Optimal Control by Multiple Shooting and Weighted Tchebycheff Penalty-based Scalarization[⋆]

Gisela C. V. Ramadas (✉)[1] [0000−0003−3883−2964], Edite M. G. P. Fernandes[2] [0000−0003−0722−9361], Ana Maria A. C. Rocha[2] [0000−0001−8679−2886], and M. Fernanda P. Costa[3] [0000−0001−6235−286X]

[1] Research Center of Mechanical Engineering (CIDEM)
School of Engineering of Porto (ISEP), Polytechnic of Porto, 4200-072 Porto, Portugal
gcv@isep.ipp.pt
[2] ALGORITMI Center
University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
{emgpf,arocha}@dps.uminho.pt
[3] Centre of Mathematics,
University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
mfc@math.uminho.pt

Abstract. Numerical direct multiple shooting (MS) methods have shown to be important and efficient tools to solve optimal control problems (OCP). The use of an MS method to solve the OCP gives rise to a finite-dimensional optimization problem with a set of "continuity constraints" that should be satisfied together with the other algebraic states and control constraints of the OCP. Using non-negative functions to measure the violation of the "continuity constraints" and of the algebraic constraints separately, the finite-dimensional problem is reformulated as a multi-objective problem with three objectives to be optimized. This paper explores the use of a multi-objective approach, the weighted Tchebycheff scalarization method, to minimize the objective functional and satisfy all the constraint conditions of the OCP. During implementation, a penalty term is added to the Tchebycheff aggregated objective function aiming to force and accelerate the convergence of the constraint violations to zero. The effectiveness of the new methodology is illustrated with the experiments carried out with six OCP.

Keywords: Optimal Control, Multiple Shooting, Tchebycheff Scalarization.

[⋆] This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020, UIDB/00013/2020 and UIDP/00013/2020 of CMAT-UM. We also acknowledge the financial support of CIDEM.

1 Introduction

In this paper, we consider solving an optimal control problem (OCP) by a direct multiple shooting (MS) method, and explore the use of a weighted Tchebycheff scalarization method to take care of the simultaneous minimization of three objective functions. An OCP is a constrained optimization problem that has a set of dynamic equations as constraints. There are three types of OCP that differ in the formulation of the functional to be optimized: OCP in the *Bolza form*, in the *Lagrange form* and in the *Mayer form*. They are equivalent and it is possible to convert a problem in one of the forms into another one [1]. Here, we assume that the OCP is in the *Mayer form*:

$$\begin{array}{ll} \min\limits_{\boldsymbol{u}(t)\in U} & J(\boldsymbol{y}(t),\boldsymbol{u}(t)) \equiv M(T,\boldsymbol{y}(T)) \\ \quad\text{s.t.} & \boldsymbol{y}'(t) = \boldsymbol{f}(t,\boldsymbol{y}(t),\boldsymbol{u}(t))\,, t\in[0,T],\ \boldsymbol{y}(0)=\boldsymbol{y}_0,\ \boldsymbol{y}(T)=\boldsymbol{y}_T, \\ & \quad 0 = h_e(t,\boldsymbol{y}(t),\boldsymbol{u}(t)), e\in E\,,\ t\in[0,T], \\ & \quad 0 \geq g_j(t,\boldsymbol{y}(t),\boldsymbol{u}(t)), j\in F\,,\ t\in[0,T], \end{array} \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^s$ is the vector of state variables, $\boldsymbol{u} \in U \subset \mathbb{R}^c$ is the vector of control, U represents a class of functions (in particular functions of class C^1 and piecewise constant), $E = \{1, 2, \ldots, l_h\}$ and $F = \{1, 2, \ldots, l_g\}$ [1]. In the problem of Mayer, the functional is not an integral but a function M that depends in general on the dependent variables $\boldsymbol{y}$ and the final point of the t-domain T. For simplicity, we assume that the initial point of the t-domain is 0.

In the OCP we want to find $\boldsymbol{u}$ that minimizes the objective functional J subject to the dynamic system of ordinary differential equations (ODE) and the mixed states and control (equality and inequality) constraints.

Methods for solving OCP like (1) can be classified into two classes. In an indirect method, the first-order necessary conditions from Pontryagin's maximum principle are used to reformulate the original problem into a boundary value problem [2]. On the other hand, direct methods solve the OCP directly. They transform the infinite-dimensional OCP into a finite-dimensional optimization problem that can be solved by efficient nonlinear programming (NLP) algorithms. All direct methods discretize the control variables but differ in the way they treat the state variables.

In a direct MS method the t-domain is partitioned into smaller subintervals and the system of ODE is integrated in each subinterval independently. Besides the control variables, the new *state start values* (for the state variables) at each subinterval make the decision variables of the finite NLP problem [2,3]. When a direct MS method is used to solve the OCP, a set of "continuity constraints" must be defined and should be satisfied together with the other algebraic mixed states and control constraints. To solve the resultant finite NLP problem, well-known NLP methods can be used, namely a sequential quadratic programming procedure or an interior-point method [3,4]. To reduce the need for numerical (or analytical) derivatives, a first-order descent method based on the filter methodology has been proposed in [5].

To minimize the objective function and satisfy all the constraints - the "continuity constraints" and the algebraic constraints - the herein proposed methodology reformulates the NLP problem as a multi-objective optimization (MOO) problem with three objectives to be simultaneously optimized. Thereafter, the weighted Tchebycheff scalarization method is used to solve the reformulated finite MOO problem. We also take advantage of the weighted Tchebycheff method by solving problems with non-convex Pareto fronts and force the simultaneous minimization of the objectives by adding a penalty term to the weighted Tchebycheff scalar function.

The paper is organized as follows. Section 2 introduces the direct MS method for solving the OCP in the *Mayer form*, Sect. 3 shows the mathematical formulation of the finite NLP problem and the objective functions that are required to be simultaneously minimized. Section 4 briefly presents the basic multi-objective concepts, the objective function to be minimized in the weighted Tchebycheff scalarization context and the details concerning the new algorithm. Section 5 illustrates the implementation of the methodology with six OCP and we conclude the paper in Sect. 6.

2 Direct Multiple Shooting Method

In a direct MS method, the controls are discretized in the NLP. On a specific grid defined by $0 = t_1 < t_2 < \cdots < t_{N-1} < t_N = T$, where $N-1$ is the total number of subintervals, the control $\boldsymbol{u}(t)$ is discretized, namely using a piecewise constant: $\boldsymbol{u}(t) = \boldsymbol{q}^i, \quad \text{for} \quad t \in [t_i, t_{i+1}] \quad \text{and} \quad i = 1, \dots, N-1$, so that $\boldsymbol{u}(t)$ only depends on the control parameters $\boldsymbol{q} = (\boldsymbol{q}^1, \boldsymbol{q}^2, \dots, \boldsymbol{q}^{N-1})$. Besides the discretized controls, the *state start values* at the nodes of the grid - herein represented by $\boldsymbol{x}^i \in \mathbb{R}^s$, $i = 1, 2, \dots, N-1$ - are also decision variables of the NLP problem [2]. The variables $\boldsymbol{x}^i$, $i = 1, 2, \dots, N-1$ are the initial values for the state variables for the $N-1$ independent initial value problems on the subintervals $[t_i, t_{i+1}]$:

$$\boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y}(t), \boldsymbol{q}^i), \ \text{ for } \ t \in [t_i, t_{i+1}] \text{ and } \ \boldsymbol{y}(t_i) = \boldsymbol{x}^i.$$

The continuity of the solution trajectories $\boldsymbol{y}^i(t; \boldsymbol{x}^i, \boldsymbol{q}^i)$ is guaranteed by satisfying the "continuity conditions":

$$\boldsymbol{y}^i(t_{i+1}; \boldsymbol{x}^i, \boldsymbol{q}^i) = \boldsymbol{x}^{i+1}, \ i = 1, \dots, N-1, \tag{2}$$

as well as the initial state, $\boldsymbol{x}^1 = \boldsymbol{y}_0$, and the final state, $\boldsymbol{x}^N = \boldsymbol{y}_T$, constraints. The dynamic system is then solved by an ODE solver on each shooting subinterval $[t_i, t_{i+1}]$ independently, and the state variables $\boldsymbol{y}(t)$ are considered as dependent variables $\boldsymbol{y}(t, \boldsymbol{q})$.

3 The NLP Problem

We assume that the NLP problem is a non-convex constrained optimization problem (COP). We also assume that the OCP is in the *Mayer form*, the ODE system has initial and boundary state values, and algebraic equality and inequality constraints, involving state and control variables, are present. The dynamic system is solved, in each subinterval $[t_i, t_{i+1}]$, by the explicit 4th order Runge-Kutta integration formula based on 5 points. The mathematical form of the COP is the following:

$$\begin{array}{ll}\min\limits_{\boldsymbol{x}^i,\, i\in I_N;\boldsymbol{q}^i,\, i\in I} & M(T,\boldsymbol{y}(T)) \\ \text{s.t.} \quad g_j(\boldsymbol{y}^i(t;\boldsymbol{x}^i,\boldsymbol{q}^i),\boldsymbol{q}^i) \leq 0, \, t\in[t_i,t_{i+1}], i\in I, j\in F \\ \qquad h_e(\boldsymbol{y}^i(t;\boldsymbol{x}^i,\boldsymbol{q}^i),\boldsymbol{q}^i) = 0, \, t\in[t_i,t_{i+1}], i\in I, e\in E \\ \quad \boldsymbol{y}^i(t_{i+1};\boldsymbol{x}^i,\boldsymbol{q}^i) - \boldsymbol{x}^{i+1} = 0, i\in I, \; \boldsymbol{x}^1 - \boldsymbol{y}_0 = 0, \boldsymbol{x}^N - \boldsymbol{y}_T = 0 \; , \end{array} \tag{3}$$

where $I = \{1, \ldots, N-1\}$ and $I_N = I \cup \{N\}$. In order to solve the optimization problem (3), the objective function, the "continuity constraints" $\boldsymbol{y}^i(t_{i+1};\boldsymbol{x}^i,\boldsymbol{q}^i) - \boldsymbol{x}^{i+1} = 0, i \in I$, the initial state and the final state constraints, and the algebraic equality and inequality constraints must be evaluated by solving the ODE system. An optimal solution to the problem (3) satisfies all the constraints and achieves the least objective function value.

To measure the violation of the "continuity constraints", initial state and final state constraints, the following non-negative function is used

$$\theta(\boldsymbol{x},\boldsymbol{q}) = \sum_{l\in L}\sum_{i\in I}(y_l^i(t_{i+1};\boldsymbol{x}^i,\boldsymbol{q}^i) - x_l^{i+1})^2 + \sum_{l\in L}(x_l^1 - y_{l_0})^2 + \sum_{l\in L}(x_l^N - y_{l_T})^2 \; , \tag{4}$$

where $L = \{1, 2, \ldots, s\}$. If the solution $(\boldsymbol{x},\boldsymbol{q})$ satisfies these constraints, $\theta(\boldsymbol{x},\boldsymbol{q})$ is zero; otherwise is positive. Similarly, the non-negative function, p, used to measure the algebraic equality and inequality constraints violation, is defined as follows:

$$p(\boldsymbol{x},\boldsymbol{q}) = \sum_{j\in F}\sum_{i\in I}\max\left\{0, g_j(\boldsymbol{y}^i(t;\boldsymbol{x}^i,\boldsymbol{q}^i),\boldsymbol{q}^i)\right\}^2 + \sum_{e\in E}\sum_{i\in I} h_e(\boldsymbol{y}^i(t;\boldsymbol{x}^i,\boldsymbol{q}^i),\boldsymbol{q}^i)^2, \tag{5}$$

where $p(\boldsymbol{x},\boldsymbol{q}) = 0$ when the corresponding constraints are satisfied, otherwise $p(\boldsymbol{x},\boldsymbol{q}) > 0$.

In this paper, the constraint violation functions $\theta(\boldsymbol{x},\boldsymbol{q})$ and $p(\boldsymbol{x},\boldsymbol{q})$, and the optimality measure $M(T,\boldsymbol{x},\boldsymbol{q})$, are used to reformulate the COP (3) into a tri-objective optimization (TOO) problem. In this TOO problem, both the feasibility measures – defined by the above defined constraint violation functions – and the optimality measure (defined by the objective function $M(T,\boldsymbol{x},\boldsymbol{q})$) are minimized simultaneously.

To simplify the notation, the letter x will be used to denote the vector of the decision variables $x = (\boldsymbol{x},\boldsymbol{q})$ (with $n = sN + c(N-1)$ components) and $f_1(x) = \theta(x)$, $f_2(x) = p(x)$ and $f_3(x) = M(T, x(T))$. Thus, the TOO problem is the following:

$$\min_{x\in\Omega\subseteq\mathbb{R}^n} (f_1(x), f_2(x), f_3(x)) \, , \tag{6}$$

where $x \in \mathbb{R}^n$ is the vector of the decision variables, n is the number of decision variables, Ω is the feasible search region (often called *feasible decision space*) and the components of the vector $\boldsymbol{f} : \mathbb{R}^n \rightarrow \mathbb{R}^3$ are the objective functions (also called criteria, payoff functions, or cost functions) to be optimized. The feasible criterion space $\mathcal{F}$ is defined as the set $\mathcal{F} = \{\boldsymbol{f}(x) \text{ such that } x \in \Omega\}$. This set is also called the *attainable set*. However, we note that there are points in the feasible objective space that do not correspond to a single point $x \in \Omega$. The space $\mathbb{R}^n$ is called the decision space and $\mathbb{R}^3$ is called the objective space.

When the objective functions are not conflicting, it is possible to find a solution where every objective function attains its minimum [6]. However, if the objectives are conflicting, i.e., the improvement of one objective leads to another objective deterioration, one single optimal solution does not exist, but a set of alternatives - the non-dominated solutions - further ahead called Pareto optimal set. The decision-maker then selects one (or more than one) compromise solution, among the alternatives, that better satisfies his/her preferences.

4 Multi-Objective Optimization

The process of optimizing systematically and simultaneously a collection of objective functions is called MOO [7]. The simultaneous optimization of several objectives has been attracting the attention of scientific researchers, since it is possible to find a set of solutions that represent different compromises between the objectives. The decision-maker is then able to choose the solution that better suits his/her goals.

4.1 Basic Concepts

First, and assuming that the number of objectives, m, is greater than 1, the definition of dominance is presented.

Definition 1. *A vector $\boldsymbol{f} = (f_1, \ldots, f_m)$ is said to dominate $\bar{\boldsymbol{f}} = (\bar{f}_1, \ldots, \bar{f}_m)$ if and only if*

$$\forall i \in \{1, \ldots, m\}\ f_i \leq \bar{f}_i \ \textit{ and } \ \exists i \in \{1, \ldots, m\} \ \textit{ such that } f_i < \bar{f}_i. \tag{7}$$

When two solutions $\boldsymbol{f}^1 = \boldsymbol{f}(x^1)$ and $\boldsymbol{f}^2 = \boldsymbol{f}(x^2)$, $x^1, x^2 \in \Omega \subseteq \mathbb{R}^n$ are compared, one of these three cases is true: i) $\boldsymbol{f}^1$ dominates $\boldsymbol{f}^2$, ii) $\boldsymbol{f}^1$ is dominated by $\boldsymbol{f}^2$, iii) $\boldsymbol{f}^1$ and $\boldsymbol{f}^2$ are non-dominated. The next definition states the condition for a feasible solution to be a Pareto optimal solution.

Definition 2. *Let $\boldsymbol{f} \in \mathbb{R}^m$ be the objective functions vector. A solution $x^1 \in \Omega$ is said to be Pareto optimal if and only if there is no other solution $x^2 \in \Omega$ for which $\boldsymbol{f}(x^2)$ dominates $\boldsymbol{f}(x^1)$.*

This means that x^1 is a Pareto optimal solution if there is no other feasible solution, x^2, which would decrease some objective f_i without causing a simultaneous increase in at least one other objective. In MOO, there is no single optimal solution, but a set of optimal solutions called Pareto optimal set (in the space of the decision variables). The corresponding function vectors are said to be non-dominated (ND) (see [8] for details concerning MOO).

Definition 3. *Given a MOO problem with objective function vector $\boldsymbol{f} \in \mathbb{R}^m$ and the Pareto optimal set X^*, the Pareto optimal front (PF^*) is defined as:*

$$PF^* = \{\boldsymbol{f} = (f_1(x), \ldots, f_m(x)) \;\; \textit{such that} \;\; x \in X^*\}.$$

4.2 Scalarization Approaches to MOO

The goal of a MOO algorithm is to find a good approximation to the Pareto front PF^* (and to the Pareto optimal set), i.e., to find a reasonable number of Pareto function vectors which are evenly distributed along the Pareto optimal front. The most popular methods to solve the MOO problem produce an approximation to the PF^* directly [9]. They are stochastic methods and although they are naturally prepared to produce many solutions, since they are in general population-based techniques, the computational effort to achieve the solutions is substantial.

Alternatively, a single solution can be found by aggregating the objective functions into a scalar objective function that is used in a single-objective optimization (SOO) context. When combining the objectives, a vector of weights should be provided by the decision-maker prior to the optimization. In order to be able to obtain an approximation to the PF^*, the SOO method must be run as many times as the desired number of points using different vectors of weights [10]. The most used aggregation method is the weighted sum approach that assigns to each objective function f_i, of the vector $\boldsymbol{f}$, a non-negative weight w_i, minimizing the function that is the weighted sum of the objectives. Although this function is differentiable and simple to implement, it suffers from a drawback since certain Pareto optimal solutions in non-convex regions of the Pareto optimal front cannot be found.

Alternatively, the weighted Tchebycheff approach also assigns a vector of weights to the objectives and relies on a nonlinear weighted aggregation of the functions f_i to form a single objective [11,12]. Thus, it is able to deal with a non-convex Pareto front [13]. In the minimization context, the resulting SOO problem has the form

$$\min_{x \in \Omega} \Psi(x; \boldsymbol{w}) \equiv \max \left\{ w_1 \left| f_1(x) - z_1^U \right|, \ldots, w_m \left| f_m - z_m^U \right| \right\} \tag{8}$$

where $\boldsymbol{w} = (w_1, \ldots, w_m)$ is the vector of weights satisfying $w_i \geq 0, i = 1, \ldots, m$ and $w_1 + \cdots + w_m = 1$, the vector $\boldsymbol{z}^U = (z_1^U, \ldots, z_m^U)$ is the ideal (or Utopia) point in the objective space, i.e., $z_j^U = \min\{f_j(x) \;\; \text{such that} \;\; x \in \Omega\}$, $j = 1, \ldots, m$. Each term can be view as a distance function that minimizes the distance between the solution point and the ideal point in the objective space. Minimizing $\Psi(x; \boldsymbol{w})$ can provide approximations to the complete Pareto optimal front by varying the vector of weights [8,10]. The function is not smooth at some points but the use of a derivative-free method to minimize $\Psi(x; \boldsymbol{w})$ overtakes this issue.

In our problem, z_1^U and z_2^U are known in advance, since f_1 and f_2 are constraint violations.

4.3 Weighted Tchebycheff Algorithm for OCP

In this section, the main ideas of the proposed methodology are presented in Algorithm 1. The algorithm has been designed to simultaneously minimize the three above defined objective functions $\theta(x)$, $p(x)$ and $M(T, x(T))$ using a weighted Tchebycheff scalarization approach that takes advantage of a penalty term to accelerate the convergence of the constraints violation to zero.

Algorithm 1 Weighted Tchebycheff algorithm with a penalty term for OCP

```
Require: $n$ (number of decision variables), $N_w$, $N_{runs}$
 1: Generate a set of $N_w$ weight vectors, $\boldsymbol{w}^i$, $i = 1, \ldots, N_w$ with positive components
 2: for $r = 1$ to $N_{runs}$ do
 3:   Compute $z_3^U$ and $Viol$ using Algorithm 2
 4:   Set $z_i^U = 0$, $i = 1, 2$
 5:   Given $x^0 \in \Omega$ based on the ODE initial conditions
 6:   for $j = 1$ to $N_w$ do
 7:     Set $Viol_{old} = Viol$
 8:     Compute $x(\boldsymbol{w}^j)$, an approximation to the subproblem

            $$\min_x \Psi(x; \boldsymbol{w}^j) + \mu_j \left( w_1^j f_1(x) + w_2^j f_2(x) \right),$$

        using $x^0$ as initial approximation, where $\mu_j = 2^\kappa$ and $\kappa = \lfloor \frac{j}{2} \rfloor$.
 9:     Set $Viol = f_1(x(\boldsymbol{w}^j)) + f_2(x(\boldsymbol{w}^j))$
10:     if $Viol < Viol_{old}$ then
11:       Update $z_3^U$ with the current value $f_3(x(w^j))$
12:     end if
13:     Update $x^0 = x(\boldsymbol{w}^j)$
14:     Set $F_i^{r,j} = f_i(x(\boldsymbol{w}^j))$, $i = 1, 2, 3$
15:   end for
16: end for
17: Identify the ND solutions among $(F_1^{r,j}, F_2^{r,j}, F_3^{r,j})$, $j = 1, \ldots, N_w$, $r = 1, \ldots, N_{runs}$.
```

Relative to the ideal point $\boldsymbol{z}^U$, the tested strategy considers $z_1^U = z_2^U = 0$ all over the iterative process and z_3^U is initially estimated using a payoff table, as shown in Algorithm 2. A large set of points in the decision space are randomly generated in Ω, the corresponding function vectors are evaluated and the smallest f_3 value is identified to give the estimate of z_3^U. There is a different estimated value for each run. During the inner cycle that runs for all vectors of weights $\boldsymbol{w}^j$, $j = 1, \ldots, N_w$ (from line 6 to line 15 in Algorithm 1) the z_3^U is updated with the most recent value of $f_3(x(\boldsymbol{w}^j))$, if the sum of f_1 and f_2 (therein called $Viol$) for that $\boldsymbol{w}^j$ has decreased relative to that of the previous $\boldsymbol{w}^{j-1}$.

Another important issue addressed in Algorithm 1 is the initial approximation provided to the NLP solver. The x^0 for the first subproblem (in the inner cycle), corresponding to the weights vector $\boldsymbol{w}^1$, is generated taking into account the ODE initial conditions. For the remaining subproblems, x^0 is the solutions of the previous subproblem (see lines 5 and 13).

Finally, the penalty term that is added to the Tchebycheff objective function (8) for the minimization in each subproblem is justified by the need to force even further and accelerate the decrease of the objective functions f_1 and f_2 (constraint violations θ (4) and p (5) respectively). The penalty parameter μ is set initially to one and doubles every two subproblems. See line 8 in Algorithm 1.

Algorithm 2 Generate payoff table to compute z_3^U

```
Require: $n$ (number of decision variables)
for $j = 1$ to $50n$ do
    Randomly generate $x^j \in \Omega$
    Compute $f_i^j \equiv f_i(x^j)$, $i = 1, 2, 3$
end for
Set $z_3^U = \min_{j=1,\ldots,50n} f_3^j$
Set $Viol = \max_{j=1,\ldots,50n}(f_1^j + f_2^j)$
```

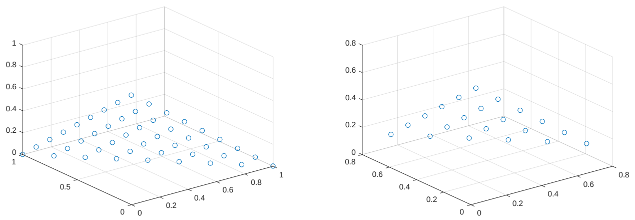
5 Numerical Results

In this preliminary study, the `fminsearch` from MATLAB® is tested to compute $x(w^i)$, in line 8 of the Algorithm 1. For all experiments, the options for `fminsearch` are set as follows: 'MaxFunEvals' = $1000n$, 'MaxIter' = $500n$ and 'TolFun' = $1e-04$, where n is the total number of decision variables in the finite optimization problem. Parameter values for all the illustrated problems are: number of runs, $N_{runs} = 5$ and number of subintervals in $[0, T]$, $N = 10$. We note that the addition of more subintervals gives very little improvement in the optimal objective values, but greatly increases the overall computational effort.

To generate the weight vectors (see line 1 of Algorithm 1), the simplex-lattice design method for generating an evenly distributed set of weights in a simplex is used. The constructive method for the creation of a $\{m, q\}$-simplex lattice, presented in [14], is used to obtain the uniformly distributed vectors of weights. With $m = 3$ (number of objectives) and $q = 8$ ($q + 1$ is the number of points on each axis), a total of 45 design points are created. Since only design points that have positive components seem adequate for this TOO problem, a total of $N_w = 21$ vectors of weights are selected. Figure 1 shows the 55 design points of the $\{3, 8\}$-simplex lattice and the 21 selected points of the simplex.

The best solution obtained by the algorithm is selected from the final computed ND solutions, after running the algorithm 5 times. It corresponds to the final ND solution that has the least value of $Viol$. The results are shown for six problems: "trajectory", "VanderPol", "obstacle", "reactor","Fuller" and "Tankreactor". Table 1 contains the best solutions obtained for the selected problems and Fig. 2 – 7 display:

- (a) the 3-dimensional approximation to the Pareto front
- (b) the 2-dimensional projection $f_1 - f_3$ of the Pareto front (for Problem 1) and the trajectory in the state space (for Problems 2 – 6)



(a) Points of the $\{3,8\}$-simplex (b) Selected 21 design points

Fig. 1. Design and selected points of the $\{3,8\}$-simplex lattice

- (c) the state variables of the selected best solution
- (d) the optimal control of the best solution.

We note that the solutions computed after repeating the process 5 times ($N_{runs} = 5$, corresponding to the outer cycle) are displayed in the (a) (and (b) for Problem 1) plots with a 'blue' filled circle and the final ND solutions (after the 5 runs) are identified with a 'red' bigger open circle.

Problem 1. "trajectory" – Find $u(t)$ that minimizes J (with $T = 3$ fixed):

$$\begin{aligned}
&\min_{u(t)} J \equiv \int_0^T \left(y^2(t) + u^2(t)\right) dt \\
&\text{s.t. } y'(t) = (1 + y(t))y(t) + u(t),\ \ t \in [0, T] \\
&\qquad y(0) = 0.05,\ \ y(T) = 0,\ \ |y(t)| \le 1,\ \ |u(t)| \le 1, t \in [0, T].
\end{aligned}$$

For Problem 1, the initial provided x^0 is $y(t_i) = 1$, $i \in I_N$ and $u(t_i) = 0$, $i \in I$. The best computed solution in terms of constraints violation (identified among the final 44 ND solutions) is $(\theta, p, M) = (7.8962\text{e-}10,\ 0,\ 0.023928)$, as depicted in Table 1. This table also shows the run (among the 5 runs) where the best solution was obtained, and j corresponds to the index of the weights vector $\boldsymbol{w}^j$ in that run. In Table 2, a comparison with other results in the literature is shown. This table displays t (time in seconds required to obtain the reported solution), fe (number of function evaluations required to produce the solution), as well as t_{avg} (average time for Algorithm 1 to produce a solution over the 5 runs and 21 weight vectors) and fe_{avg} (average value of function evaluations – over the 5 runs and 21 weight vectors). Figure 2 shows the plots above mentioned relative to this problem. The selected solution is very satisfactory and the profiles for the state and control are as expected.

Problem 2. "VanderPol" – Find $u(t)$ that minimizes J (with $T = 5$ fixed):

$$\begin{aligned}
&\min_{u(t)} J \equiv \tfrac{1}{2} \int_0^T \left(y_1^2(t) + y_2^2(t) + u^2(t)\right) dt \\
&\text{s.t. } y_1'(t) = y_2(t), \\
&\qquad y_2'(t) = -y_1(t) + (1 - y_1^2(t))y_2(t) + u(t),\ \ t \in [0, T] \\
&\qquad y_1(0) = 1,\ y_2(0) = 0,\ y_1(T) - y_2(T) + 1 = 0\ .
\end{aligned}$$

Table 1. Best selected solutions

problem	ND solutions	(θ, p, M)	run (j^a)
"trajectory"	44	(7.8962e-10, 0, 0.023928)	2 (14–21)
"VanderPol"	42	(2.0699e-07, 0, 1.716531)[b]	5 (21)
"obstacle"	8	(4.7033e-11, 0, 0.652339)[c]	1 (21)
"reactor"	52	(8.3010e-12, 0, 0.552876)	3 (21)
"Fuller"	62	(2.1857e-07, 0, 3.133665)[d]	1 (21)
"Tankreactor"	39	(6.4011e-08, 0, 0.026682)	1 (21)

[a] j corresponds to the index of the vector of weights ($j = 1, \ldots, N_w$).

[b] another interesting ND solution (4.14976e-05, 0, 1.692121) from run 5 (j=15).

[c] another interesting ND solution (4.3023e-10, 0, 0.185424) from run 4 (j=21).

[d] another interesting ND solution (3.2306e-05, 1.4921e-07, 3.105910) from run 3 (j=15).

Table 2. Solutions comparison

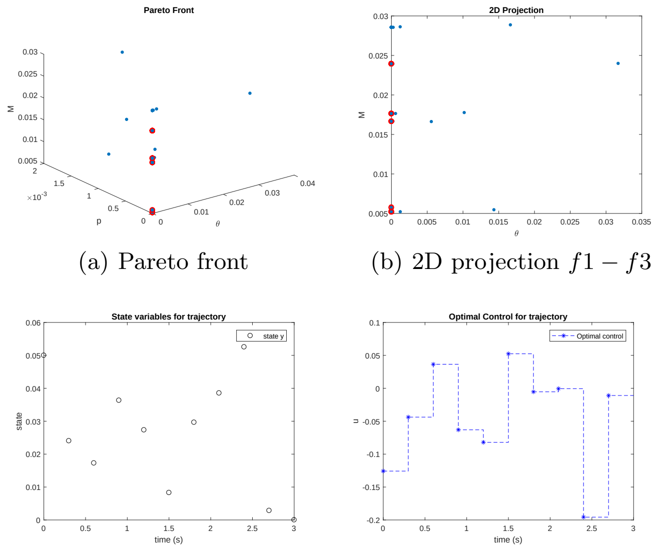
problem	Algorithm 1		other methods	
	fe/fe_{avg}	t/t_{avg}	$(\theta, p, M)/M$	fe (t)
"trajectory"	$413^*/3094$	$0.62^*/4.30$	(8.848e-11, 0, 0.264)[a]	21494 (22.5)
"VanderPol"	966/16943	0.54/9.77	$M = 1.6860$[b]	$6^\ddagger + 6^\dagger$ (0.31)
"obstacle"	1293/6676	0.72/3.92	(1.31e-08, 4.88e-10, 2.46)[a]	52702 (53.7)
			$M = 0.03180$[c]	– (0.66)
"reactor"	1505/16504	1.07/9.36	$M = 0.572162$[d]	– (0.29)
"Fuller"	1171/15959	0.74/9.26	$M = 3.0305914$[e]	–
"Tankreactor"	2223/20669	1.36/12.56	(9.950e-05, 0, 0.036)[a]	16320 (18.0)
			$M = 0.02680$[f]$/0.028196$[g]	–

[*] average of 8 runs with 8 weight values (equal function vectors). [‡] number of gradient evaluations; [†] number of function evaluations. [a] [5]; [b] [3]; [c] [15]; [d] [16]; [e] [1]; [f] [17]; [g] [18].

For Problem 2, the best computed ND solution was found among 42 final ND solutions. The initial approximation $y_1(t_i) = 1$, $y_2(t_i) = 0, i \in I_N$ and $u(t_i) = 0.5, i \in I$ is used, see Tables 1 and 2. Figure 3 displays the ND solutions, the trajectory in the state space, the state variables and control relative to the best solution. The two selected solutions and the profiles for the states and control (of the identified best solution) are similar to those shown in [3].

Problem 3. - "obstacle" Find $u(t)$ that solves (with $T = 2.9$ fixed):

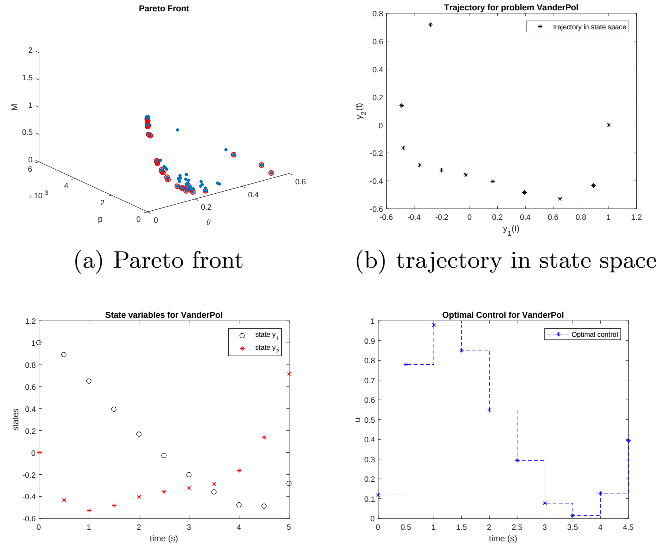
$$\begin{aligned}
\min_{u(t)} \; & J \equiv 5y_1(T)^2 + y_2(T)^2 \\
\text{s.t. } & y_1'(t) = y_2(t) \\
& y_2'(t) = u(t) - 0.1(1 + 2y_1(t)^2)y_2(t), \; t \in [0, T] \\
& y_1(0) = 1, \; y_2(0) = 1, \\
& 1 - 9(y_1(t) - 1)^2 - (\tfrac{y_2(t) - 0.4}{0.3})^2 \leq 0, \; -0.8 - y_2(t) \leq 0, \; |u(t)| \leq 1, \;\; t \in [0, T].
\end{aligned}$$



(a) Pareto front (b) 2D projection $f1 - f3$

(c) State variable (d) Control variable

Fig. 2. ND solutions and state and control variables for Problem 1 “trajectory”

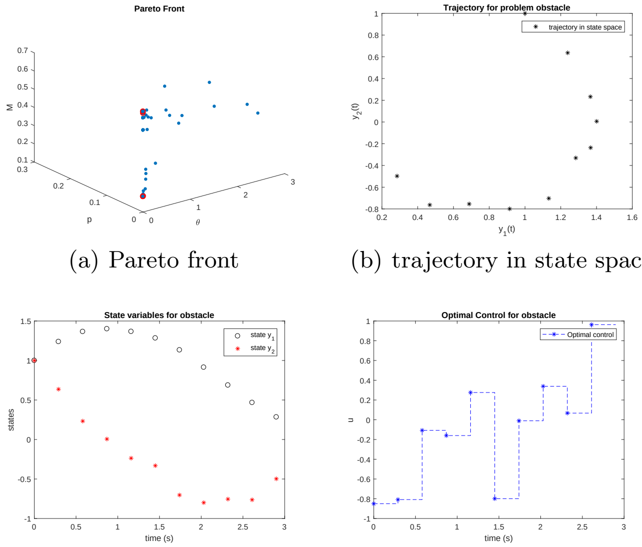


(a) Pareto front (b) trajectory in state space

(c) State variables (d) Control variable

Fig. 3. ND solutions, trajectory, and states and control for Problem 2 “VanderPol”

For Problem 3, the best solution was found among 8 final ND solutions. The initial approximation $y_1(t_i) = 1$, $y_2(t_i) = 1, i \in I_N$ and $u(t_i) = 1, i \in I$ is used and the results are shown in the Tables 1 and 2. Figure 4 displays the ND solutions, the trajectory in state space, the trajectory of the state variables and control relative to the best solution. The two solutions identified in Table 1 are satisfactory and the profiles of the state variables and control (relative to the defined best solution) follow the pattern shown in [15] (where larger numbers of subintervals in $[0, T]$ are used).



(a) Pareto front

(b) trajectory in state space

(c) State variable

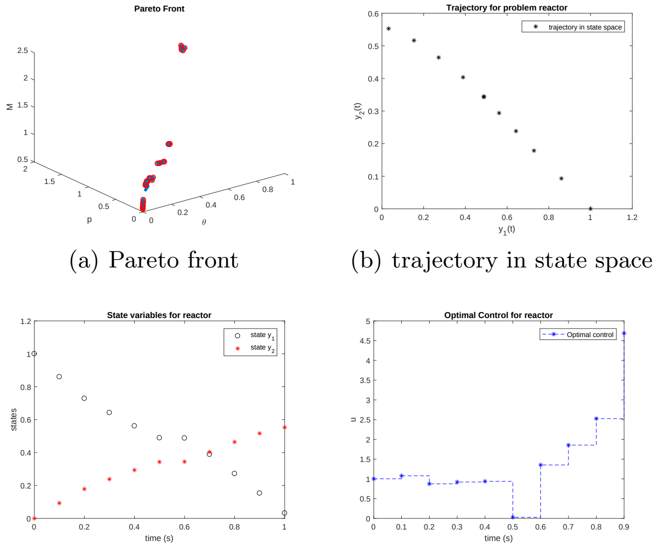
(d) Control variable

Fig. 4. ND solutions, trajectory and states and control for Problem 3 “obstacle”

Problem 4. “reactor” – Maximize yield of $y_2(t)$ after one hour operation by manipulating a transformed temperature $u(t)$:

$$\begin{aligned}
&\max_{u(t)} J \equiv y_2(T) \\
&\text{ s.t. } y_1'(t) = -y_1(t)\left(u(t) + \frac{u^2(t)}{2}\right) \\
&\qquad y_2'(t) = y_1(t)u(t), \;\; t \in [0, T] \\
&\qquad y_1(0) = 1, \;\; y_2(0) = 0, \; 0 \le y_1(t), y_2(t) \le 1, \;\; 0 \le u(t) \le 5, \;\; t \in [0, T].
\end{aligned}$$

For Problem 4 the best computed solution was found among 52 final ND solutions. The initial approximation $y_1(t_i) = 1$, $y_2(t_i) = 1, i \in I_N$ and $u(t_i) = 0, i \in I$ is used and the results are shown in the Tables 1 and 2. The usual plots



(a) Pareto front (b) trajectory in state space

(c) State variables (d) Control variable

Fig. 5. ND solutions, trajectory and states and control for Problem 4 "reactor"

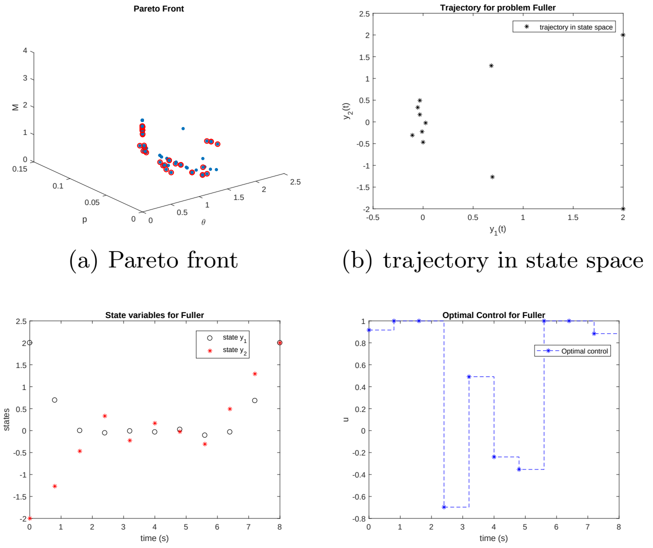
are shown in Fig. 5. The profiles of states and optimal control are identical to those shown in [16].

Problem 5. " Fuller" – Minimize J (with $T = 8$ fixed) in this OCP linear in the control:

$$\begin{array}{l} \min\limits_{u(t)} J \equiv \displaystyle\int_0^T y_1^2(t)\,dt \\ \text{s.t. } y_1'(t) = y_2(t), \\ \qquad y_2'(t) = u(t), \;\; t \in [0,T] \\ \qquad y_1(0) = 2,\, y_2(0) = -2,\, y_1(T) = 2,\, y_2(T) = 2,\, |u(t)| \leq 1,\, t \in [0,T]. \\ \qquad |u(t)| \leq 1, \;\; t \in [0,T] \;. \end{array}$$

The initial approximation used is $y_1(t_i) = 2, y_2(t_i) = -2, i \in I_N$ and $u(t_i) = 1, i \in I$. From the results shown in Tables 1 and 2 and Fig. 6, relative to Problem 5, we can conclude that the proposed methodology is able to compute reasonably good solutions and the profiles of the trajectory in state space, the trajectory of the states y_1 and y_2, and the control u, are identical to those presented in [1].

Problem 6. "Tankreactor" – In a continuous stirred-tank chemical reactor, y_1 represents the deviation from the steady-state temperature, y_2 represents the deviation from the steady-state concentration and u is the effect of the coolant



(a) Pareto front (b) trajectory in state space

(c) State variables (d) Control variable

Fig. 6. ND solutions, trajectory and state and control variables for Problem 5 "Fuller"

flow on the chemical reaction:

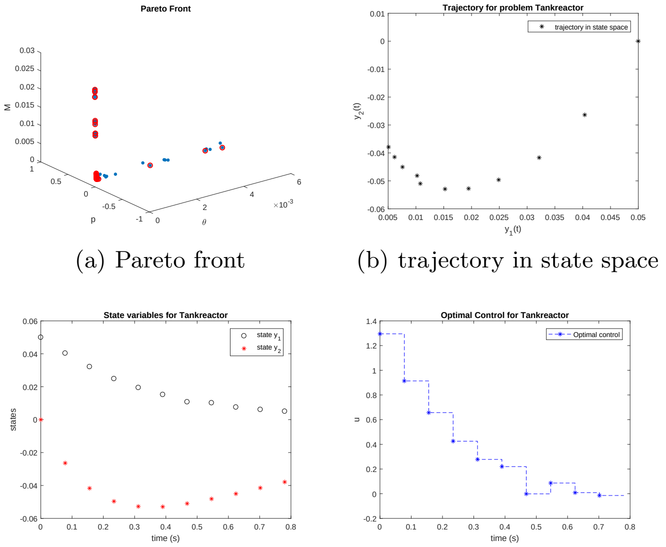
$$
\begin{aligned}
&\min_{u(t)} J \equiv \int_0^T (y_1(t)^2 + y_2(t)^2 + Ru(t)^2)\,dt \\
&\text{ s.t. } y_1'(t) = -2(y_1(t)+0.25) + (y_2(t)+0.5)\exp\left(\tfrac{25y_1(t)}{y_1(t)+2}\right) \\
&\qquad -(y_1(t)+0.25)u(t) \\
&\qquad y_2'(t) = 0.5 - y_2(t) - (y_2(t)+0.5)\exp\left(\tfrac{25y_1(t)}{y_1(t)+2}\right),\quad t \in [0,T] \\
&\qquad y_1(0) = 0.05,\ y_2(0) = 0.
\end{aligned}
$$

The optimal solution reported in [17], for $T = 0.78$ and $R = 0.1$, is $J^* = 0.02680$. Using the initial approximation $y_1(t_i) = 0.05$, $y_2(t_i) = 0, i \in I_N$ and $u(t_i) = 0, i \in I$, the results obtained by Algorithm 1 are shown in Tables 1 and 2. The identified best solution is very satisfactory and the resulting plots (c) and (d) in Figure 7 are similar to those in [18].

6 Conclusions

A weighted Tchebycheff scalarization methodology is proposed to solve a finite-dimensional nonlinear optimization problem that arises from the use of a direct multiple shooting method when applied to an OCP.

The proposed methodology aggregates three objective functions. Two of them measure constraint violations – from the "continuity constraints" and from the



(a) Pareto front

(b) trajectory in state space

(c) State variables

(d) Control variable

Fig. 7. ND solutions, trajectory and states and control for Problem 6 “Tankreactor”

algebraic state and control constraints – that are required to be as close as possible to zero, and the other is the optimality measure – the objective functional from the OCP. Moreover, a penalty term is added to the Tchebycheff objective to force the constraint violations to decrease even faster. The preliminary numerical experiments show the effectiveness of the methodology when compared to similar strategies. A significant reduction in function evaluations (and time) and objective function improvements are achieved with the proposed methodology, when compared to the multiple shooting descent-based filter method presented in [5]. The comparison with other direct multiple shooting techniques (and even with indirect methods) shows similar computational effort and very good approximated solutions in general.

Acknowledgments. The authors wish to thank two anonymous referees for their comments and suggestions to improve the paper.

References

1. Frego, M.: Numerical Methods for Optimal Control Problems with Applications to Autonomous Vehicles. Ph.D. Thesis, University of Trento (2014)
2. Assassa, F., Marquardt, W.: Dynamic optimization using adaptive direct multiple shooting. Comput. Chem. Eng. **60**, 242–259 (2014)
3. Bock, H.G. and Plitt, K.J.: A multiple shooting algorithm for direct solution of optimal control problems. IFAC Proceedings Volumes, **17**(2), 1603–1608 (1984)

4. Diehl, M., Bock, H.G, Diedam, H., Wieber, P.-B.: Fast direct multiple shooting algorithms for optimal robot control. In: Diehl, M., Mombaur, K. (eds.) Fast motions in biomechanics and robotics. Lecture Notes in Control and Information Sciences **340**, pp. 65–93 (2006)
5. Ramadas G.C.V., Fernandes, E.M.G.P., Rocha, A.M.A.C., Costa, M.F.P.: A multiple shooting descent-based filter method for optimal control problems. In Gaspar-Cunha, A. et al. (eds.), Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences, Computational Methods in Applied Sciences **55**, pp. 377–392 (2020)
6. Angelo, J.S., Barbosa, H.J.C.: On ant colony optimization algorithms for multiobjective problems. In: Ostfeld, A. (Ed.), Ant Colony Optimization - Methods and Application, InTech Europe pp. 53–74 (2011)
7. Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering. Struct. Multidisc. Optim. **26**, 369–395 (2004)
8. Miettinen, K.: Nonlinear Multiobjective Optimization, Kluwer Academic Publishers (1999)
9. Deb, K.: Multi-objective Optimization using Evolutionary Algorithms. John Wiley & Sons (2001)
10. Feng, Z., Zhang, Q., Zhang, Q., Tang, Q., Yang, T., Ma, Y.: A multiobjective optimization based framework to balance the global and local exploitation in expensive optimization. J. Glob. Optim. **61**, 677–694 (2015)
11. Zhang, Q., Liu, W., Tsang, E., Virginas, B.: Expensive multiobjective optimization by MOEA/D with Gaussian process model. IEEE T. Evolut. Comput. **14**(3), 456–474 (2010)
12. Steuer, R.E., Choo, E.U.: An interactive weighted Tchebycheff procedure for multiple objective programming. Math. Program. **26**, 326–344 (1983)
13. Emmerich, M.T.M., Deutz, A.H.: A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Nat. Computing **17**, 585–609 (2018)
14. Das, I., Dennis, J.: Normal-boundary intersection: An alternative method for generating Pareto optimal points in multicriteria optimization problems, Institute for Computer Application in Science an Engineering, NASA Langley Research Center - Hampton, VA 23681-0001, Tech. Rep. (1996)
15. Schlegel, M., Stockmann, K., Binder, T., Marquardt, W.: Dynamic optimization using adaptive control vector parameterization. Comput. Chem. Eng. **29**(8), 1731–1751 (2005)
16. Tamimi, J., Li, P.: Nonlinear Model Predictive Control Using Multiple Shooting Combined with Collocation on Finite Elements. IFAC Proceedings Volumes, **42**(11), 703–708 (2009)
17. Kirk, D.E.: Optimal Control Theory: An Introduction. Dover Publications, Inc. (2004)
18. Wang, X.: Solving optimal control problems with Matlab – Indirect methods. ISE. Dept., NCSU, Raleigh, NC 27695, Tech. Rep. (21 pages) In http://solmaz.eng.uci.edu/Teaching/MAE274/SolvingOptContProb_MATLAB.pdf