Strathprints Institutional Repository

Global Solution of Multi-objective Optimal Control Problems with Multi Agent Collaborative Search and Direct Finite Elements Transcription

Lorenzo A. Ricciardi
Department of Mechanical and
Aerospace Engineering
University of Strathclyde
Glasgow, Scotland
Email: lorenzo.ricciardi@strath.ac.uk

Massimiliano Vasile
Department of Mechanical and
Aerospace Engineering
University of Strathclyde
Glasgow, Scotland
Email: massimiliano.vasile@strath.ac.uk

Christie Maddock
Department of Mechanical and
Aerospace Engineering
University of Strathclyde
Glasgow, Scotland
Email: christie.maddock@strath.ac.uk

Abstract—This paper addresses the solution of optimal control problems with multiple and possibly conflicting objective functions. The solution strategy is based on the integration of Direct Finite Elements in Time (DFET) transcription into the Multi Agent Collaborative Search (MACS) framework. Multi Agent Collaborative Search is a memetic algorithm in which a population of agents performs a set of individual and social actions looking for the Pareto front. Direct Finite Elements in Time transcribe an optimal control problem into a constrained Non-linear Programming Problem (NLP) by collocating states and controls on spectral bases. MACS operates directly on the NLP problem and generates nearly-feasible trial solutions which are then submitted to a NLP solver. If the NLP solver converges to a feasible solution, an updated solution for the control parameters is returned to MACS, along with the corresponding value of the objective functions. Both the updated guess and the objective function values will be used by MACS to generate new trial solutions and converge, as uniformly as possible, to the Pareto front. To demonstrate the applicability of this strategy, the paper presents the solution of the multi-objective extensions of two well-known space related optimal control problems: the Goddard Rocket problem, and the maximum energy orbit rise problem.

I. INTRODUCTION

Optimal control theory is a part of mathematical optimisation that is concerned with finding optimal control policies. The control policy is generally required to regulate the evolution of a process or bring a system to a desired state by minimising a cost functional and satisfying a set of constraints. It is customary to classify methods to solve optimal control problems in two classes: indirect and direct methods. Indirect methods explicitly derive the necessary conditions for optimality from Pontryagin's maximum principle. The necessary conditions for optimality are a set of algebraic differential equations (ADE) that govern the time evolution of the states, the controls and the adjoint variables (or co-states) [1]. The solution of this coupled system of ADEs provides an accurate, yet local, solution to the optimal control problem.

Direct methods instead transcribe the optimal control problem into a non-linear programming problem (NLP) and look for a solution that satisfies the resulting set of constraints and minimises the objective function[2]. Direct methods do not require the derivation of the necessary conditions of optimality for every specific optimal control problem and do not need a first guess for the co-states. The higher flexibility of direct methods makes them easy to integrate into any optimisation scheme and to apply any optimisation algorithm without the need of an ad hoc formulation of the problem.

Most of the methods for optimal control problems that can be found in the literature solve problems with a single cost functional. However, in many cases it is desirable to have a full set of solutions that are optimal with respect to a number of conflicting cost functions.

In Coverstone et al. [3] the authors combined Genetic Algorithms and optimal control theory in a dual loop algorithm. In the outer loop, NSGAII 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. In Ober-Blobaum et al. a direct transcription approach is used, coupled with an approach that scalarises the multi-objective vector along directions pointing at predefined unreachable points in the criteria space. Each scalar problem is then solved with a standard NLP solver. In [4] a similar approach is proposed that uses a smoothed version of Tchebycheff scalarisation to scalarise the MOO problem. In [5] the authors proposed a dual loop algorithm in which the outer loop solves a multi-objective problem handling a set of categorical variables and the inner loop solves a set of single objective constrained optimal control problems using Monotonic Basin Hopping.

This paper proposes an evolutionary-based approach to find a set of Pareto optimal control policies that satisfy a set of dynamic and algebraic constraints. The proposed approach presents two main novelties: i) Pareto optimal solutions are globally sought in the parameter space with a memetic approach based on the Multi-Agent Colaborative Search framework [6]; ii) the optimal control problem is first transcribed into a non-linear Programming Problem with Direct Finite Element Transcritpion (DFET) [7] and then

into a bi-level optimisation problem where constraints satisfaction and multi-objective optimisation are decoupled and approached separately. Although gradient based approaches are well established and give necessary conditions for optimality, the evaluation or approximation of the gradients (and especially Hessians, if required) may be too expensive. Moreover, gradient based methods are local in nature, thus a global approach might find better solutions. Finally, the separate handling of the constraints avoids the necessity to penalise their violation by artificially adding this penalisation to the objective functions. This strategy has often proved to be problematic, as penalisation requires some form of weighting, and a bad choice of weights can greatly hinder the convergence of any method to the true solution.

II. Problem Formulation

This paper is concerned with the following multi-objective optimal control problem:

$$
\begin{aligned}
&\min_{\mathbf{u}\in U}\mathbf{J} = [J_1, J_2, ..., J_i..., J_m]^T \\
&s.t. \\
&\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u}, t) \\
&\mathbf{g}(\mathbf{x}, \mathbf{u}, t) \geq 0 \\
&\boldsymbol{\psi}(\mathbf{x}_0, \mathbf{x}_f, t_0, t_f) \geq 0 \\
&t \in [t_0, t_f]
\end{aligned}
\tag{1}
$$

where $\mathbf{J}$ is a vector of objectives J_i, that are functions of the state variable $\mathbf{x} : [t_0, t_f] \to \mathbb{R}^n$, control variable $\mathbf{u} \in L^\infty$ and time t. The functions $\mathbf{x}$ belong to the Sobolev space $W^{1,\infty}$ while the objective functions are $J_i : \mathbb{R}^{n+2} \times \mathbb{R}^p \times [t_0, t_f] \longrightarrow \mathbb{R}$. The objective vector is subject to a set of dynamic constraints with $\mathbf{F} : \mathbb{R}^n \times \mathbb{R}^p \times [t_0, t_f] \longrightarrow \mathbb{R}^n$, algebraic constraints $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^p \times [t_0, t_f] \longrightarrow \mathbb{R}^s$, and boundary conditions $\mathbb{R}^{2n+2} \longrightarrow \mathbb{R}^q$. Note that problem (1) generally is non-smooth and can include a number of additional static parameters.

III. Problem Transcription

Problem (1) is here transcribed into a multi-objective nonlinear programming problem via DFET. DFET was initially proposed in [8] and uses finite elements in time on spectral bases to transcribe the differential equations into a set of algebraic equations. Finite Elements in Time for the indirect solution of optimal control problems were initially proposed by Hodges et al. in [9], and during the late 1990s evolved to the discontinuous version. As pointed out by Bottasso et al. in [10], 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 [11], to the Sun [12], to the design of WSB transfers to the Moon, low-thrust transfers in the restricted three body problem and optimal landing trajectories to the Moon [8].

For each individual cost function consider the following Bolza's problem:

$$
\min_{\mathbf{u}\in U}J_i = \alpha_i\phi_i(\mathbf{x_0}, \mathbf{x_f}, t_0, t_f) + \beta_i \int_{t_0}^{t_f} L_i(\mathbf{x}, \mathbf{u}, t)dt
\tag{2}
$$

where α_i and β_i are positive weights. In multi-objective optimisation this formulation corresponds to a weighted sum scalarisation, which is known to be unable to represent points on non-convex regions of the Pareto front. Therefore, to avoid this problem, in this paper only cases with $(\alpha_i = 1, \beta_i = 0)$ or $(\alpha_i = 0, \beta_i = 1)$ are considered. The differential constraints can be recast in weak form and integrated by parts, leading to

$$
\int_{t_0}^{t_f} \dot{\mathbf{w}}^T\mathbf{x} + \mathbf{w}^T\mathbf{F}(\mathbf{x}, \mathbf{u}, t)dt - \mathbf{w}_f^T\mathbf{x}_f^b + \mathbf{w}_0^T\mathbf{x}_0^b = 0
\tag{3}
$$

where $\mathbf{w}$ are the generalised weight functions and $\mathbf{x}^b$ are the boundary values of the states, that may be either imposed or free. Let the time domain D be decomposed into N finite elements such that

$$
D = \bigcup_{j=1}^N D_j(t_{j-1}, t_j)
\tag{4}
$$

and parametrise, over each D_j, the states, controls and weight functions as

$$
\mathbf{x}(t) = \overset{N}{\underset{j=1}{\mathbf{U}}}\mathbf{X}_j = \overset{N}{\underset{j=1}{\mathbf{U}}}\sum_{s=0}^l f_{sj}(t)\mathbf{x}_{sj}
\tag{5}
$$

$$
\mathbf{u}(t) = \overset{N}{\underset{j=1}{\mathbf{U}}}\mathbf{U}_j = \overset{N}{\underset{j=1}{\mathbf{U}}}\sum_{s=0}^m g_{sj}(t)\mathbf{u}_{sj}
\tag{6}
$$

$$
\mathbf{w}(t) = \overset{N}{\underset{j=1}{\mathbf{U}}}\mathbf{W}_j = \overset{N}{\underset{j=1}{\mathbf{U}}}\sum_{s=0}^{l+1} h_{sj}(t)\mathbf{w}_{sj}
\tag{7}
$$

where $\overset{N}{\underset{j=1}{\mathbf{U}}}$ denotes the juxtaposition of the polynomials defined over each sub-interval, the functions f_{sj}, g_{sj} and h_{sj} are chosen among the space of polynomials of degree l, m and $l + 1$ respectively. It is practical to define each D_j over the normalised interval $[-1, 1]$ through the transformation

$$
\tau = 2\frac{t - \frac{t_j-t_{j-1}}{2}}{t_j - t_{j-1}}
\tag{8}
$$

This way it's easy to express the polynomials f_{sj}, g_{sj} and h_{sj} as the Lagrange interpolation on the Gauss nodes in the normalised interval:

$$
f_{sj} = \prod_{k=0, k\neq s}^l \frac{\tau - \tau_k}{\tau_s - \tau_i}
\tag{9}
$$

and similarly for g_{sj} and h_{sj}. Different Gauss nodes will lead to schemes with slightly different characteristics. In this work, Gauss-Lobatto nodes will be used for the generation of the polynomials for states and weight functions, while Gauss-Legendre nodes will be used for the controls. Substituting the

definitions of the polynomials into the objective functions and integrating with Gauss quadrature formulas leads to

$$\tilde{J}_i = \alpha_i \phi_i(\mathbf{X}_0^b, \mathbf{X}_f^b, t_0, t_f) +$$
$$\beta_i \sum_{j=1}^{N} \sum_{k=1}^{l+1} \sigma_k L_i(\mathbf{X}_j(\tau_k), \mathbf{U}_j(\tau_k), \tau_k) \frac{\Delta t}{2} \quad (10)$$

and for the variational constraints leads to the system

$$\sum_{k=1}^{l+1} \sigma_k \left[\dot{\mathbf{W}}_j(\tau_k)^T \mathbf{X}_j(\tau_k) + \mathbf{W}_j(\tau_k)^T \mathbf{F}_j(\tau_k) \frac{\Delta t}{2} \right]$$
$$- \mathbf{W}_{p+1}^T \mathbf{X}_j^b + \mathbf{W}_1^T \mathbf{X}_j^b = 0 \quad (11)$$

where τ_k and σ_k are the Gauss nodes and weights, and $\mathbf{F}_j(\tau_k)$ is the shorthand notation for $\mathbf{F}(\mathbf{X}_j(\tau_k), \mathbf{U}_j(\tau_k), \tau_k)$. Gauss-Legendre weights and nodes are used for the numerical quadrature in this work, i.e., the polynomials generated through the Lagrange interpolation over the Gauss-Legendre or Gauss-Lobatto nodes are then evaluated at the Gauss-Legendre nodes over each interval. With DFET, optimal control problem (2) was transcribed into the non-linear programming problem (10), which in compact reads as:

$$\min_{\mathbf{p}} \tilde{\mathbf{J}}(\mathbf{x}_s, \mathbf{p})$$
$$s.t. \quad (12)$$
$$\mathbf{c}(\mathbf{x}_s, \mathbf{p}) \geq 0$$

where the vector $\mathbf{x}_s$ contains all the nodal values for the states and $\mathbf{p} = [\mathbf{u}_s, \mathbf{x}_0, \mathbf{x}_f, t_0, t_f]^T$ collects all the static and dynamic control variables.

IV. SOLUTION APPROACH

Problem (12) is further translated into the following two-level optimisation problem:

$$\min_{\mathbf{P}^*} \tilde{\mathbf{J}}(\mathbf{x}^*, \mathbf{p}^*)$$
$$s.t. \quad (13)$$
$$(\mathbf{x}^*, \mathbf{p}^*) = \text{argmin}\{f(\mathbf{x}_s, \mathbf{p}) | \mathbf{c}(\mathbf{x}_s, \mathbf{p}) \geq 0\}$$

and solved with a two level algorithm. The outer level generates sets of candidate solution vectors $\mathbf{p^c}$ that are submitted to an inner level. The inner level takes the candidate solutions and attempts to satisfy constraints $\mathbf{c}$. If the candidate vector $\mathbf{p^c}$ does not lead to convergence of the inner level, a penalty value L for all the objective functions is returned to the outer level, while if it converges to a feasible solution $(\mathbf{x}^*, \mathbf{p}^*)$, it returns the corresponding values $\tilde{\mathbf{J}}(\mathbf{x}^*, \mathbf{p}^*)$ for the objective functions, together with the feasible vector $(\mathbf{x}^*, \mathbf{p}^*)$. The penalty value L is chosen to be large enough that the non feasible solution will surely be worse than the objective function of any feasible solution.

Note that states are also part of the solution vector of the inner level and thus a first guess for the state vector $\mathbf{x}_s$ is also required. For the implementation in this paper, at the first iteration the state variables are initialised by simply cloning the initial condition. This was found to be simple yet sufficiently robust to provide convergence at the first iteration on the test cases presented in this paper. Then the inner level is

---

Algorithm 1 Algorithm 1: Outer level
1: Initialise control parameters $\mathbf{p}$ and search directions λ
2: Initialise state vector $\mathbf{x}_s$
3: Perform individual actions: evaluate objective vector with Algorithm 2
4: Evaluate dominance and Tchebycheff scalarisation of the population and of the archive
5: Archive the non dominated solutions
6: $\mathbf{p} \leftarrow \mathbf{p}^*, \mathbf{x}_s \leftarrow \mathbf{x}_s^*$
7: Perform social actions: evaluate objective vector with Algorithm 2
8: Evaluate dominance and Tchebycheff scalarisation of the population and of the archive
9: Archive the non dominated solutions
10: $\mathbf{p} \leftarrow \mathbf{p}^*, \mathbf{x}_s \leftarrow \mathbf{x}_s^*$
11: **if** $fun\_evals < max\_fun\_eval$ **then**
12: GoTo 3
13: **end if**

---

solved with the Matlab function *fmincon* with $f(\mathbf{x}_s, \mathbf{p}) = 1$, so that it focuses only on the satisfaction of the constraints. The outer level is solved with an implementation of Multi-Agent Collaborative Search. MACS sees the inner level as a single function evaluation returning objective function values, since the objective functions are evaluated only once the non-linear solver has returned a solution. The state variables $\mathbf{x}^*$ are returned to the outer level together with the vector $\mathbf{p}^*$ and are used to initialise the state vector of a new candidate vector $\mathbf{p}^c$. This way, after the very first iteration, the outer level has a good initial guess for the state vector, and the inner level will start from a better initial solution, with resulting higher chances of convergence and less computational effort. The algorithms for the inner and outer levels are summarised in Algorithm 1 and 2. Algorithm 1 follows the general implementation of MACS as described in [6], but includes the following modifications: the solution vector contains both states and optimisation values, so additional information is needed to tell weather a given variable is a state variable, a dynamic optimisation variable (control) or a static optimisation variable. This information is needed so that MACS knows which segments of the solution vector it can alter through its heuristics. Moreover, as explained before, the objective function evaluation also returns updated (feasible) values for the solution vector, and at the beginning, the initialisation of solution vectors requires MACS to also generate the state variables.

V. NUMERICAL TESTS

In this section we present two simple test cases to show the applicability of the present work: the first one is the Goddard Rocket problem and the second one is the maximum energy orbit rise problem. These two cases are very well known in the aerospace related optimal control community, and thus provide an interesting starting point. However, to the authors' knowledge, they have always been studied as single objective

Algorithm 2 Algorithm 2: Inner level
___
1: Initialise states and controls with $\mathbf{x}_s$ and $\mathbf{p}^c$ from outer level
2: Run *fmincon*
3: **if** $[p^*, x_s^*]$ feasible **then**
4:     Return $\mathbf{p}^*, \mathbf{x}_s^*, \mathbf{J}(\mathbf{p}^*, \mathbf{x}_s^*)$
5: **else**
6:     Return $\mathbf{J} = L$
7: **end if**
___

problems, thus a multi-objective extension could provide some interesting insight.

A. Goddard Rocket problem

The Goddard Rocket problem is a simple rocket ascent problem from a flat celestial body and no atmosphere. The control variable is the thrust angle and both the gravity and the thrust accelerations are constant. The final state is constrained so that at the final time the altitude has a specified value and the vertical component of velocity is zero. The optimal control formulation of the problem and its analytical solution can be found in [1], while a solution with DFET for single objective optimisation can be found in [7].

In the literature solutions exist for either the minimum time to reach a target altitude or the maximum horizontal component of the velocity at a target altitude. In this paper the problem is reformulated to consider the two objective functions simultaneously. Therefore, the following multi-objective extension is solved:

$$\min_{t_f, u}[J_1, J_2]^T = [t_f, -v_x(t_f)]^T \qquad (14)$$

subject to the dynamic constraints:

$$\begin{cases} \dot{x} & =v_x \\ \dot{v}_x & =a\cos(u) \\ \dot{y} & =v_y \\ \dot{v}_y & =-g+a\sin(u) \end{cases} \qquad (15)$$

where g is the gravity acceleration, a the thrust acceleration, x and y are the component of the position vector, v_x and v_y the component of the velocity vector and u the control. The dynamics is integrated from time $t=0$ to time $t=t_f$. The boundary conditions are:

$$\begin{cases} x(0)=0; & v_x(0)=0 \\ y(0)=0; & v_y(0)=0 \\ y(t_f)=\overline{h}; & v_y(t_f)=0 \end{cases} \qquad (16)$$

The parameters g, a and h were respectively set to $1.6 \cdot 10^{-3}$, $4 \cdot 10^{-3}$ and 10. Following [7], the DFET method was applied splitting the time domain into 4 elements, with polynomials of order 6 for each control and state variable. Control angles were bounded between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$, while total mission time was bounded between 100 and 250. In total there are 29 optimisation variables: 28 dynamic (i.e. the coefficients of the polynomials for the control) and 1 static (final time). The transcribed problem was then optimised with MACS2.1. A brief explanation of the settings reported in table I follows: max_fun_eval is the maximum number of objective functions evaluation, pop_size is the number of agents performing the search, ρ_ini is the initial radius for the pattern search local action, F and CR are the standard parameters for the Differential Evolution local action, p_social is the ratio of the agents also performing social actions, max_arch is the number of solutions to store in the archive, $coord\_ratio$ is the number of coordinates initially scanned by pattern search, $contr\_ratio$ is the amount the radius of pattern search contracts after the local actions fail, and max_contr_ratio is the maximum number of times the radius of pattern search can contract before it returns to the initial value. Settings reported in Table II instead refer to the parameters of *fmincon*: max_con_eval is the maximum number of constraints evaluation (for each call to the objective functions) and tol_con is the threshold under which the solution is considered to be feasible. All other *fmincon* settings are left as default.

Algorithm 1 was run 30 times to collect some statistics on its convergence behaviour as a reference for future studies (see Table III). Since the two objectives have different orders of magnitude, to compute the statistics the reference front was scaled between 0 and 1. This way the metrics are equally sensitive to both objectives. For self consistency, statistics were computed using the collective front as the reference front. Figure 1 shows the collection of all the non dominated solutions of those 30 runs, along with 4 representative solutions (marked with plusses) and the objective values computed from the analytic solutions with the same time of the representative solutions (marked with circles). The gaps in the front are only apparent, as the settings of MACS were chosen to have only 10 points in the front, and the different runs did not generate exactly the same front. To show the actual solutions of the points indicated as crosses on the Pareto front, the corresponding trajectories and time histories of the controls and velocities are plot in figures 2 and 3 to 6. The same figures also show the comparison of the

Problem	mean GD (variance)	mean IGD (variance)	average runtime (s)
Goddard	5.701e-3 (1.107e-5)	2.588e-2 (1.909e-5)	2456
Orbit	7.384e-4 (4.217e-7)	5.179e-3 (5.052e-6)	41113

time histories computed with a standard gradient based single objective optimisation approach on the same transcription for the single objective cases (i.e solved directly with *fmincon*), and with the analytic solution for the same mission time. As can be seen, the control law becomes progressively flatter as the allowed mission time increases, and the solution obtained with the present approach is very close to the solution obtained with the classical gradient based single objective formulation or with the analytic one. The control laws computed with the proposed approach however present some imperfections in the form of discontinuities or wiggles. These are due to two reasons: first, as pointed out by Vasile in [7], h-p adaptivity is required to correctly capture the steep changes in the control law for the shortest mission times, and second, wiggles are a sign of incomplete optimality, indicating that the solution does not lie on the true Pareto front. Both these considerations are evident from the given plots. Despite these imperfections which can be mitigated or eliminated with h-p adaptivity, more function evaluations or better local strategies, MACS+DFET can give a very good idea of the trade-off required to satisfy the two objectives and the resulting control laws. This approach seems thus very promising because the problem was treated as a genuinely multi-objective problem in all the aspects (i.e. without scalarising the objective functions in the optimal control formulation), and was treated in all regards as a black box (i.e. no gradient of the objective functions was supplied or required). Obviously, for some problems it could be possible to automatically compute the gradients of the objective functions with respect to the control parameters, and thus employ that precious information in the optimisation. However, a more general approach like the one adopted seems more applicable if one wishes to solve problems where it is too expensive or not evident how to compute those gradients (provided they exist at all).

B. Maximum Energy Orbit Rise

The maximum energy orbit rise is another common optimal control problem in space engineering. It's original formulation and some solution strategies can be found in [13] and [7]. In this case, a spacecraft is orbiting around a celestial body, and it is required to increase its total energy by changing its altitude and velocity. The only control variable is again the thrusting angle, and the only other force affecting the spacecraft is gravity (in this case it is considered variable with altitude, so the dynamics of the system are non-linear). In the following it is proposed, as a simple multi-objective extension, the problem
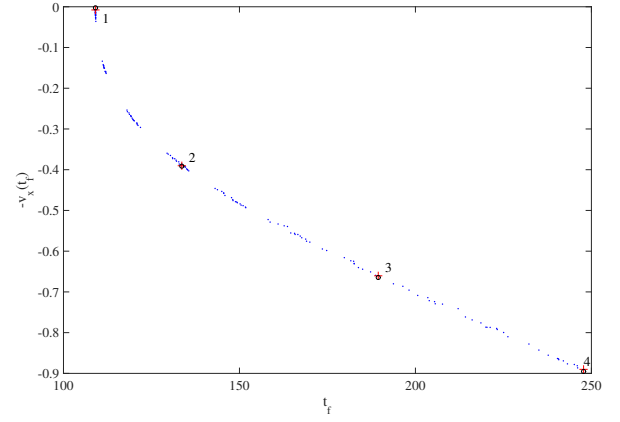

Fig. 1. Non dominated solutions of 30 different runs for the Goddard problem. Crosses indicate solutions for which trajectories, velocities and control law over time are also plotted. Circles indicate the objective values corresponding to the analytic solution with the same time as the solutions marked with crosses
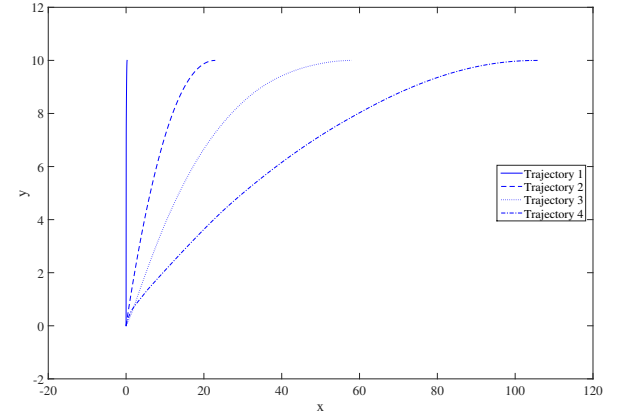

Fig. 2. Comparison of the trajectories for the 4 selected points on the Pareto front

of maximising the final energy and minimising the manoeuvre time:

$$\min_{t_f,u}[J_1, J_2]^T = \left[t_f, -\frac{\left(v_r^2(t_f) + v_t^2(t_f)\right)}{2} + \frac{1}{r(t_f)} \right]^T \quad (17)$$
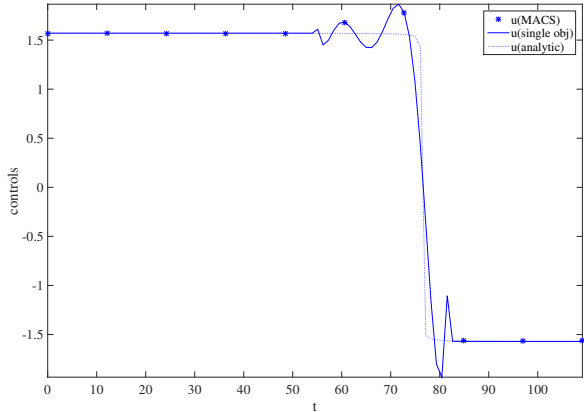
subject to the dynamic constraints:

$$\begin{cases} \dot{r} &= v_r \\ \dot{v}_r &= \frac{v_t^2}{r} - \frac{1}{r^2} + a\cos(u) \\ \dot{\theta} &= \frac{v_t}{r} \\ \dot{v}_t &= -\frac{v_t v_r}{r} + a\sin(u) \end{cases} \quad (18)$$

where r and θ are the polar coordinates of the spacecraft, v_r and v_t are the radial and tangential velocities, and a is the magnitude of the acceleration generated by the thrust. In this
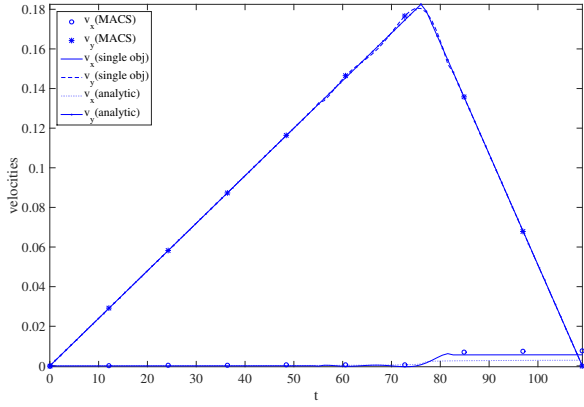
(a) Time history for the velocities

(b) Time history for the controls

Fig. 3. Time history for velocities and controls, point 1 on the Pareto front
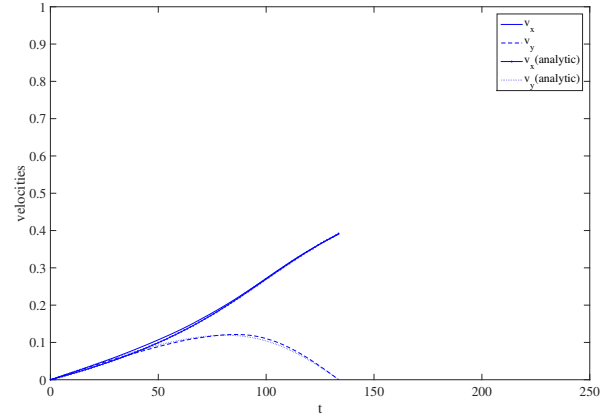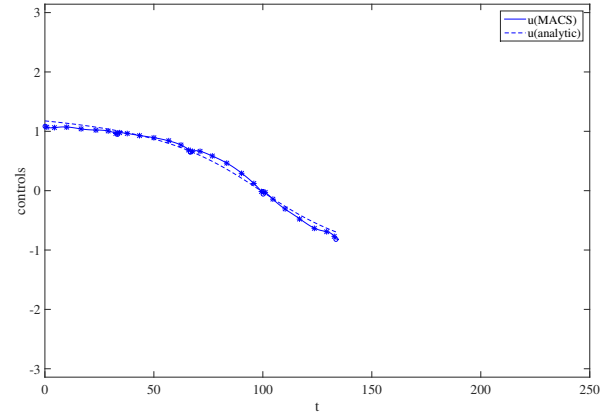

(a) Time history for the velocities

(b) Time history for the controls

Fig. 4. Time history for velocities and controls, point 2 on the Pareto front

work, $a = 1e - 2$. The boundary conditions are:

$$\begin{cases} r(0) = 1.1; & v_r(0) = 0 \\ \theta(0) = 0; & v_t(0) = \dfrac{1}{\sqrt{1.1}} \end{cases} \quad (19)$$

The DFET transcription method described in the previous section was employed, and, following [7], the time domain was subdivided into 30 elements of order 1 for each state and control variable. Control angles were bound between $-\pi$ and π, while total mission time was bounded between 20 and 80. In total there are 61 optimisation variables: 60 dynamic (i.e. the coefficients of the polynomials for the controls) ad 1 static (final time). The transcribed problem was then optimised with MACS2.1, with the same settings as in the previous case, except for max_fun_eval which was increased to 50000, tol_con which was set to $1e - 9$ and max_con_eval which was increased to 20000. The non dominated Pareto front of the combined 30 runs is reported in figure 7, while the same statistics performed in the previous case are reported in Table III. Again, gaps in this front are only due to the fact that only 10 points for each run were sought, and the different runs did not find exactly the same solutions. In this case, we plot the comparison of 4 trajectories, and the corresponding control laws in figures 8 and 9 to 12. The fact that the 4 trajectories are

overlapping and that the controls for the two extreme points on the Pareto front match very closely to the gradient based single objective solutions indicates that the whole spectrum of solutions is very close to the real Pareto front. In facts, the control laws found are basically each the continuation of the previous one, and so are the trajectories.

VI. CONCLUSION

A novel method for the solution of multi-objective optimal control problems was presented. This method has several interesting properties: the formulation of optimal control problems is genuinely multi-objective and does not require any scalarisation, no gradients of the objective functions are required because the optimisation is performed directly through a memetic algorithm, and constraints are always satisfied before objective functions are evaluated, avoiding any need to penalise constraints violations. These characteristics seem very interesting, because this way multi-objective optimal control problems can be treated as black boxes by the optimisers, thus even problems with very complex non-differentiable objectives can be tackled. The approach has been tested on the multi-objective extensions of two very common problems arising in the space sector. Although not perfect, the results seem encouraging because even with a relatively
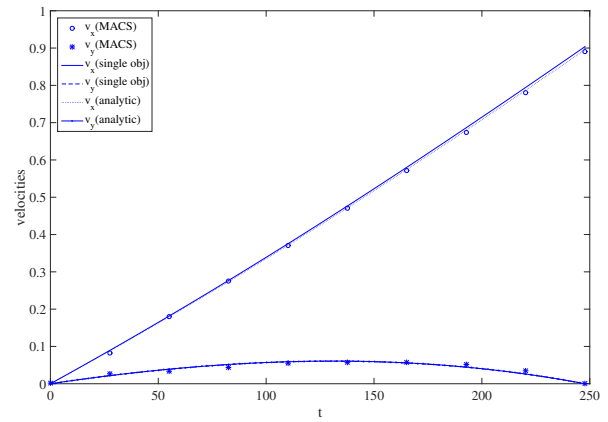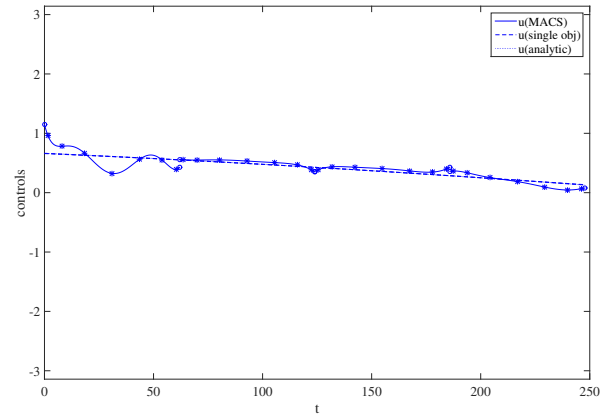
(a) Time history for the velocities

(a) Time history for the velocities
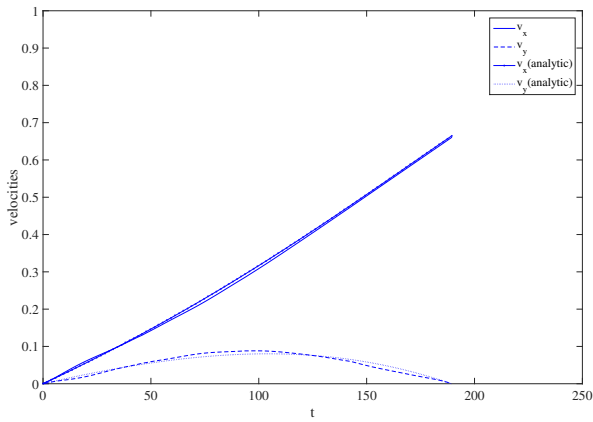
(b) Time history for the controls
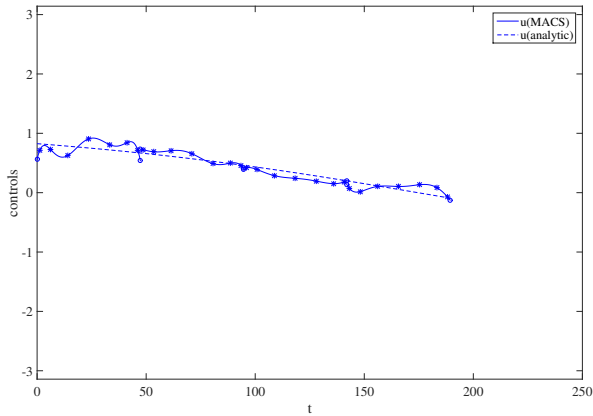
(b) Time history for the controls

Fig. 5. Time history for velocities and controls, point 3 on the Pareto front

Fig. 6. Time history for velocities and controls, point 4 on the Pareto front

low number of function evaluations and simple heuristics it was possible to obtain sensible control laws, appreciate the trade-off required to deal with the various objectives, and find solutions very similar to those computed with a gradient based single objective optimisation for the extreme points on the Pareto front or with existing analytic solutions. Further work can obviously focus on improving the local actions to get better solutions, introduce h-p adaptivity in the transcription (a feature that comes with the notable added complexity of having decision vectors of different lengths), employ more efficient programming languages and non-linear solvers, or use a different existing multi-objective optimiser algorithm for the external level.
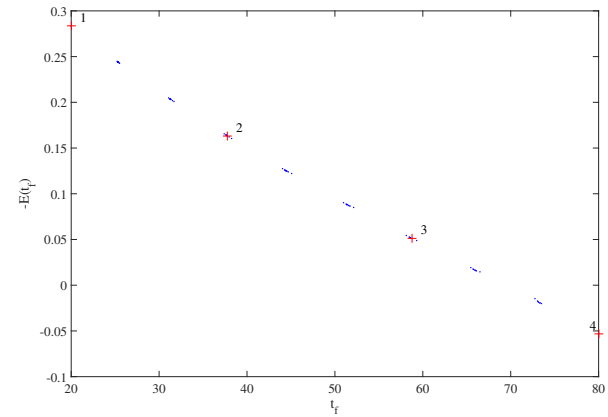
Fig. 7. Non dominated solutions of 30 different runs for the maximum energy rise problem. Crosses indicate solutions for which trajectory and control law over time are also plotted

REFERENCES

[1] A. E. Bryson, Y. Ho, *Applied optimal control: optimization, estimation and control, Revised printing*, CRC Press, 1975.

[2] John T. Betts, *Practical methods for optimal control using non-linear programming*, SIAM, Philadelphia, PA, 2001, ISBN 0-89871-488-5.

[3] V. Coverstone-Carroll, J. W. Hartmann and W. J Mason, *Optimal multi-objective low-thrust spacecraft trajectories*, Computer methods in applied mechanics and engineering, volume 186, number 2, 2000, pp 387–402

[4] C..cn Kaya, H. Maurer, *A Numerical Method for Nonconvex Multi-objective Optimal Control Problems*, Computational Optimization and Applications, 2013.

[5] J. A. Englander, M. A. Vavrina, A. R. Ghosh, Hybrid Optimal Control
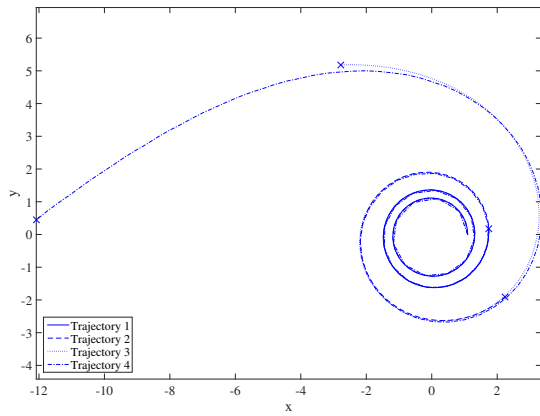
Fig. 8. Comparison of the trajectories for the 4 selected points on the Pareto front
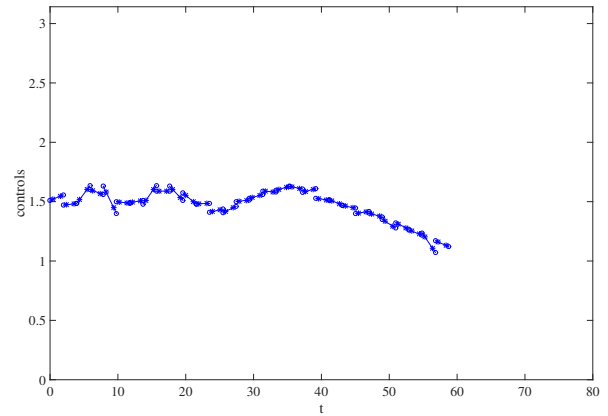

Fig. 11. Time history for total energy and controls, point 3 on the Pareto front
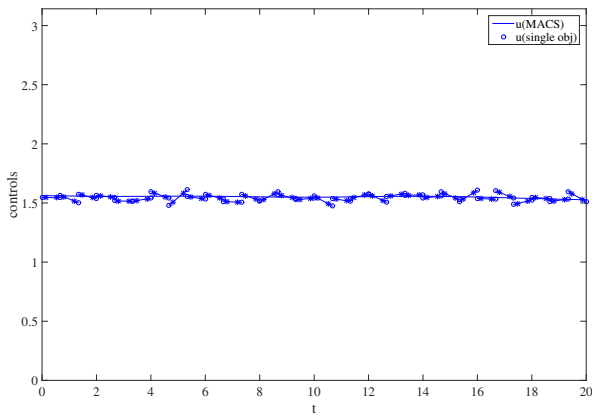

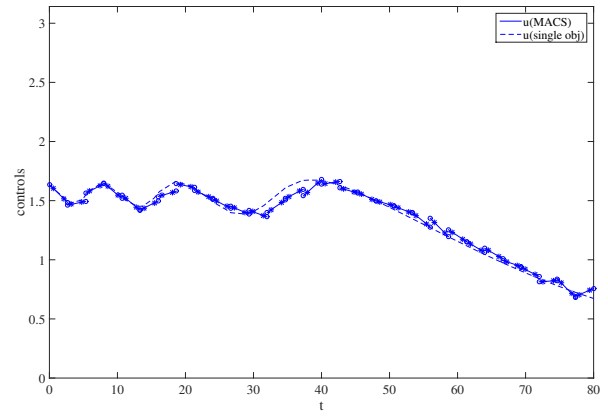Fig. 9. Time history for total energy and controls, point 1 on the Pareto front

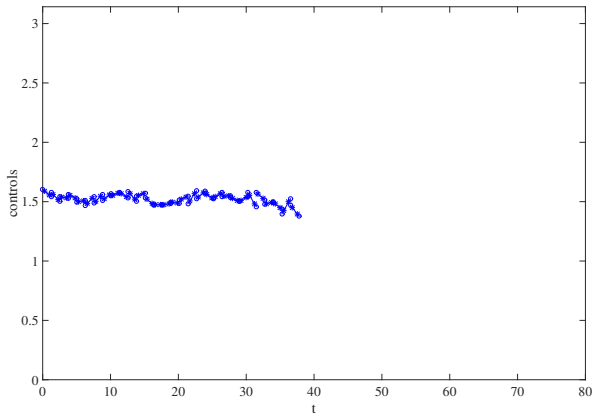Fig. 12. Time history for total energy and controls, point 4 on the Pareto front

Fig. 10. Time history for total energy and controls, point 2 on the Pareto front

for Multiple-Flyby Low-Thrust Mission Design. AAS 15-227, 25th AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA.

[6] L. A. Ricciardi and M. Vasile, *Improved archiving and search strategies for Multi Agent Collaborative Search*, Eurogen 2015

[7] M. Vasile, *Finite elements in time: a direct transcription method for optimal control problems*, AIAA 2010-8275, AIAA/AAS Astrodynamics Specialist Conference, 2 - 5 August 2010, Toronto, Ontario Canada

[8] M. Vasile and A. E. Finzi, *Direct lunar descent optimisation by finite elements in time approach*, International Journal of Mechanics and Control, volume 1, number 1, 2000.

[9] H. D. Hodges and R. R. Bless, *Weak Hamiltonian finite element method for optimal control problems*, Journal of Guidance, Control, and Dynamics, volume 14, number 1, 1991, pp 148–156.

[10] C. L. Bottasso and A. Ragazzi, *Finite element and Runge-Kutta methods for boundary-value and optimal control problems*, Journal of Guidance, Control, and Dynamics, volume 23, number 4, 2000, pp 749–751.

[11] M. Vasile and F. Bernelli-Zazzera, *Optimizing low-thrust and gravity assist manoeuvres to design interplanetary trajectories*, The Journal of the astronautical sciences, volume 51, number 1, 2003, pp 13–35.

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

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

[14] M. Vasile and S. Campagnola, *Design of Low-Thrust Gravity Assist Trajectories to Europa*, Journal of the British Interplanetary Society, volume 62, number 1, 2009, pp 15–31.

[15] M. Schlueter, C. H Yam, T. Watanabe and A. Oyama, *Many-objective optimization of interplanetary space mission trajectories*, IEEE Congress on Evolutionary Computation (CEC), 2015, pp 3256–3262

[16] S. Ober-Blobaum, M. Ringkamp and G. zum Felde, *Solving multi-objective optimal control problems in space mission design using discrete mechanics and reference point techniques*, IEEE 51st Annual Conference on Decision and Control (CDC), 2012, pp 5711–5716