Ricciardi, Lorenzo Angelo and Vasile, Massimiliano (2018) MODHOC - Multi Objective Direct Hybrid Optimal Control. In: 7th International Conference on Astrodynamics Tools and Techniques, 2018-11-06 - 2018-11-09. ,

MODHOC - MULTI OBJECTIVE DIRECT HYBRID OPTIMAL CONTROL

Lorenzo A. Ricciardi, Massimiliano Vasile

University of Strathclyde
Mechanical and Aerospace Engineering
75 Montrose Street, G1 1XJ, Glasgow, United Kingdom

ABSTRACT

MODHOC (Multi Objective Direct Hybrid Optimal Control) is a toolbox for the design, optimisation and trade off study of space systems and missions. It solves general nonlinear multi phase optimal control problems, automatically computing a well spread set of optimal trade off solutions. In addition, it is able to handle discrete optimisation parameters. In order to do so, MODHOC combines a direct transcription method based on finite elements and a global multi objective optimisation algorithm combining evolutionary heuristics and mathematical programming. MODHOC has been applied to a variety of applications: from the optimisation of launch vehicles and their ascent, abort and re entry trajectories, to the design of the optimal deployment of constellations of satellites, to the design of multi target missions. In this paper, the main elements of MODHOC are described and the application of the software in space and non space related sample problems is demonstrated.

***Index Terms*—** Optimal Control, Multi Objective optimisation, Global optimisation, Mixed integer optimisation

1. INTRODUCTION

Space systems are complex engineering systems that need to operate reliably and optimally in a harsh environment and with limited resources. Their design is a challenging endeavour involving several disciplines simultaneously: decisions are to be taken at the subsystem or vehicle level, at the trajectory level and at the mission design level, and each choice can affect several other aspects.

The design process inherently implies the presence of trade-offs: the performances of a subsystem might need to be increased in order to mitigate the demands on another subsystem, while the achievement of a more ambitious mission goal might require higher overall economic cost, time and technical complexity of the solution.

This paper presents MODHOC, an open source tool conceived specifically to help in the preliminary design stages of complex systems. The previous version of MODHOC is available under the Strath-ACE/SMART-o2c repository of Github[1]. Its main goal is to present to the decision makers a set of optimal trade off solutions, allowing them to take more informed decisions.

MODHOC combines a transcription method for nonlinear multi-phase optimal control problems, a population based memetic multi-objective optimisation algorithm and mathematical programming solvers. The transcription method allows to treat general optimal control problems, thus it can tackle problems with any kind of dynamic model. The memetic multi-objective optimisation algorithm allows for a global exploration of the search space and is able to treat problems with an arbitrary number of objectives. The mathematical programming solvers are used to refine the solutions obtained and guarantee the local optimality of the solutions found while also satisfying tight constraints.

The combination of these tools allows MODHOC to simultaneously optimise the design of a vehicle and its trajectory, returning several solutions, each striking a different balance between the high level goals set by the decision maker. MODHOC has been used in the past to perform the multi-objective optimisation of multi-stage launch vehicles, of the ascent, abort and reentering trajectories of spaceplanes, and of the deployment of constellations of satellites. [1, 2, 3, 4, 5]

In this paper, MODHOC will be applied to a three objective design and trajectory optimisation problem of a launch vehicle, a two objective trajectory optimisation from LEO to GEO orbit, and a two objective planning and scheduling problem where the order in which the targets are visited is not specified a priori.

2. MODHOC

MODHOC is composed of these three main building blocks

- DFET, Direct Finite Elements Transcription method
- MACS, Multi Agent Collaborative Search
- NLP, Nonlinear programming solvers

[1] At the repo site https://github.com/strath-ace/smart-o2c it is possible to download the optimiser MACS, the DFET transcription method and several working examples. The version currently open for public access does not handle discrete variables yet, but will be updated soon.

2.1. DFET

Direct Finite Elements in Time (DFET) is a direct transcription method to solve optimal control problems and was initially proposed by Vasile and Finzi [6] in 2000. Finite Elements in Time (FET) for the indirect solution of optimal control problems were initially proposed by Hodges and Bless [7], and during the late 1990s evolved to the discontinuous version. Borri [8] and Borri et al. [9] have shown that the resulting scheme is unconditionally stable, and if the dynamics of the system does not contain any damping term the FET formulation results in a symplectic integration scheme, thus conserving the total energy of the system. Moreover, Bottasso [10] pointed out that FET for the forward integration of ordinary differential equations are equivalent to some classes of implicit Runge-Kutta integration schemes. In addition, they can be extended to arbitrary high-order, are numerically very robust and allow full h-p adaptivity.

The basic idea of DFET is to discretise the time domain in several elements, and represent on each element both the states and controls as polynomials. The differential equations are not directly satisfied on the nodes like in the various collocation methods. Instead, they are first recast in weak form. This important step ensures that the resulting formulation is mathematically correct even in the presence of discontinuities for the dynamics within an element. The transcription process also translates the objective function and the other boundary and path constraints. The original problem is thus converted into a finite dimensional problem, which can be solved directly through one of the several NLP solvers available.

In the past decade, DFET has been successfully used to solve a range of difficult problems: from the design of low-thrust multi-gravity assist trajectories to Mercury [11] and the Sun [12], 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 [6]. More recently they have been used to perform multi-objective optimal control of spacecraft [2, 4], ascent trajectories of launchers [3], or abort trajectories of reusable launch vehicles [13, 5].

The significant flexibility of DFET has been recently exploited by Ricciardi and Vasile [14] to devise a new scheme. By using a new set of basis functions, the new DFET method is able to completely remove the spurious oscillations originating from a sharp variation of the controls, like in case of a bang-bang control solution. In addition a theorem was proved, which guarantees the satisfaction of inequality path constraints for all times if this new version of DFET is employed and the feasible region of the path constraints is convex.

All these features make DFET a very powerful and general transcription method, with unique characteristics which make it particularly suited for space applications.

2.2. MACS

MACS is a memetic algorithm to solve multi-objective optimisation problems. It was proposed some time ago to solve robust optimisation problems in space mission design [15, 16], where the location of global optima is particularly important and challenging.

In MACS, a population of virtual agents is deployed at random locations in the search space. Each agent locally explores its neighbourhood performing a set of local search actions, also named individual actions. The individual actions are Inertia, Pattern Search and Differential Evolution. Each agent performs each action sequentially until an improvement is registered. A combination of Pareto dominance and Tchebycheff scalarisation is employed to select potential improvements towards the Pareto front. Then the population as a whole performs a set of social actions, to concurrently advance towards the front. An external archive is used to store the current best representation of the Pareto set.

Previous studies by Vasile and Zuiani [17, 18, 19, 20] showed the effectiveness of this approach on different benchmark and challenging real problems, testing numerous strategies both for the individual and the social actions. MACS was successfully used for the design of space missions for the removal of space debris by means of low-thrust, many revolutions orbits, and for the design of the initial, low-thrust rising phase for the technology demonstrator mission DESTINY. Ricciardi and Vasile [21] then introduced a new archiving algorithm able to improve the spreading of the solutions on the Pareto front. This modification, together with additional modifications on the heuristics employed by MACS, allowed to further improve the results of the optimiser.

Since then, MACS has been used to solve complex optimisation problems like the optimal deployment strategies for a constellation of satellites [1], the optimisation of an asteroid deflection mission through laser ablation [22] the multi-target space debirs removal mission of the 9^{th} edition of the Global Trajectory Optimisation Competition [23], and, combined for the first time with DFET to solve multiobjective optimal control problems [2].

2.3. NLP

NLP solvers are gradient based solvers for constrained nonlinear optimisation. Several high quality, open source and commercial implementations are available. These methods are able to tackle problems with millions of variables and constraints. If the problem has a good sparsity pattern, like in case of optimal control problems, these methods are able to produce a solution in a matter of seconds to hours even on an ordinary desktop.

The main strength of NLP methods is their ability to guarantee both a tight satisfaction of the constraints and the local optimality of the solution. However, their local nature means that they require an initial guess and can converge to an optimal solution which is far from the global one. For this reason, MODHOC employs a combination of NLP solvers and MACS to synergistically leverage the respective strength of the algorithms: the NLP solvers produce fully feasible and locally optimal solutions, while MACS automatically generates guesses for the NLP, provides global search capabilities and retains a set of Pareto optimal solutions.

To solve a multi objective optimal control problem, MODHOC first transcribes it using DFET. MACS then produces random first guesses for the solutions through Latin Hypercube Sampling: these guesses are passed to the NLP solver, which tries to make them feasible by solving a feasibility problem. These solutions are passed back to MACS, which employs its dominance and Tchebychev scalarisation criteria to evaluate the quality of the solutions and store them in its archive. MACS then employs its heuristics to generate new candidate solutions, which are passed back to the NLP to restore feasibilty. Its important to highlight that MACS takes a fully feasible solution and only changes its static and control variables. Thus the NLP receives a good warm start solution and typically takes a fraction of a second to return a new fully feasible solution. This way, global exploration is performed and a Pareto front is produced.

In addition, once every user specified number of iterations, the NLP solver is invoked in a different mode, with the task of refining the solutions and guaranteeing local optimality. Since the NLP solver is inherently single objective, the way the various objectives are combined to produce a single objective problem is of paramount importance. In this phase MODHOC employs the Pascoletti-Serafini scalarisation, which can be seen as a constrained but continuous and differentiable form of the Tchebychev scalarisation. Thus, MODHOC has the unique capability of performing both global and local search, employing the more suitable scalarisation scheme depending on weather it is performing global or local search, and seamlessly transition between the two equivalent scalarisation schemes. More details about the implementation can be found in [5].

This way, MODHOC has been able to automatically produce Pareto optimal solutions of problems with known single objective solution, like the reentry of a shuttle like vehicle, the maximum energy orbit rise of a spacecraft and the minimum time transfer to rectilinear path [4, 5]. In addition, it was able to solve complex coupled vehicle and trajectory design problems without requiring any user supplied guess [5].

2.4. Treatment of Discrete variables

Since MACS was initially conceived to solve problems with continuous variables only, its heuristics have been extended in order to deal also with integer variables. The heuristics have been modified in such a way that after their application the value of the discrete variables remains integer and within the allowed bound, while leaving unchanged the behaviour of the heuristics when operating on the continuous variables. This enables MACS to directly treat nonlinear mixed integer multi objective optimisation problems.

However, MODHOC relies on the NLP solver in order to ensure tight satisfaction of the constraints. For this reason, when the solutions are passed to the NLP problems, the discrete variables are relaxed and treated as continuous. This allows the NLP solver to change the value of the relaxed variables, if this is needed to get a feasible solution. After a solution of the relaxed problem is found an additional constraint is imposed on each relaxed variable x_i in order to force them to assume only integer values within the prescribed bounds:

$$\sin(\pi x_i) = 0 \qquad L_i \leq x_i \leq U_i \tag{1}$$

This constraint is smooth and is satisfied only for integer values of the relaxed variable x_i. The NLP solver is run again with the imposition of this constraint, and the fully feasible non-relaxed solution is returned to MACS. This way, no heuristics are needed to round the relaxed variables.

3. APPLICATIONS

This section shows three example applications: a three objective design and trajectory optimisation problem of a launch vehicle, a two objective trajectory optimisation from LEO to GEO, and a two objective planning and scheduling problem of a simplified car model, where three target destinations need to be visited but the order in which they are visited is not specified a priori.

3.1. 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 $200\,\mathrm{m\,s^{-1}}$ at an altitude of $10\,\mathrm{km}$ eastbound along the equator, with an initial flight path angle of $10°$. It has to deliver a $500\,\mathrm{kg}$ payload to a $650\,\mathrm{km}$ 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. 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.

3.1.1. Structural mass models

For each stage, the dry mass was computed as a function of the engine mass and propellent mass. The vacuum thrust of the engines was used to estimate their structural mass based on an empirical linear relationship of existing commercial engines. The mass model was developed in parallel for an industrial vehicle and cannot be released publicly [24]. For the first stage engine, $0 \leq T_{vac} \leq 2\,\text{MN}$ and $I_{sp} = 332\,\text{s}$, while for the second stage engine $0 \leq T_{vac} \leq 200\,\text{kN}$ and $I_{sp} = 352\,\text{s}$. Propellent masses were limited to $100\,\text{t}$ for the first stage and $20\,\text{t}$ for the second. The maximum gross takeoff mass for the first stage was also assumed to be $100\,\text{t}$.

3.1.2. 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.

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

3.1.3. 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:

$$m_{0,2} = m_{f,1} - m_{dry,1} \tag{3}$$

3.1.4. Results

Figure 1 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.

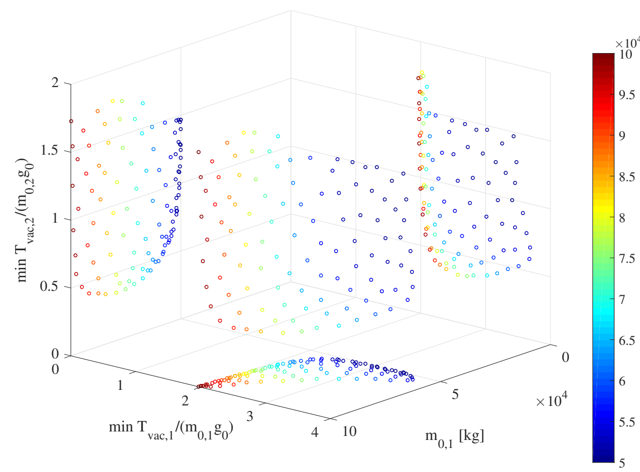


Fig. 1: Three-Objective Ascent: set of Pareto-optimal solutions, colorbar indicates gross mass of the vehicle. The 3D Pareto front is in the middle, with orthographic projections shown on each coordinate plane.

Figures 2 and 3 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 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. 3b. The total flight duration is also slightly longer than the other two.

Table 1 reports the vehicle design parameters for the Pareto extrema (i.e., the solutions that minimise each ob-

Table 1: 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 [$\mathrm{km\,s^{-1}}$]
$\min(m_{0,1})$	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

jective 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, 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.

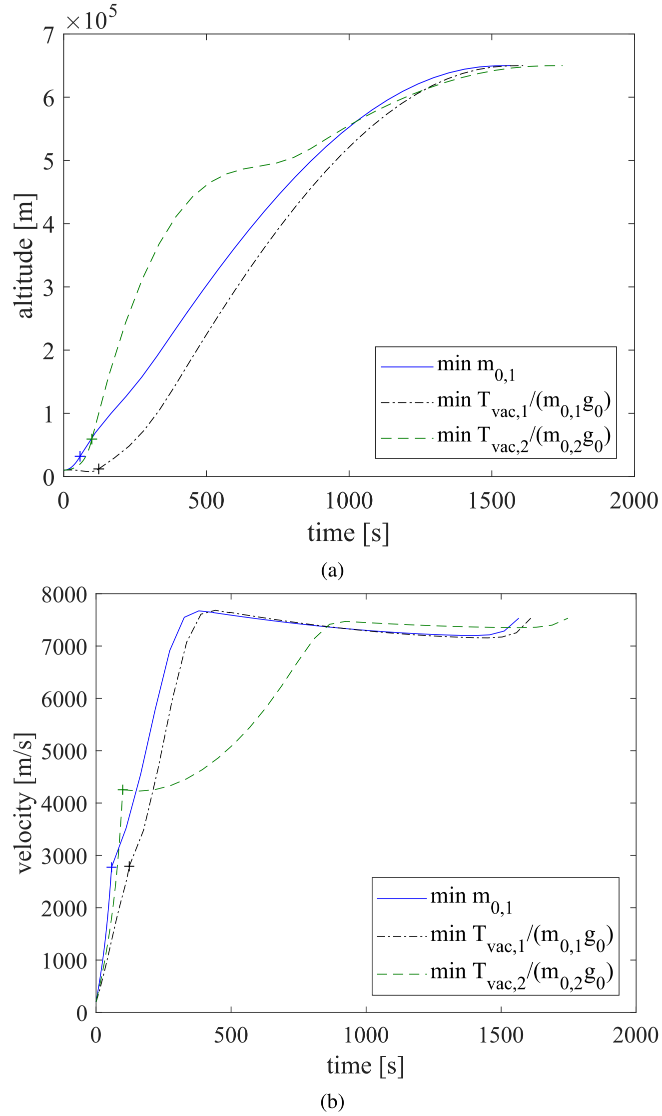
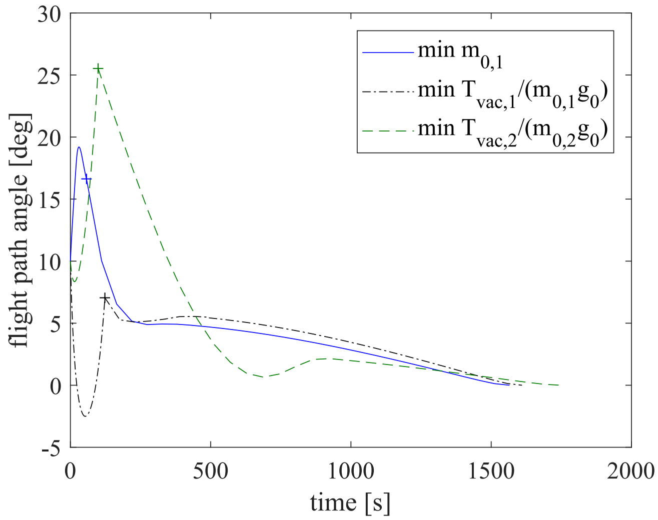
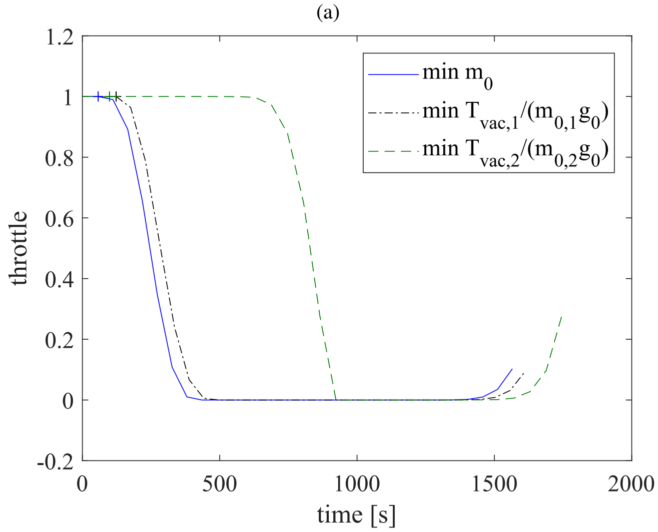


Fig. 2: Three-Objective Ascent: time-history of a) the altitude and b) the velocity for the three extreme solutions of the Pareto front in Fig. 1. The + indicates the stage separation point.



(a)



(b)

Fig. 3: 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. 1. The $+$ indicates the stage separation point.

3.2. Two objective LEO to GEO transfer

This test case is a multi-objective extension of a problem presented in [25], the trajectory optimisation for a transfer from LEO orbit to GEO orbit. To remain consistent with the original problem, in this case the Imperial System was used. The vehicle is controlled by two finite length impulses of a chemical engine. The engine has an I_{sp} of 300 seconds and can only operate at full thrust of 1.25 lbs, but the duration of the thrusting arcs and the direction of thrusting are free. The spacecraft has a notionary mass of 1 lbs, and is orbiting earth at an altitude of 150 Nautical miles along a circular orbit with 28 degrees of inclination. In order to avoid any singularity, the dynamics is formulated using the modified equinoctial elements as in [25]. The objectives of the optimisation are to minimise the transfer time, and to maximise the final mass.

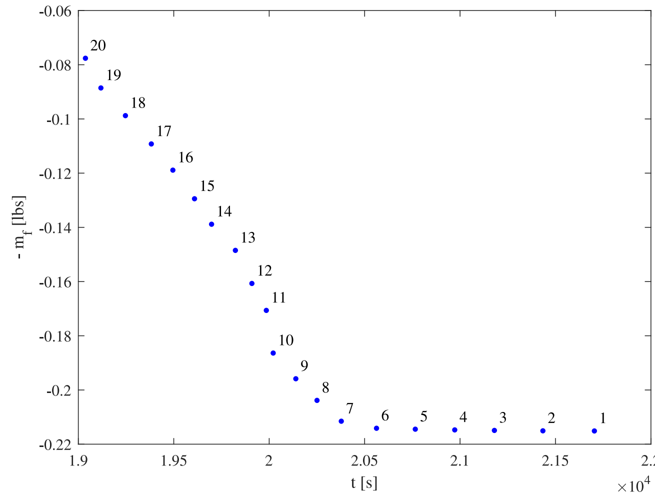


Fig. 4: Pareto front for the LEO to GEO transfer

3.2.1. Numerical settings

The problem was formulated as a 4 phases problem: phase 1 and 3 model the coasting arc, and are discretised with 3 DFET elements of order 6. Phases 2 and 4 are thrusting arcs, but due to their expected short duration they are discretised with a single DFET element of order 6. Continuity of the state variables was imposed between the phases, and the final time of each phase was left as a free parameter to be determined by the optimiser. MODHOC was run for a total of 50000 function evaluations, with gradient based refinement taking place every 10 iterations. 20 points on the Pareto front were sought.

3.2.2. Results

Figure 4 shows the Pareto front: as it is evident, lower mission times imply lower final mass. Solution marked with 1 coincides with the solution found in the reference, corresponding to the maximum final mass and a transfer time of 21704 seconds. The transfer time of the reference solution took 21683 seconds. This slight difference can be attributed to the difference of the integration schemes, the NLP solvers and settings used, and the fact that the solution in the reference comes after 4 mesh refinement iterations. However, the difference in final time is lower than 0.1%. The Pareto front also suggests an interesting trade-off region in solutions marked between 1 and 7, where the transfer time can be reduced by approximately 10% with a reduction of the final mass of less than 1%.

Figure 5 represents all the 20 trajectories. Light blue and green trajectories correspond to solutions 1 to 6, characterised by higher mission times and higher final mass. Blue and purple solutions correspond to solution 7-19, while solution 20

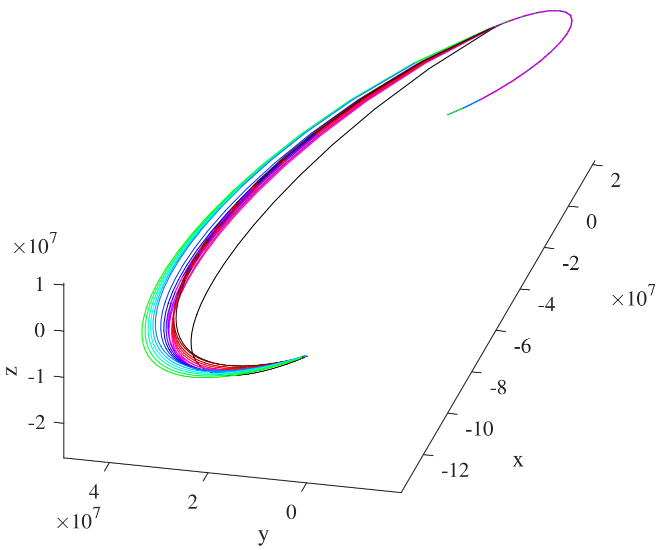


Fig. 5: Trajectories for the LEO to GEO transfer

is the black one, which is the minimum mission time solution and is markedly different from the others

3.3. A planning and scheduling problem

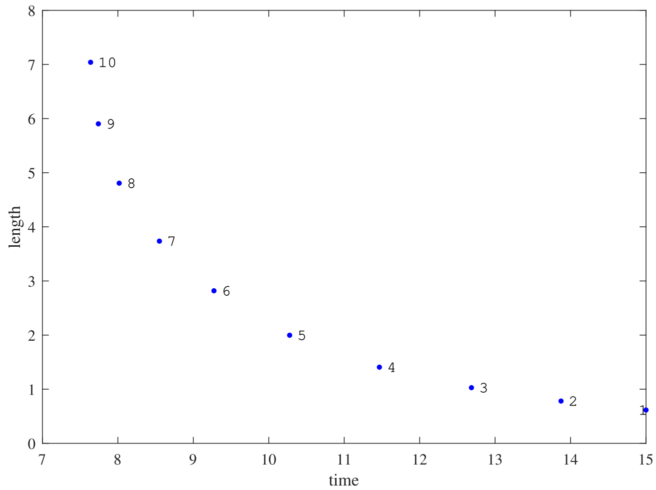
This section describes a non space related application. It is a multi-objective extension of a problem presented in [26]: a vehicle, described by a simple two dimensional dynamic model, starts from the origin of the plane and has to visit three target destinations before finally returning to its starting position. It is controlled by the magnitude of the acceleration and by the steering rate, both of which are limited. The order in which the three targets are to be visited is not specified a priori but has to be found as part of the solution, which in the original reference had to minimise the total mission time. The vehicle has to pass on each target point without any restriction on the velocity or direction of velocity at the rendez-vous, while at the final time it had to be at rest at the original position. This problem can be seen as an extension of a classic Travelling Salesman Problem, where the presence of a dynamical model for the Salesmen significantly changes the nature of the problem and its complexity. The problem is here extended with a second objective minimising the total energy expense, measured as the integral of the square of the instantaneous acceleration.

3.3.1. Numerical settings

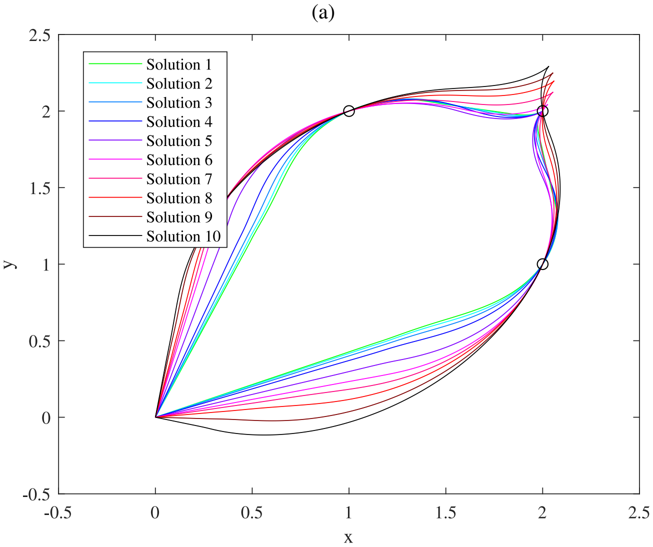
The problem was divided into four phases. At the end of the first three phases, conditions were imposed to enforce the position of the car to match with the position of one of the three targets, while for the last phase the destination was the origin, with zero final velocity. A special set of constraints was imposed to deal with the categorical choice of the destinations, including the constraint that each target must be visited only once. On each phase, the problem was discretised using 3 DFET elements of order 7. MODHOC was run for a total of 40000 objective function evaluations, with 10 agents to find 10 points on the Pareto front.

3.3.2. Results

Figure 6a shows the Pareto front. Solution 10 corresponds to the minimum time solution computed in [26]. The solution computed with MODHOC has a minimum time of $7.6387\,\mathrm{s}$, while the solution computed in the reference has a minimum time of $7.6166\,\mathrm{s}$. The difference is below 0.3% and can be attributed to the different integration schemes and resolutions. The shape of the trajectory, of the control law and of the target visiting order is the same: the first solution to be visited is the bottom right one and then proceeds counter-clockwise. The authors of the reference noted the presence of several local minima even for the same set of integer variables, but did not explain with which approach they found their best solution. With MODHOC, this comes naturally as part of the global exploration and the simultaneous treatment of both discrete and continuous variables. The minimum energy solution takes, as expected, the maximum allowed time of 15 seconds. All solutions visit the targets in the same order. Figure 6b shows the trajectories of the 10 computed solutions and the 3 targets to be visited, marked as black circles: as it is evident, minimum energy solutions are very close to straight trajectories between the targets, while minimum time solutions are longer and with more pronounced curves. Figure 7a shows the magnitude of the velocities over time and the instant at which every solution encounters a target is marked with a $+$. It allows to understand an interesting aspect of the problem: since the velocity of the vehicle is allowed to become negative, all solutions proceed forward until they visit the second target. The velocity at the second target is zero only for solutions 1 to 5, while for the other solutions the vehicle still has positive velocity but is decelerating. After that turning point the vehicle proceeds backwards, as it is possible to see from the negative value of the velocities. This interesting and unexpected feature is telling that the steering rate of the vehicle is insufficient to visit all the three targets by going forwards only without penalising too much the mission time. It is instead more efficient to proceed forwards for a period of time, and then backwards for the second part of the mission. Figure 7b shows the control profile for the acceleration and again the instants of time where the targets are encountered is marked with a $+$. As expected, the minimum time solutions corresponds to a double bang-bang solution, i.e. a maximum acceleration followed by a minimum deceleration and then again a maximum acceleration until the vehicle stops. The minimum energy solution instead is composed of two linear profiles of opposite slope, while the solutions in between have steeper linear profiles or different switching times for their bang bang solutions.

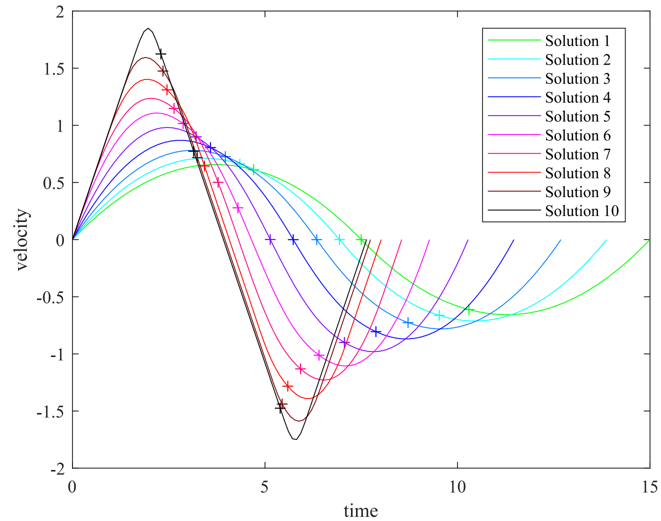


(a)

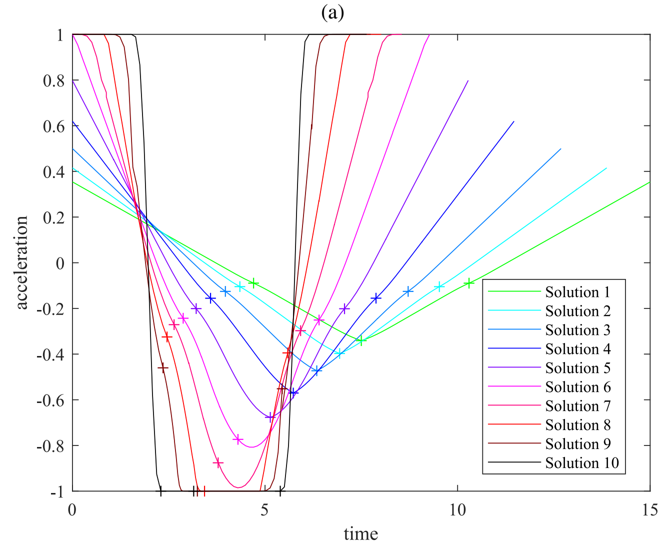


(b)

Fig. 6: Dynamic planning and scheduling problem: a) Pareto front, b) Trajectories. Circled indicate target destinations.



(a)



(b)

Fig. 7: Dynamic planning and scheduling problem: time histories for a) velocities and b) accelerations. + indicate when a target destination is visited.

4. CONCLUSIONS

This paper presented MODHOC, a direct method for solving multiobjective hybrid optimal control problems. By employing DFET, a general and powerful transcription method, it is able to solve nonlinear optimal control problems with user defined dynamical models and path constraints. MACS, the memetic multiobjective optimisation algorithm, allows to globally explore the search space, automatically generate initial guesses and return an evenly spread set of solutions. The smart coupling with the NLP solver through the bi-level and the single level approaches allows for the strict satisfaction of general nonlinear constraints, and to guarantee local optimality of the solutions. The simultaneous treatment of discrete and continuous variables allows to treat complex mixed integer problems, where the selection of discrete variables does not imply the existence of a unique optimal solution depending only on the continuous variables. Moreover, it is possible to treat problems with nonlinear constraints involving both the continuous and the integer variables. The paper showcased three different applications, each one characterised by different kinds of complexity: a three objective coupled system and trajectory design for a two stage launch vehicle, a two objective trajectory design for a LEO to GEO transfer, and a two objective mixed integer planning and scheduling problem. MODHOC returned evenly spread sets of solutions, allowing to understand interesting relations between the objectives and the underlying physics of the problem. In several cases, the results obtained also included unexpected but useful features, giving even more insight to the decision makers.

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.

5. REFERENCES

[1] Marilena Di Carlo, Lorenzo Angelo Ricciardi, and Massimiliano Vasile, "Multi-objective optimisation of constellation deployment using low-thrust propulsion," in *AIAA/AAS Astrodynamics Specialist Conference*. sep 2016, American Institute of Aeronautics and Astronautics.

[2] Lorenzo A Ricciardi, Massimiliano Vasile, and Christie Maddock, "Global solution of multi-objective optimal control problems with multi agent collaborative search and direct finite elements transcription," in *Evolutionary Computation (CEC), 2016 IEEE Congress on*. IEEE, 2016, pp. 869–876.

[3] Lorenzo Angelo Ricciardi, Massimiliano Vasile, Federico Toso, and Christie A Maddock, "Multi-objective optimal control of the ascent trajectories of launch vehicles," in *AIAA/AAS Astrodynamics Specialist Conference*, 2016, p. 5669.

[4] Massimiliano Vasile and Lorenzo Ricciardi, "A direct memetic approach to the solution of multi-objective optimal control problems," in *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*. IEEE, 2016, pp. 1–8.

[5] Lorenzo A. Ricciardi, Christie A. Maddock, and Massimiliano Vasile, "Direct solution of multi-objective optimal control problems applied to spaceplane mission design," *Journal of Guidance, Control and Dynamics*, p. Accepted for publication on 26 August 2018, 2018.

[6] Massimiliano Vasile and Amalia E Finzi, "Direct lunar descent optimisation by finite elements in time approach," *International Journal of Mechanics and Control*, vol. 1, no. 1, 2000.

[7] D. H. Hodges and R. R. Bless, "Weak hamiltonian finite element method for optimal control problems," *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 1, pp. 148–156, jan 1991.

[8] Marco Borri, Carlo Bottasso, and Paolo Mantegazza, "Basic features of the time finite element approach for dynamics," *Meccanica*, vol. 27, no. 2, pp. 119–130, Jun 1992.

[9] M. Borri and C. Bottasso, "A general framework for interpreting time finite element formulations," *Computational Mechanics*, vol. 13, no. 3, pp. 133–142, Dec 1993.

[10] Carlo L Bottasso and Andrea Ragazzi, "Finite element and runge-kutta methods for boundary-value and optimal control problems," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 4, pp. 749–751, 2000.

[11] Massimiliano Vasile and Franco Bernelli-Zazzera, "Optimizing low-thrust and gravity assist maneuvers to design interplanetary trajectories," *arXiv preprint arXiv:1105.1829*, 2011.

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

[13] Lorenzo Ricciardi, Christie Maddock, and Massimiliano L. Vasile, "Multi-objective optimal control of reentry and abort scenarios," in *2018 Space Flight Mechanics Meeting*. jan 2018, American Institute of Aeronautics and Astronautics.

[14] Lorenzo A. Ricciardi and Massimiliano Vasile, "Direct transcription of optimal control problems with finite elements on bernstein basis," *Journal of Guidance, Control and Dynamics*, p. Accepted for publication on 26 August 2018, 2018.

[15] Federico Zuiani and Massimiliano Vasile, "Preliminary design of debris removal missions by means of simplified models for low-thrust, many-revolution transfers," *International Journal of Aerospace Engineering*, vol. 2012, 2012.

[16] Federico Zuiani, Yasuhiro Kawakatsu, and Massimiliano Vasile, "Multi-objective optimisation of many-revolution, low-thrust orbit raising for destiny mission," in *23rd AAS/AIAA Space Flight Mechanics Conference*, 2013.

[17] M. Vasile and F. Zuiani, "Multi-agent collaborative search: an agent-based memetic multi-objective optimization algorithm applied to space trajectory design," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 225, no. 11, pp. 1211–1227, 2011.

[18] Federico Zuiani and Massimiliano Vasile, "Improved individualistic actions for multi-agent collaborative search," *In EVOLVE*, July 2013.

[19] Federico Zuiani and Massimiliano Vasile, "Multi agent collaborative search based on tchebycheff decomposition," *Computational Optimization and Applications*, vol. 56, no. 1, pp. 189–208, 2013.

[20] Federico Zuiani and Massimiliano Vasile, “Multi agent collaborative search with thecycheff decomposition and monotonic basin hopping,” *In BIOMA*, May 2012.

[21] Lorenzo A. Ricciardi and Massimiliano Vasile, “Improved archiving and search strategies for multi agent collaborative search,” in *Computational Methods in Applied Sciences*, pp. 435–455. Springer International Publishing, jul 2018.

[22] Nicolas Thiry and Massimiliano Vasile, “Statistical multi-criteria evaluation of non-nuclear asteroid deflection methods,” *Acta Astronautica*, vol. 140, pp. 293 – 307, 2017.

[23] Carlos Ortega Absil, Lorenzo Ricciardi, Marilena Di Carlo, Cristian Greco, Romain Serra, Mateusz Polnik, Aram Vroom, Annalisa Riccardi, Edmondo Minisci, and Massimiliano Vasile, “Gtoc 9: Results from university of strathclyde (team strath++),” *Acta Futura*, 2018.

[24] Christie Maddock, Federico Toso, Lorenzo Ricciardi, Alessandro Mogavero, Kin Hing Lo, Sriram Rengarajan, Konstantinos Kontis, Andy Milne, David Evans, Michael West, and Stuart McIntyre, “Vehicle and mission design of a future small payload launcher,” in *21st AIAA International Space Planes and Hypersonics Technologies Conference*. mar 2017, American Institute of Aeronautics and Astronautics.

[25] John T Betts, *Practical methods for optimal control and estimation using nonlinear programming*, SIAM, 2010.

[26] O Von Stryk and M Glocker, “Numerical mixed-integer optimal control and motorized traveling salesmen problems. commande numerique optimale a entiers mixtes et problemes de voyageur de commerce motorise,” *Journal Européen des Systèmes Automatisés*, vol. 35, no. 4, pp. 519–534, 2001.