteristics and development of the aerodynamic database of the X-34 reusable launch vehicle," *International Symposium on Atmospheric Reentry Vehicles and Systems*, Arcachon, France, 16-18 Mar 1999.

[49] Brauckmann, G. J., “X-34 Vehicle Aerodynamic Characteristics,” *Journal of Spacecraft and Rockets*, Vol. 36, No. 2, 1999, pp. 229–239. doi:10.2514/2.3453.

[50] Rohrschneider, R., “Development of a mass estimating relationship database for launch vehicle conceptual design,” Tech. rep., School of Aerospace Engineering, Georgia Institute of Technology, 2002.