SOLUTION OF SINGULAR OPTIMAL CONTROL PROBLEMS USING THE IMPROVED DIFFERENTIAL EVOLUTION ALGORITHM

Fran Sérgio Lobato[1], Valder Steffen Jr [2] Antônio J. Silva Neto[3]

[1]*School of Chemical Engineering, Federal University of Uberlândia*
e-mail:fslobato@feq.ufu.br

[2] *School of Mechanical Engineering, Federal University of Uberlândia,*
Av. João Naves de Ávila 2121, Campus Santa Mônica, P.O. Box 593,
38408-144, Uberlândia-MG, Brazil
e-mail:vsteffen@mecanica.ufu.br

[3] *Department of Mechanical Engineering and Energy, Polytechnic Institute - IPRJ,*
State University of Rio de Janeiro, Rua Alberto Rangel, s/n°,
Vila Nova 28630-050, Nova Friburgo-RJ, Brazil
e-mail:ajsneto@iprj.uerj.br

Abstract

The Differential Evolution algorithm, like other evolutionary techniques, presents as main disadvantage the high number of objective function evaluations as compared with classical methods. To overcome this disadvantage, this work proposes a new strategy for the dynamic updating of the population size to reduce the number of objective function evaluations. This strategy is based on the definition of convergence rate to evaluate the homogeneity of the population in the evolutionary process. The methodology is applied to the solution of singular optimal control problems in chemical and mechanical engineering. The results demonstrated that the methodology proposed represents a promising alternative as compared with other competing strategies.

1 Introduction

The Optimal Control Theory has been developed for over forty years. With the advances of digital computer techniques, associated with the popularity of dynamic simulation tools, and the existence of a competitive global market and environmental constraints, new methodologies for optimal control have been proposed. Numerous applications using the Optimal Control Theory can be found in fields of science and engineering, such as aerospace, process control, robotics, bioengineering, economics, finance, and management science [1, 2].

The Optimal Control Problem consists in the determination of the control variable profiles that minimize (maximize) a given performance index. For this problem, when the differential index fluctuation occurs as due to the activation and deactivation of inequality constrains or when the control variable behaves linearly in the Hamiltonian function, this is called Singular Optimal Control Problem (SOCP) [1].

Several numerical methods have been proposed in the literature to solve SOCP [1, 3]. They are usually classified according to three broad categories regarding their underlying formulation: direct optimization methods, Pontryagin's Maximum Principle (PMP) based methods, and HJB-based (Hamilton-Jacob-Bellman) methods. Among these methods, the direct approach has been preferentially used in the last several years. This approach uses control parameterization or state and control parameterizations, transforming the original problem into a finite dimensional optimization problem. The solution of nonlinear programming problems (NLP) of great dimension or the attainment of the gradients of the objective function in the sequential method is not trivial [2, 4, 5, 6].

Recently, algorithms based on heuristic approaches have been used to solve the SOCP due to the success observed in the solution of general optimization problems. In this context, the Differential Evolution Algorithm (DE), proposed by Storn and Price [7], has been applied successfully. Kapadi and Gudi [8] determined the substrate concentration profiles in a feed-batch reactor with singular arc. Lobato et al. [9] presented a new algorithm for dealing with control optimization problems. The proposed methodology consists in the extension of the DE to problems with multiple objectives, through the incorporation of mechanisms such as rank ordering and neighborhood potential solution-candidates exploration. This algorithm is applied to determine the switching times (events) and operation time of the lysine fermentation process. Wang and Chiou [10] proposed an algorithm based on DE to determine the optimal control and optimal time location problems of differential-algebraic systems. Recently, Chowdhury et al. [11] presented a hybrid evolutionary direct search technique based on DE to solve optimal control problems.

In spite of numerous applications, this algorithm uses fixed population size during the evolutionary process. According to Vellev [12] this characteristic can affect the robustness and the computational cost of the algorithms. Small population size may result in local convergence; large population size will increase computational efforts and may lead to slow convergence. So, an appropriate population size can assure the effectiveness of the algorithm.

In this context, the main goal of this paper is to introduce a systematic methodology to find the control strategy and the switching times (events) by using the Differential Evolution Algorithm with a new strategy for the dynamic updating of the population size. This work is organized as follows. Sections 2 and 3 present the general aspects regarding the SOCP and singular arcs, respectively. A review dedicated to the DE technique is presented in Section 4. In Section 5 the proposed methodology is described. The results and discussion are presented in Section 6. Finally, the conclusions are outlined in Section 7.

2 Singular Optimal Control Problems

Mathematically, the SOCP can be formulated as follows [1]:

$$\min_{u(t),t_f} J = \Psi(z(t_f),t_f) + \int_{t_0}^{t_f} L(z,u,t)dt \tag{1}$$

where z is the state variables vector and u is the control variables vector. Ψ and L are the first and second terms of the performance index, respectively. The objective is subject to the implicit Differential-Algebraic Equations (DAE) system as given by:

$$f(\dot{z},z,u,t) = 0 \tag{2}$$

$$g(z,u,t) \leq 0 \tag{3}$$

$$p(u,t) \leq 0 \tag{4}$$

$$q(z,u,t)|_{t=t_f} = 0 \tag{5}$$

with consistent initial conditions given by:

$$\varphi(\dot{z}(t_0),z(t_0),u(t_0),t_0) = 0 \tag{6}$$

where $J(.)$, $L(.)$, $\Psi(.) \rightarrow \mathbb{R}$; $f(.),\varphi(.) \rightarrow \mathbb{R}^{m_z}$; $z \in \mathbb{R}^{m_z}$; $u \in \mathbb{R}^{m_u}$; $g \in \mathbb{R}^{m_g}$; $p \in \mathbb{R}^{m_p}$ and $q \in \mathbb{R}^{m_q}$.

According to the Optimal Control Theory [1, 2, 3], the solution of the SOCP, defined by Eq.(1) to Eq.(6), is satisfied by the co-state equations and the stationary condition given, respectively, by:

$$\dot{\lambda}^T \equiv -\frac{\partial H}{\partial z} \qquad \lambda(t_f) = \left.\frac{\partial \Psi}{\partial z}\right|_{t=t_f} \tag{7}$$

$$\frac{\partial H}{\partial u} = 0 \tag{8}$$

where H is the Hamiltonian function defined by:

$$H \equiv L + \lambda^T f \tag{9}$$

The system formed by Eq.(7) to Eq.(9) is known as the Euler-Lagrange equations, which are characterized by boundary value problems (BVPs). According Bryson and Ho [1] and Feehery [2] the main difficulties associated with the SOCP solution are the following: the existence of end-point conditions or region constraints implies in multipliers and associated complementary conditions that significantly increase the difficulty of solving the BVP by the indirect method; the existence of constraints involving the state variables and the application of slack variables method may originate DAE of higher index, regardless the constraint activation status, even in problems where the number of inequality constraints is equal to the number of control variables; the Lagrange multipliers can be very sensitive to the initial conditions.

3 Singular Arcs

The solution of SOCP presents special challenges because it demands the knowledge of the sequence and the number of constraint activations and deactivations (events) along the trajectory. When the amount of constraints is reduced, it is usually possible to determine this sequence by examining the solution of the problem without constraints. However, the presence of a large number of restrictions brings a problem of combinatorial nature [13, 14]. In the events, due to discontinuities in the state and/or the co-state variables, changes in the functional form of the DAE and/or in the trajectories of the control variable in each phase may occur. As a consequence, the differential index of the system can change throughout the solution trajectory, increasing when the inequality becomes active. The existence of sections of fluctuating index leads to different errors in these sections, demanding the application of adjusted numerical strategies for each section. Therefore, it is necessary to know previously the moments of activation and deactivation of the restrictions in order to solve the problem adequately. Another difficulty is the presence of singular arcs, where the second derivative matrix of the Hamiltonian with respect to the control is only positive semi-definite [1, 2].

A particular case of great interest is the one that appears when the control variable behaves linearly in the Hamiltonian function. In general, no minimum optimal solution will exist for such problems unless inequality constraints in the state and/or control are specified. If the inequality constraints are linear with respect to the control variable, it is reasonable to expect that the minimum solution, if it exists, will always impose that the control variables are located at a point belonging to the border of the viable region of control [1, 3, 8, 9].

Consider the following system of equations:

$$\dot{z} = F_1(z) + F_2(z)\,u z(t_o) = z_o \tag{10}$$

with the control variable given by:

$$u_{\min} \leq u \leq u_{\max} \tag{11}$$

The Hamiltonian function is defined as:

$$H = \lambda^T (F_1(z) + F_2(z)\,u) \tag{12}$$

For this class of control we have:

$$u = \begin{cases} u_{\max} & \lambda^T F_2 < 0 \\ \Im & \lambda^T F_2 = 0 \\ u_{\min} & \lambda^T F_2 > 0 \end{cases} \tag{13}$$

where $\Im$ is the Switching Function [3, 9, 30].

4 The Differential Evolution Algorithm

Differential Evolution (DE) is an optimization technique that belongs to the family of evolutionary computation, which differs from other evolutionary algorithms in the mutation and recombination schemes. DE executes its mutation operation by adding a weighted difference vector between two individuals to a third individual. Then, the mutated individuals will perform discrete crossover and greedy selection with the corresponding individuals from the last generation to produce the offspring.

The key control parameters for DE are the following: *NP* - the population size, *CR* - the crossover constant, and *F* - the weight applied to the random differential (scaling factor).

A classical DE algorithm is presented bellow [15, 16].

Algorithm: Differential Evolution

Initialize and evaluate population P

while (not done) {

for (i = 0; $i < N$; i++) {

Create candidate $C[i]$

```
        Evaluate C[i]
        if (C[i] is better than P[i])
        P′[i] = C[i]
        else
        P′[i] = P[i]
    }
    P = P′
}
```

Algorithm: Create candidate $C[i]$

```
Randomly select parents P[i1], P[i2], and P[i3]
where i, i1, i2, and i3
are different.
Create initial candidate
    C′[i] = P[i1] + D × (P[i2] − P[i3]).
Create final candidate C[i] by crossing over the
genes of P[i] and
C′[i] as follows:
for (j = 0; j < N; j++) {
    if (r < CR)
        C[i][j] = C′[i][j]
        else
        C[i][j] = P[i][j]
}
```

NP is the population size, *P* is the population of the current generation, and *P'* is the population to be formed for the next generation, $C[i]$ is the candidate solution with population index i, $C[i][j]$ is the j-th entry in the solution vector of $C[i]$ and r is a random number between 0 and 1.

Price and Storn [7] have given some simple rules for choosing the key parameters of DE for general applications. Normally, *NP* should be about 5 to 10 times the dimension (number of parameters in a vector) of the problem. As for F, it lies in the range 0.4 to 1.0. Initially $F = 0.5$ can be tried, then F and/or *NP* is increased if the population converges prematurely.

Storn et al. [15] proposed various mutation schemes for the generation of new vectors (candidate solutions) by combining the vectors that are randomly chosen from the current population as shown:

– rand/1: $x = x_{r1} + F\,(x_{r2} - x_{r3})$

– rand/2: $x = x_{r1} + F\,(x_{r2} - x_{r3} + x_{r4} - x_{r5})$

– best/1: $x = x_{best} + F\,(x_{r2} - x_{r3})$

– best/2: $x = x_{best} + F\,(x_{r2} - x_{r3} + x_{r4} - x_{r5})$

– rand/best/1: $x = x_{r1} + F\,(x_{best} - x_{r1} + x_{r1} - x_{r2})$

– rand/best/2: $x = x_{r1} + F\,(x_{best} - x_{r1}) + {}$ $+F\,(x_{r1} - x_{r2} + x_{r3} - x_{r4})$

DE has been successfully tested in various fields, such as: solution of multi-objective optimal control problems with index fluctuation applied to fermentation process [9], digital filter design [17], synthesis and optimization of heat integrated distillation system [18], multi-objective optimization of mechanical structures [19], solution of inverse radiative transfer problems in two-layer participating media [21], estimation of drying parameters in rotary dryers [20], apparent thermal diffusivity estimation of the drying of fruits [22], Gibbs free energy minimization in a real system [23], estimation of space-dependent single scattering albedo in radiative transfer problems [24, 25, 26], design of fractional order PID controllers [27], and other applications [15, 16].

4.1 Dynamic Updating of the Population Size

All classical selection algorithms keep the population size fixed during the evolutionary process. This aspect simplifies the algorithms but it is an artificial restriction and does not follow any analogy to the biological evolution, where the number of individuals in a population varies continuously with time, increasing when there are highly-fitted individuals and abundant resources, and decreasing otherwise. Intuition hints that it may be beneficial for the population to expand in the early generations when there is high phenotype diversity and there is opportunity to experiment with different characteristics of the individuals, and to shrink with the increase of population convergence, when the unification of the individuals in terms of structure and fitness no longer justifies the maintenance of a large population and the higher computational costs associated with it [12].

In this context, Sun et al. [28] studied the influence of two strategies for the dynamic updating of the individuals number of the population during the evolutionary process:

$$NP = (NP_{\max} - NP_{\min}) \left(\frac{Iter_{\max} - iter}{Iter_{\max}} \right) + NP_{\min} \tag{14}$$

$$NP = \max \left(\frac{NP_{\max}}{2} \sin \left(\frac{iter}{A} \right) + \frac{NP_{\max}}{2}, NP_{\min} \right) \tag{15}$$

where $Iter_{max}$ is the generation maximum number, $iter$ is the current generation, NP_{min}, and NP_{max}, are the minimal number and maximum number of individuals in the population, respectively and A is the amplitude in sin function.

In this work, an adaptive population size method with partial increasing or decreasing number of individuals according to diversities in the end of each generation is adopted. Besides, initially the convergence rate (ω) is defined as:

$$\omega = \frac{f_{average}}{f_{worst}} \tag{16}$$

where $f_{average}$ is the average value of the objective function and f_{worst} is the worst value of the objective function. Consequently, the defined convergence rate evaluates the homogeneity of the population in the evolutionary process: if ω is close to zero, e.g., the value worst of the objective function is different of the average value of the objective function; if ω is close to one, the population is homogeneous. Thus, a simple equation for dynamic updating of the population size is proposed:

$$NP = \text{round}(NP_{\min}\omega + NP_{\max}(1-\omega)) \tag{17}$$

where the operator round(.) indicates the rounding to the nearest integer.

It should be emphasized that the equation Eq.(17) updates the population size as based on the convergence rate, differently from Eq.(14) and Eq.(15) in which any information regarding the evolution of the process is considered. This modification in the canonical DE algorithm is defined as Improved Differential Evolution algorithm (IDE).

5 Methodology

The methodology proposed in this work consists in transforming the original SOCP with differential index fluctuation, into a nonlinear optimization problem with constant differential index (equal to one).

Let the time interval $t \in [t_o, \ ..., \ t_f]$ be discretized using N time nodes, t_i, such that $t_0 = t_o < t_1 < ... < t_N = t_f$. In each subinterval $t \in [t_i \ \ t_{i+1}]$, i=1, 2, ..., N, let the control input be approximated by

$$u \equiv u_i \quad \text{for} \quad t_i \leq t \leq t_{i+1} \tag{18}$$

With the piecewise linear approximation the unknown control input $u(t)$ is replaced by N unknown parameters $u_1, u_2, ..., u_N$. Besides, in this formulation, the localization of each event t_i also is unknown and should be calculated, resulting in 2N-1 design variables. The resulting non-linear optimization problem (with differential index equal to one) is solved by using IDE.

6 Results and Discussion

In order to evaluate the performance of the IDE algorithm, the Adaptive Differential Evolution algorithm ADE(i) is considered, e.g., using the Eq.(14) (i=1) and Eq.(15) (i=2) to update the population size in DE algorithm. In this sense, three classical SOCP, with different level of complexity, are considered in this section. For evaluating the methodology proposed in this work, some practical points regarding the application of the procedure should be emphasized:

- The parameters used by DE algorithm are the following [7, 8, 9, 10]: 25 individuals, 200 generations, perturbation rate and crossover probability equal to 0.8 and $DE/rand/1/bin$ strategy for the generation of potential candidates.

- The parameters used by IDE algorithm are the following [7, 8, 9, 10]: same used by DE algorithm, minimal and maximum numbers of individuals in the population equals to 5 and 25, respectively.

- The parameters used by the algorithms ADE(1) and ADE(2) are the following [28]: same used by IDE algorithm and the amplitude assigned to the *sin* function equal to 2.0.

- The stopping criteria used: when the difference between the best value and the average value of population is smaller than 10^{-8}, the optimization process is finalized.
- All case studies were run 20 times independently to obtain the average values shown in the upcoming tables.
- To initialize the random generator in each simulation the following seeds are used: [0 1 2 3 4 ... 19]. This procedure allows a better comparison among the evolutionary strategies utilized in this work.
- It should be emphasized that to solve the system of differential equations the well known Runge-Kutta Method 4-5th order was used.

6.1 Catalyst Mixing Problem

This problem considers in a plug-flow reactor packed with two catalysts and involving the reactions proposed by Gunn and Thomas [29], and subsequently considered by Logsdon [30], and Vassiliadis [31]:

$$S_1 \underset{k_2}{\overset{k_1}{\leftrightarrow}} S_2 \overset{k_3}{\rightarrow} S_3$$

The symbols k_1 and k_2 are, respectively, the reaction rate constants of the first two reactions in a reactor where the catalyst consists entirely of the substance which catalyzes the reversible reactions $S_1 \leftrightarrow S_2$, while the symbol k_3 is the reaction rate constant of the third reaction in a reactor where the catalyst consists entirely of the substance which catalyzes the reaction $S_1 \leftrightarrow S_2$. The optimal mixing policy of the two catalysts has to be determined in order to maximize the production of species S_3:

$$J = 1 - x_1(t_f) - x_2(t_f) \quad (19)$$

subject to mass balance of species S_1 (x_1) and S_2 (x_2),

$$\frac{dx_1}{dt} = u(10x_2 - x_1) \quad x_1(0) = 1 \quad (20)$$

$$\frac{dx_2}{dt} = u(x_1 - 10x_2) - (1-u)x_2 \quad x_2(0) = 0 \quad (21)$$

$$0 \leq u \leq 1, \quad t_f = 1 \quad (22)$$

where t represents the residence time of the substances from the instant of entry to the reactor. The catalyst blending fraction u is the fraction of the catalyst formed by the substance that catalyzes the reaction $S_1 \leftrightarrow S_2$. This fraction can be varied along the axial position of the reactor.

This classical problem with differential index equals to 3 was first posed by Gunn and Thomas [29] and has been solved by Logsdon [30] using orthogonal collocation on finite elements, by Vassiliadis [31] using the control parameterization technique, by Lobato [3] using a hybrid approach (direct optimization methods associated with PMP method), and by Lobato and Steffen [32] by using the control parameterization technique and the Multi-Particle Collision Algorithm (MPCA).

Tables 1 and 2 present a comparison of the results obtained (objective function, number of objective function evaluations (N_{eval}), events localization and control strategy) using DE, ADE(1), ADE(2) and IDE algorithms and other optimal control techniques. In this case 3 control elements (N=3) were used, and consequently, 5 (2N-1) design variables were used into account in all evolutionary strategies. In this first table it is possible to observe that all the algorithms with the strategy of dynamic ally updating the population size observes a reduction in the number of objective function evaluations. The best performance was obtained by IDE (32% with respect to DE canonical).

Table 1. Comparison of the results obtained for the catalyst mixing problem using various optimal control techniques.

Reference	J (Eq.(19))	N_{eval}
NPSOL [31]	0.048055	-
SNOPT [33]	0.048080	-
IA [3]	0.048057	-
MPCA [32]	0.047732	12000
DE	0.048080	5025
ADE(1)	0.048069	4025
ADE(2)	0.047990	3850
IDE	0.048079	3220

Table 2. Events and control strategy.

	t_{s1}	t_{s2}	u_1	u_2	u_3
[3]*	0.128	0.737	1.000	0.226	0.000
[32]**	0.129	0.732	1.000	0.227	0.000
DE	0.128	0.733	1.000	0.227	0.000
ADE(1)	0.128	0.734	1.000	0.226	0.000
ADE(2)	0.127	0.734	1.000	0.227	0.001
IDE	0.128	0.733	1.000	0.226	0.000

Figure 1 presents the evolution of the best values found for the objective function. In this figure, it is possible to observe that the optimal solution is found after 20 generations. The following generations are used for the refinement of the solution.

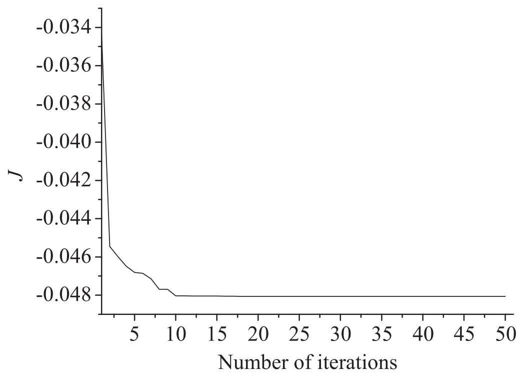

Figure 1. Objective function (J) versus number of iterations.

Figures 2 and 3 show the state and control variables profiles obtained by using IDE.

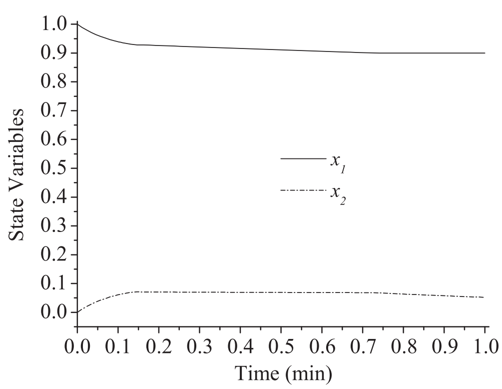

Figure 2. State variables profiles.

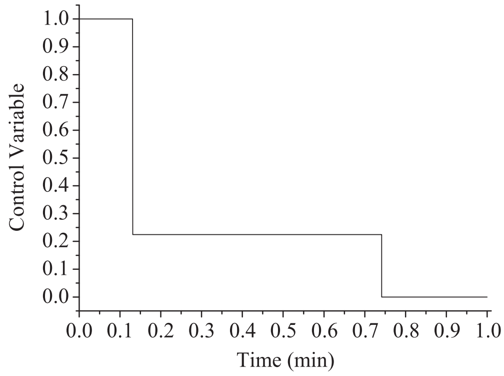

Figure 3. Control variable profile.

6.2 Batch Reactor

Let us first consider the consecutive chemical reaction $A \rightarrow B \rightarrow C$ in a batch reactor, as presented by Bilous and Amundson [34], and further used as an example for practical control by Marroquin and Luyben [35], Luus and Okongwu [36] using iterative dynamic programming (IDP), and by Lobato and Steffen [32] using the control parametrization technique and the Multi-Particle Collision algorithm.

In the consecutive reaction scheme it is required to maximize the production of the desired component B. The reaction in each step is assumed as a first order one, so that the system is described by the two following differential equations:

$$\frac{dx_1}{dt} = -k_1 x_1 x_1(0) = 0.95 \tag{23}$$

$$\frac{dx_2}{dt} = k_1 x_1 - k_2 x_2 x_2(0) = 0.05 \tag{24}$$

where x_1 is the concentration (mol/l) of the reactant A and x_2 is the concentration of the desired product B. The symbols k_l and k_2 are, respectively, the reaction rate constants given by

$$k_1 = 5.35 \times 10^{10} \exp\left(\frac{-9000}{u}\right) \tag{25}$$

$$k_2 = 4.61 \times 10^{17} \exp\left(\frac{-15000}{u}\right) \tag{26}$$

and the batch time is specified as t_f=30 min. The performance index to be maximized is the concentration of the component B at the specified final time, e.g.,

$$J = x_2(t_f) \tag{27}$$

The standard optimal control problem is then to find the temperature profile (u), so that the performance index expressed by Eq.(27) is maximized. The control variable is bounded by

$$300 \leq u \leq 400 \quad (28)$$

Table 3 presents a comparison of the results obtained by using the algorithms DE, ADE(1), ADE(2) and IDE and by other optimal control techniques. All evolutionary algorithms used 20 control elements, and consequently, 39 design variables.

Table 3. Comparison of the results obtained for the batch reactor problem using various optimal control techniques.

Reference	J (Eq.(27))	N_{eval}
IDP [36]	0.768370	-
MPCA [32]	0.768311	10250
DE	0.768369	4525
ADE(1)	0.768299	3850
ADE(2)	0.768270	3700
IDE	0.768370	2950

In this table all the algorithms with strategy of dynamic ally updating the population size observes a reduction in the number of objective function evaluations (N_{eval}). The best performance was obtained by IDE (35% with relation to DE canonical).

Figure 4 presents the evolution of the best values found for the objective function. Similarly to the previous case, with 100 generations, the optimal solution is found.

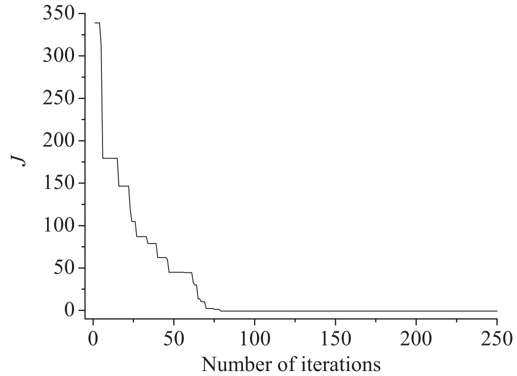

Figure 4. Objective function (J) versus number of iterations.

Figures 5 and 6 presents the state and control variables profiles simulated by using the best design variables obtained by IDE.

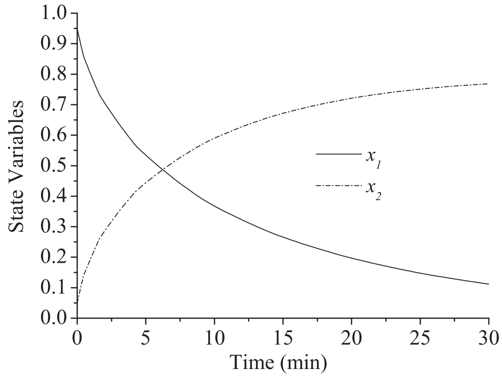

Figure 5. State variables profiles.

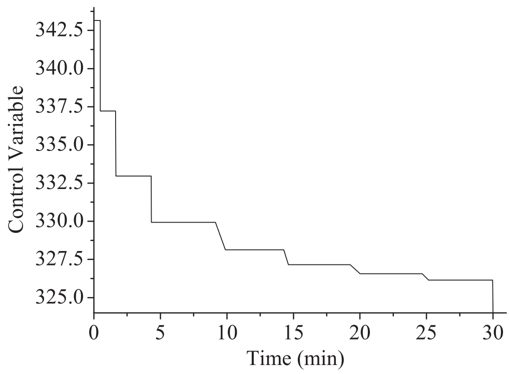

Figure 6. Control variable profile.

6.3 Goddard Problem

This problem was first proposed by the American rocket pioneer R. H. Goddard in 1919 when he was building a rocket to be fired vertically to reach high altitudes (a *sounding rocket*). It can be stated as follows: find the thrust profile to maximize the final altitude of a sounding rocket, given the initial mass, the fuel mass, and the drag characteristics of the rocket. The equations of motion are [1, 37, 38] to the following:

$$x_3\dot{x}_1 = u - D(x_1, x_2) - x_3G(x_2) - x_3G(x_2) \quad (29)$$

$$\dot{x}_2 = x_1 \quad (30)$$

$$c\dot{x}_3 = -u \quad (31)$$

where $[x_1(0)\, x_2(0)\, x_3(0)] = [x_{1o}\, x_{2o}\, x_{3o}]$ is the initial conditions vector, x_1 is the vertical velocity, x_2 is the radial distance from the center of the Earth, x_3 is the mass of the rocket, u is the rocket thrust, D is the aerodynamic drag, g is the gravitational force per unit of mass, c is the specific impulse of the rocket fuel (a constant). The fuel mass is $x_{3o} - x_{3f}$. We wish to find u to maximize $x_2(t_f)$ with $x_3(t_f) = x_{3f}$ and bounds on the rocket thrust:

$$J = \max x_2(t_f) \quad (32)$$

$$0 \leq u \leq u_{\max} \quad (33)$$

This problem presents a singular control due to the presence of the control variable u in the linear form. Besides, as presented in Eq.(13), there are three possibilities for the control strategy: bang-singular-bang.

Goddard Problem parameters [1, 37, 38]: c=0.5, β=500 miles^{-1}, x_{3o}=1, x_{3f}=0.6, u_{max}=3.5, $\rho_o C_D$=620, G_o=32.2 kft/sec^2 and x_{2o}=0 (Earth radius = 2.1×10^7 miles). In this paper the following expression for D and G were considered:

$$D = \frac{1}{2}\rho_o x_2^2 C_D S \exp(-\beta x_1) \quad (34)$$

$$G = \frac{1}{x_1^2} \quad (35)$$

Table 4 presents the results obtained by using the DE, ADE(1), ADE(2) and IDE algorithms. All evolutionary strategies used 40 control elements, and consequently, 79 design variables.

Table 4. Comparison of the results obtained for the Goddard problem.

Reference	J(Eq(32))	t_{s1}	t_{s2}	t_f	N_{eval}
NPSOL [2]	50.874	18.97	58.44	160.24	-
DE	50.874	18.97	58.44	160.24	5000
ADE(1)	50.873	18.96	58.43	160.22	3850
ADE(2)	50.874	18.94	58.43	160.20	3650
IDE	50.873	18.97	58.43	160.22	2900

As observed in earlier test cases, all the algorithms with the strategy of dynamically updating the population size observes a reduction in the number of objective function evaluations (N_{eval}) in relation to the canonical DE. Besides, the performance obtained by IDE (42%) is better. Figure 7 presents the evolution of the best values found for the objective function by using the IDE algorithm.

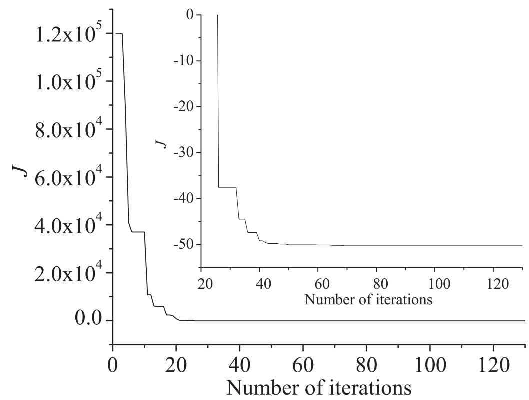

Figure 7. Objective function (J) versus number of iterations.

Figures 8 to 11 show the altitude, velocity, mass and thrust profiles. In these figures, it is easy to locate the events and the behavior of the control variable in a singular arc. This singular arc, depicted the in in Figure 11, corresponds to thrust slightly larger than the drag plus weight, consequently, the rocket is accelerating upward but not wasting fuel to overcome the larger drag it would have encountered if the maximum thrust had been used.

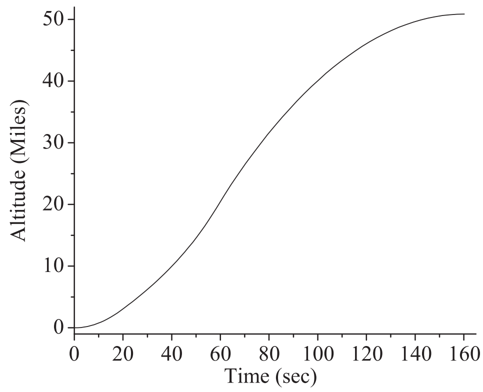

Figure 8. Altitude profile (x_1).

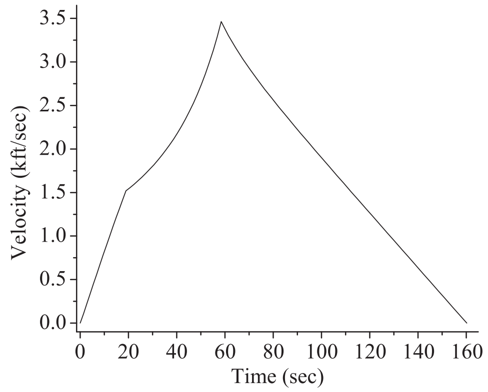

Figure 9. Velocity profile (x_2).

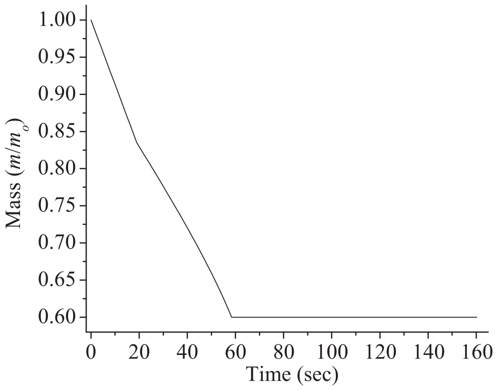

Figure 10. Mass (x_3) profile.

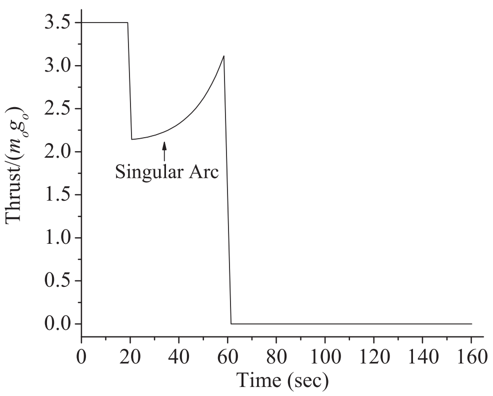

Figure 11. Thrust profile (u).

7 Conclusions

In this paper a new strategy for the dynamic updating of the population size using the Differential Evolution algorithm was presented for dealing with optimization problems. This methodology permits the reduction of the population size during the evolutionary process and, as a consequence, a reduction of the number of objective function evaluations. It was shown that the proposed methodology can be easily incorporated to the Differential Evolution algorithm. The methodology was used to solve singular optimal control problems with different levels of complexity, where the original continuous control trajectory is approximated by linear functions on time intervals. The results showed that the proposed algorithm represents an interesting alternative for the treatment of optimization problems, once the same solution quality achieved by other techniques can be obtained by using a smaller number of generations. Further works will be dedicated to approaches related to dynamically updating the parameters and mutation strategies of the Differential Evolution Algorithm, as presented in [39, 40].

Acknowledgment

Dr. Lobato acknowledges the financial support provided by FAPEMIG, Fundação de Amparo à Pesquisa de Minas Gerais. Dr. Steffen is thankful to the financial support provided by the agencies FAPEMIG and CNPq (INCT-EIE). Dr. Silva Neto acknowledges the financial support provided by the research agencies FAPERJ, Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro and CNPq, Conselho Nacional de Desenvolvimento Cientfico e Tecnológico.

References

[1] A. E. Bryson and Y. C. Ho. *Applied Optimal Control.* Hemisphere Publishing, Washington, 1975.

[2] W. F. Feehery. *Dynamic Optimization with Path Constraints.* PhD thesis, MIT, 1998.

[3] F. S. Lobato, *Hybrid Approach for Dynamic Optimization Problems,* M.Sc. Thesis, FEQUI/UFU, Uberlândia, Brazil, 2004 (in portuguese).

[4] O. von Stryk and R. Bulirsch, *Direct and Indirect Methods for Trajectory Optimization.* Annals

of Operations Research, Vol. 37, pages 357–373, 1992.

[5] O. von Stryk, *User's Guide for DIRCOL - A Direct Collocation Method for the Numerical Solution of Optimal Control Problems.* Technische Universitat Darmastadt, Fachgebiet Simulation and Systemoptimierung (SIM), 1999.

[6] L. T. Biegler, A. M. Cervantes and A. Wachter, *Advances in Simultaneous Strategies for Dynamic Process Optimization.* Chemical Engineering Science, Vol. 57, pages 575–593, 2002.

[7] R. Storn and K. V. Price, *Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces.* International Computer Science Institute, Vol. 12, pages 1–16, 1995.

[8] M. D. Kapadi and R. D. Gudi, *Optimal Control of Fed-batch Fermentation involving Multiple Feeds using Differential Evolution,* Process Biochemistry, Vol. 39, pages 1709–1721.

[9] F. S. Lobato, L. C. Oliveira-Lopes, V. V. Murata and V. Steffen Jr., *Solution of Multi-objective Optimal Control Problems with Index Fluctuation using Differential Evolution.* 6th Brazilian Conference on Dynamics, Control and Applications - DINCON, 2007.

[10] F. Wang and J. Chiou, *Optimal Control and Optimal Time Location Problems of Differential-Algebraic Systems by Differential Evolution,* Ind. Eng. Chem. Res., Vol. 36, pages 5348–5357, 1997.

[11] A. Chowdhury, A. Ghosh, S. Das and A. Abraham, *A Hybrid Evolutionary Direct Search Technique for Solving Optimal Control Problems,* 10th International Conference on Hybrid Intelligent Systems (HIS 2010), Atlanta, USA, 2010.

[12] S. Vellev, *An Adaptive Genetic Algorithm with Dynamic Population Size for Optimizing Join Queries, In International Conference: Intelligent Information and Engineering Systems,* INFOS 2008, Varna, Bulgaria, June-July, 2008.

[13] E. Balsa-Canto, A. A. Alonso and V. S. Vassiliadis, *Efficient Optimal Control of Bioprocess using Second-order Information,* Ind. Eng. Chem. Res., Vol. 39, pages 4287–4295, 2000.

[14] L. T. Biegler and V. M. Zavala, *Large-scale Nonlinear Programming using IPOPT: An Integrating Framework for Enterprise-Wide Dynamic Optimization,* Computers and Chemical Engineering, Vol. 33, pages 575–582, 2009.

[15] R. M. Storn, K. V. Price and J. A. Lampinen, *Differential Evolution - A Practical Approach to Global Optimization.* Springer - Natural Computing Series, 2005.

[16] S. Das and P. N. Suganthan, *Differential Evolution: A Survey of the State-of-the-Art,* IEEE Trans. on Evolutionary Computation, DOI: 10.1109/TEVC.2010.2059031, 2011.

[17] R. Storn, *Differential Evolution Design of an IIR-filter with Requirements for Magnitude and Group Delay,* International Computer Science Institute, TR-95-026, 1995.

[18] B. V. Babu and R. P. Singh. *Synthesis and Optimization of Heat Integrated Distillation Systems using Differential Evolution, in Proceedings of All-India seminar on Chemical Engineering Progress on Resource Development.*

[19] F. S. Lobato and V. Steffen Jr., *Engineering System Design with Multi-objective Differential Evolution*, 19 International Congress of Mechanical Engineering - COBEM 2007.

[20] F. S. Lobato, E. B. Arruda, M. A. S. Barrozo and V. Steffen Jr., *Estimation of Drying Parameters in Rotary Dryers using Differential Evolution. Journal of Physics.* Conference Series, Vol. 135, pages 1–8, 2008.

[21] F. S. Lobato, V. Steffen Jr. and A. J. Silva Neto. *Solution of Inverse Radiative Transfer Problems in Two-layer Participating Media with Differential Evolution,* EngOpt 2008 - International Conference on Engineering Optimization, Rio de Janeiro - Brasil, 2008.

[22] V. C. Mariani, A. G. B. Lima and L. S. Coelho, *Apparent Thermal Diffusivity Estimation of the Banana During Drying using Inverse Method,* Journal of Food Engineering, Vol. 85, pp. 569–579, 2008.

[23] F. S. Lobato, C. E. Figueira, R. R. Soares and V. Steffen Jr., *A Comparative Study of Gibbs Free Energy Minimization in a Real System Using Heuristic Methods.* Computer-Aided Chemical Engineering, Vol .27, pages 1059–1064, 2009.

[24] F. S. Lobato, V. Steffen Jr. and A. J. Silva Neto, *Estimation of Space-dependent Single Scattering Albedo in Radiative Transfer Problems,* Inverse Problems, Design and Optimization Symposium, João Pessoa, Brazil, August 25–27, 2010.

[25] F. S. Lobato, V. Steffen Jr. and A. J. Silva Neto, *A Comparative Study of the Application of Differential Evolution and Simulated Annealing in Inverse Radiative Transfer Problems.* Journal of the Brazilian Society of Mechanical Sciences and Engineering, Vol. XXXII, pages 518–526, 2010.

[26] F. S. Lobato, V. Steffen Jr. and A. J. Silva Neto, *Self-Adaptive Differential Evolution Based on the Concept of Population Diversity Applied to Simultaneous Estimation of Anisotropic Scattering Phase Function, Albedo and Optical Thickness.* Computer Modeling in Engineering & Sciences, Vol. 1, pages 1–17, 2010.

[27] A. Biswas, S. Das, A. Abraham and S. Dasgupta, *Design of Fractional order PID Controllers with an Improved Differential Evolution,* Engineering Applications of Artificial Intelligence, Elsevier Science, Vol. 22, No. 2, pages 343–350, 2009.

[28] S. Y. Sun, Yc. Quiang, Y. Liang, Y. Liu and Q. Pan, *Dynamic Population Size Based on Particle Swarm Optimization,* ISICA 2007, Spring-Verlag Berlin, pages 382–392, 2007.

[29] D. J. Gunn and W. J. Thomas, *Mass Transport and Chemical Reaction in Multifunctional Catalyst Systems,* Chemical Engineering Science, vol. 20, pages 89–100, 1965.

[30] J. S. Logsdon, *Efficient Determination of Optimal Control Profiles for Differential Algebraic Systems,* Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, 1990.

[31] V. Vassiliadis, *Computational Solution of Dynamic Optimization Problems with General Differential-Algebraic Constraints,* Ph.D. thesis, University of London, London, UK, 1993.

[32] F. S. Lobato and V. Steffen Jr., *Solution of Optimal Control Problems using Multi-Particle Collision algoritm,* 9th Conference on Dynamics, Control and Their Applications, June 07–11, 2010.

[33] M. L. Bell and R. W. H. Sargent, *Optimal Control of Inequality Constrained DAE Systems, Computers and Chemical Engineering,* Vol. 24, pages 2385–2404, 2000.

[34] O. Bilous and N. R. Amundson, *Optimum Temperature Gradients in Tubular Reactors II: Numerical Study,* Chemical Engineering Science, Vol. 5, pages 115–126, 1956.

[35] G. Marroquin and W. L. Luyben, *Practical Control Studies of Batch Reactors using Realistic Mathematical Models,* Chemical Engineering Science, Vol. 28, pages 993–1003, 1973.

[36] R. Luus and O. N. Okongwu, *Towards Practical Optimal Control of Batch Reactors,* Chemical Engineering Journal, Vol. 75, pages 1–9, 1999.

[37] H. S. Tsien and R. C. Evans, *Optimum Thrust Programming for a Sounding Rocket,* American Rocket Society Journal, Vol. 21, pages 99–107, 1951.

[38] B. Garfinkel, *A Solution of the Goddard Problem,* SIAM Journal, Vol. 1, pages 349–368, 1963.

[39] A. K. Qin, V. L. Huang, and P. N. Suganthan, *Differential Evolution Algorithm with Strategy Adaptation for Global Numerical Optimization,* IEEE Trans. on Evolutionary Computations, DOI: 10.1109/TEVC.2008.927706, pages 398–417, 2009.

[40] R. Mallipeddi, P. N. Suganthan, Q. K. Pan and M. F. Tasgetiren, *Differential Evolution Algorithm with Ensemble of Parameters and Mutation Strategies,* Applied Soft Computing, DOI:10.1016/j.asoc.2010.04.024, 2011 (Accepted).