Boundary Constraints Strategies in Differential Evolution Algorithms Applied to Optimal Control Problems

Lopez-Cruz I. L., Rojano-Aguilar A. and Salazar-Moreno R.

Postgrado en Ingeniería Agrícola y Uso Integral del Agua
Universidad Autónoma Chapingo, Chapingo, México.
ilopez@chapingo.mx

Abstract. The aim of current research is the evaluation of four strategies: clipping technique, random reinitialization, bounce-back and averaged bounce-back, to handle boundary constraints of the control inputs in solving optimal control problems using Differential Evolution (DE) algorithms are efficient in solving both multimodal, and also singular optimal control problems especially when a relatively greater number of variables have to be optimized. DE algorithms are simple and efficient evolutionary methods when are compared to other evolutionary methods, regarding the number of function evaluations to converge to a solution. Results showed that clipping and bounce-back strategies performed better on a high-multimodal and also a singular optimal control problem, than random reinitialization and averaged bounce-back strategies.

1 Introduction

During the last decade interest on the application of global optimization methods in optimal control problems has significantly increased. Evolutionary Algorithms are stochastic optimization methods that have shown several advantages as global optimization methods. They have been applied mostly to solve static optimization problems and only rarely in solving optimal control problems. It is well known that optimal control problems with singular arcs are very hard to solve by the Pontryagin minimum principle [1],[2]. Singular optimal control problems are frequently found in the optimization of bioreactors [3], [4] and likely also in other biosystems [5]. Also multimodal optimal control problems are frequently found in optimization of bioreactors [6]. Luus [6,7] has applied Iterative Dynamic Programming (IDP), which can be considered as another global optimization method, to solve multimodal and also singular control problems. Tholudur and Ramirez [8], who also used IDP, found highly oscillatory behavior of optimal control trajectories in singular optimal control problems. Therefore, they proposed two filters in order to calculate smoother optimal trajectories. Recently, Roubos *et al.* [5] suggested two smoother evolutionary operators for a Genetic Algorithm with floating-point representation of the individuals and applied this approach to calculate solutions for two fed-batch bioreactors.

Theoretical and empirical results [9] have shown that Evolutionary Algorithms (like those based in Genetic Algorithms) that use low mutation rates for mutation and high probability for crossover are not good candidates to solve optimal control problems efficiently since they may require highly number of function evaluations when

© G. Sidorov, M. Aldape, M. Martínez, S. Torres. (Eds.)
Advances in Computer Science and Engineering.
Research in Computing Science 42, 2009, pp. 359-370

Received 15/03/09
Accepted 27/04/09
Final version 06/05/09

many variables are optimized or these variables are correlated. There is a necessity of developing more efficient global optimization algorithms for solving optimal control problems, in general, and multimodal and singular optimal control problems, in particular. Lately, a new family of evolutionary algorithms named Differential Evolution (DE) has been proposed [10, 11, 12, 13] which is not only simple but also remarkably efficient compared to other Evolutionary Algorithms, Simulated Annealing and Stochastic Differential equations [10, 12]. While Differential Evolution algorithms are studied mainly on unconstrained optimization problems, the present work evaluates four strategies for boundary constraints handling: clipping technique, random reinitialization, bounce-back and averaged bounce-back on the standard differential evolution algorithm (*DE/rand/bin/1*) in solving an multimodal an also and singular arc optimal control problems. As has been shown recently [14] a so-called median filter operator considerably improved solution of singular optimal control problems by using evolutionary algorithms. Therefore, a standard differential evolution algorithm with a median filter operator was also used to evaluated the four strategies to handle boundary constraints in solving the singular arc optimal control problem.

2 The Optimal Control Problem

A continuous-time optimal control problem [15] implies to find an optimal control $u^*(t)$ which causes the system

$$\dot{x} = f(x((t),u(t),t)\,,\ x(t_0) = x_0\,. \tag{1}$$

to follow an admisible trajectory $x^*(t)$ that optimizes the performance measure given by the functional :

$$J = \phi(x(t_f),t_f) + \int_0^{tf} L(x(t),u(t),t)dt \tag{2}$$

where $x \in R^n$ denotes the states of the system and $u \in R^m$ denotes a control vector. In addition the controls are constrained $\alpha \le u(t) \le \beta$. The final time t_f is fixed. As the Hamiltonian function:

$$H(t) = \lambda^T(t) f(x(t),u(t),t) \tag{3}$$

is linear with respect to the controls, the optimal control problem becomes singular [16]. Singular optimal control problems are difficult to solve by classical methods and direct methods seem to be a promising approach. To apply a direct optimization method a parameterization of the controls is necessary, for instance piecewise constant control can be applied

$$u(t) = u(t_k)\,,\ t \in [t_k, t_{k+1})\,,\ k = 0,1,...N-1\,. \tag{4}$$

where N is the number of sub-intervals for the time interval $[t_0, t_f]$. In this way a vector of parameters $\tilde{u} = [u_1^T, u_2^T, ..., u_{N-1}^T]$ is defined and the value that optimizes the original performance index (2) can be obtained by parameter optimization methods or solving a Non-Linear Programming (NLP) optimization problem. The numerical solution of these problems is challenging due to the non-linear and discontinuous dynamics. Likely, there is not a unique global solution. Standard gradient-based algorithms are basically local search methods, they will converge to a local solution. In order to surmount these difficulties global optimization methods must be used in order to ensure proper convergence to the global optimum.

3 Differential Evolution Algorithms

A differential evolution algorithm is as follows:

```
Generate a population ( P(0) ) of solutions.
Evaluate each solution.
g=1;
while (convergence is not reached)
      for i=1 to μ
       Apply differential mutation.
       Execute differential crossover.
       Use a handling boundary constraints technique if
       necessary.
       Evaluate the new solution.
       Apply differential selection.
      end
      g=g+1;
end
```

Firstly, a population $P(0)$ of floating-point vectors $\vec{u}_i, i = 1,...,\mu$ is generated randomly from the domain of the variables to be optimized, where $\vec{u} = [u_1,...,u_d]$ and μ denotes the population size. Next, each vector is evaluated by calculating its associated cost function (eqn. 2), $i = 1,...,\mu$. Notice that the evaluation of each solution implies to carry out a numerical integration of the dynamic model (1). After that, a loop begins in which the evolutionary operators: differential mutation, differential crossover and selection are applied to the population ($P(g)$), where g denotes a generation number. Differential Evolution operators are quite different than those frequently found in other evolutionary algorithms. In DE, the differential mutation operator consists of the generation of μ mutated vectors according to the equation:

$$\vec{v}_i = \vec{u}_{r_1} + F \cdot (\vec{u}_{r_2} - \vec{u}_{r_3}), \; i = 1,2,...,\mu \tag{5}$$

where the random indices $r_1, r_2, r_3 \in [1,2,...,\mu]$ are mutually different and also different from the index i. $F \in [0,2]$ is a real constant parameter that affects the differential

variation between two vectors. Greater values of F and/or the population size (μ) tend to increase the global search capabilities of the algorithm because more areas of the search space are explored.

The crossover operator combines the previously mutated vector $\vec{v}_i = [v_{1i}, v_{2i}, ..., v_{di}]$ with a so-called target vector (a parent solution from the old population) $\vec{u}_i = [u_{1i}, u_{2i}, ..., u_{di}]$ to generate a so-called trial vector $\vec{u}'_i = [u'_{1i}, u'_{2i}, ..., u'_{di}]$ according to:

$$u'_{ji} = \begin{cases} v_{ji} & if \ (randb(j) \le CR) \ \ or \ \ j = rnbr(i) \\ u_{ji} & if \ (randb(j) > CR) \ \ and \ \ j \ne rnbr(i) \end{cases}, \ j = 1,2,...,d; \ i = 1,2,...,\mu \tag{6}$$

where $randb(j) \in [0,1]$ is the j-th evaluation of a uniform random number generator, $rnbr(i) \in 1,2,...,d$ is a randomly chosen index. $CR \in [0,1]$ is the crossover constant, a parameter that increases the diversity of the individuals in the population. Greater values of CR give rise to a child vector ($\vec{u}'_i$) more similar to the mutated vector ($\vec{v}_i$). Therefore, the speed of convergence of the algorithm is increased. As can be seen from equation (6), each member of the population plays once the role of a target vector. It is important to realize that even when $CR = 0$, equation (6) ensures that parent and child vectors differ by at least one gene (variable). The three algorithm parameters that steer the search of the algorithm, the population size (μ), the crossover constant (CR) and differential variation factor (F) remain constant during an optimization.

The selection operator compares the cost function value of the target vector $\vec{u}_i$ with that of the associated trial vector $\vec{u}'_i$, $i = 1,2,...,\mu$ and the best vector of these two becomes a member of the population for the next generation. That is,

$$\text{if } \phi(\vec{u}'_i(g)) < \phi(\vec{u}_i(g)) \text{ then } \vec{u}_i(g+1) := \vec{u}'_i(g)$$
$$\text{else } \vec{u}_i(g+1) := \vec{u}_i(g); \ i = 1,...,\mu$$

Several DE algorithms can be identified according to their type of mutation (x), number of difference vectors (y) and type of crossover (z). Commonly, the notation $DE/x/y/z$ is used to named a DE algorithm. Where x, means the way the vector to be mutated is chosen, y indicates the number of difference vectors is used, and z is the type of differential crossover implemented. For instance, the previously described algorithm is known as the $DE/rand/1/bin$ which means than the to be mutated vector is selected randomly, only one difference vector is calculated and the scheme of crossover is binomial. In general $x \in \{rand, best, current-to-rand\}$, $y \in \{1,2,...,n\}$, and $z \in \{bin, \exp\}$.

Statistics to Measure Differential Evolution Performance

To measure algorithm's convergence in addition to the mean and standard deviation from several runs the Q-measure (Q_m)[13] was used, but also the expected number of functions evaluations per success (ENES) [12] and the average (number of function) evaluations per success (AES)[12] from a number of consecutive successful trials

were calculated. The Q-measure[13] is given by the ratio of a convergence measure (C) and a probability of convergence (P_C) according to equation (7):

$$Q_m = \frac{C}{P_C}. \tag{7}$$

Where the convergence measure and AES are calculated by equation (8)

$$AES = C = \frac{\sum_{j=1}^{n_s} E_j}{n_s}. \tag{8}$$

Where E_j is the number of function evaluations in the ith trial and n_s is the number of successful trials. A probability of convergence is calculated by the ration between the number of successful trial and the total number of trial given by equation (9)

$$P_C = \frac{n_s}{n_t}\% \tag{9}$$

On the other hand ENES is calculated as follows:

$$ENES = \frac{\sum_{i=1}^{n_t} E_i}{n_s}. \tag{10}$$

Boundary Constraints Handling Strategies for Optimal Control Problems

Since originally DE algorithms were designed to solve unconstrained static optimization problems, a modification is required in order to deal with constraints for the controls. A **clipping technique** has been introduced to guarantee that only feasible trial vectors are generated after the mutation and crossover operators:

$$u'_{ji}(g) = \begin{cases} \beta_j & if \ u'_{ji}(g) > \beta_j \\ \alpha_j & if \ u'_{ji}(g) < \alpha_j \end{cases} \quad j = 1,2,...,d; i = 1,2,...,\mu. \tag{11}$$

where α_j and β_j represent the lower and upper boundaries of the control variables, respectively. This approach has been successfully applied to optimal control problems [14], [17]. **Random reinitialization** [12] is described by equation (8) as follows:

$$u'_{ji}(g) = \alpha_j + rand_j(0,1)(\beta_j - \alpha_j) \ if \ u'_{ji}(g) < \alpha_j \ or \ u'_{ji}(g) > \beta_j \tag{12}$$

The **bounce-back** technique has been proposed recently [12] as is described in equation (9):

$$u'_{ji}(g) = \begin{cases} \overline{u}_{r_1} + rand(0,1)(\alpha_j - \overline{u}_{r_1}) \ if \ u'_{ji}(g) < \alpha_j \\ \overline{u}_{r_1} + rand(0,1)(\beta_j - \overline{u}_{r_1}) \ if \ u'_{ji}(g) > \beta_j \end{cases} \tag{13}$$

The averaged bounce-back strategy was suggested in [11] and [12] according to the equation (10):

$$u'_{ji}(g) = \begin{cases} (\alpha_j + \overline{u}_{r_1})/2 \ if \ u'_{ji}(g) < \alpha_j \\ (\beta_j + \overline{u}_{r_1})/2 \ if \ u'_{ji}(g) > \beta_j \end{cases} \tag{14}$$

The Smoother Operator to Singular Optimal Control Problems

A smoother operator is defined according to [8] as follows:

$$u_{j,i} = median(u_{j-F,i}, u_{j-F+1,i}, ..., u_{j,i}, ..., u_{j+F-1,i}, u_{j+F,i}) \qquad (15)$$
$$j = F+1, F+2, ..., N-F \; ; \; i = 1,2,...,\mu$$

where $F = 1,2,..$ is the filtering radius. All the boundary constraints handling strategies and the standard Differential Evolution algorithm were programmed as an m-file in the Matlab environment.

4 Multimodal Optimal Control of Bifunctional Catalyst Blend

A chemical process converting methylcyclopentane to benzene in a tubular reactor is modelled by a set of seven differential equations:

$$\dot{x}_1 = -k_1 x_1 \; . \qquad (16)$$

$$\dot{x}_2 = k_1 x_1 - (k_2 + k_3) x_2 + k_4 x_5 \; . \qquad (17)$$

$$\dot{x}_3 = k_2 x_2 \; . \qquad (18)$$

$$\dot{x}_4 = -k_6 x_4 + k_5 x_5 \qquad (19)$$

$$\dot{x}_5 = k_3 x_2 + k_6 x_4 - (k_4 + k_5 + k_8 + k_9) x_5 + k_7 x_6 + k_{10} x_7 \; . \qquad (20)$$

$$\dot{x}_6 = k_8 x_5 - k_7 x_6 \; . \qquad (21)$$

$$\dot{x}_7 = k_9 x_5 - k_{10} x_7 \; . \qquad (22)$$

where $x_i, i = 1,...,7$ are the mole fractions of the chemical species, and the rate constants (k_i) are cubic functions of the catalyst blend $u(t)$:

$$k_i = c_{i1} + c_{i2} u + c_{i3} u^2 + c_{i4} u^3 \; , \; i = 1,...,10 \qquad (23)$$

The values of the coefficients c_{ij} are given in [7]. The upper and lower bounds on the mass fraction of the hydrogenation catalyst are: $0.6 \le u(t) \le 0.9$, and the initial vector of mole fraction is $\mathbf{x}[0] = [1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]^T$. This is a continuous process operated in steady state, so that 'time' in equations (16)-(23) is equivalent to travel time and thus length along the reactor. The optimal control problem is to find the catalyst blend along the length of the reactor, which in the control problem formulation is considered at times $0 \le t \le t_f$ where the final effective residence time $t_f = 2000 g \cdot h / mol$ such that the concentration in the reactor is maximised:

$J = x_7(t_f)\times 10^3$. Esposito and Floudas [18] found recently 300 local minima of this problem, so this is a challenging multimodal optimal control problem as the number of intervals N=10 was used. A variable step size four order Runge-Kutta integration method with a relative tolerance of 10^{-8} was applied to solve the dynamic equations (16-22).

5 Singular optimal control of the Park-Ramirez bioreactor

One optimal control problem that has a singular optimal solution was used to test the modified DE algorithm [8]. In this problem the goal is to maximize the production of protein. The system is described by the following differential equations:

$$\dot{x}_1 = g_1(x_2 - x_1) - \frac{x_1}{x_5}u. \tag{24}$$

$$\dot{x}_2 = g_2 x_3 - \frac{x_2}{x_5}u. \tag{25}$$

$$\dot{x}_3 = g_3 x_3 - \frac{x_3}{x_5}u. \tag{26}$$

$$\dot{x}_4 = -g_4 g_3 x_3 - \frac{m - x_4}{x_5}u. \tag{27}$$

$$\dot{x}_5 = u. \tag{28}$$

where $g_1 = \dfrac{4.75 g_3}{0.12 + g_3}$, $g_2 = \dfrac{21.88 x_4}{(x_4 + 0.4)(x_4 + 62.5)}$, $g_3 = \dfrac{x_4 \exp(-5.01 x_4)}{0.10 + x_4}$,

$g_4 = 58.75 g_2^2 + 1.71$.

The state variable x_1 represents amount of secreted protein [unit culture volume L^{-1}], x_2 denotes the total protein amount [unit culture volume L^{-1}], x_3 means culture cell density [g L^{-1}], x_4 culture glucose concentration [g L^{-1}], and x_5 the culture volume [L]. The control $u(t)$ represents the rate at which glucose is fed into the reactor [Lh^{-1}]. The secretion rate constant is given by g_1, the protein expression rate is calculated by g_2, the specific growth rate by g_3 and the biomass to glucose yield is estimated by g_4. The optimal control problem consists in the maximization of the amount of the secreted protein in a given time $t_f = 15h$. Therefore the performance index is given by $J = x_1(t_f)x_5(t_f)$. The control input satisfying the constraints $0 \le u(t) \le 2.5$ and the system initial conditions are $x(0) = [0,0,1.0,5.0,1.0]$. The dynamic model (eqns. 24-28) was programmed in the Matlab-Simulink environment. A C-MEX file containing the dynamic equations was implemented in order to speed up

the simulations. A variable step size four-order Runge-Kutta integration method with a relative tolerance of 10^{-8} was applied. The DE algorithm was initialized randomly from the control's domain. Since DE algorithms are probabilistic methods the optimizations were repeated 30 times. The problem was solved for N=100 variables.

6 Results and discussion

Multimodal Optimal Control Problem

The parameters for the standard DE algorithm were: NP=25, F=0.9, CR=0.0. Instead of using as a convergence criteria the difference between worst and best solution the value to-reach (VTR) condition as suggested in [12] was used. The value to-reach (VTR) was defined as 10.09415 since it is known that the global optimum for this problem is 10.0942. The maximum number of function evaluations ($E_{\max}$) was defined as 15000. The total number of trials was thirty. Table 1 sumarizes the observed behaviour of the four boundary constraints handling strategies taking into account the number of functions evaluations. It is apparent that the clipping method performed much better since it has the lesser Q_m measure, and also the lesser AES and ENES than the others strategies. Random reinitialization was the worst strategy. Strategies that used bounce-back were a little higher than clipping technique but considerably more efficient than random reinitialization.

Table 1. Statistics calculated to evaluate the four boundary constraints-handling strategies in the DE/rand/1/bin algorihtm on a multimodal optimal control problem.

	Clipping technique	Random reinitialization	Bounce-back	Averaged bounce-back
Mean	3059.2	9561.6	3893.3	3348.3
Std	346.9	407.4	157.0	236.6
ENES	3059.2	9924.2	3893.3	3865.5
AES	3059.2	9561.6	3893.3	3348.3
C	3059.2	9561.6	3893.3	3348.3
P_C	100%	93.3	100%	96.6
Q_m	30.5	102.4	39.8	34.6

These results are according to remarks made by Price et al. [12]. An statistical test of significance among the four strategies was carried out. The null hypothesis: means are equal, was rejected in all cases at the significance level 0.05. Figure 1 shows the optimal control trajectory found by DE algorithm using the four handling constraints strategies. Almost no differences among them can be detected. The smoother operator was not applied to this problem. Numerical values for the best solution found by the algorithm using each constraints-handling strategy are showed in Table 2.

Table 2. Numerical values corresponding to the optimal controls calculated by DE/rand/1/bin algorithm using four boundary contraints-handling strategies.

Control	Clipping technique	Random reinitialization	Bounced back	Averaged bounce-back
1	0.6656	0.6661	0.6662	0.6661
2	0.6733	0.6735	0.6735	0.6736
3	0.6764	0.6763	0.6763	0.6762
4	0.9000	0.9000	0.9000	0.9000
5	0.9000	0.9000	0.9000	0.9000
6	0.9000	0.9000	0.9000	0.9000
7	0.9000	0.9000	0.9000	0.9000
8	0.9000	0.9000	0.9000	0.9000
9	0.9000	0.9000	0.9000	0.9000
10	0.9000	0.9000	0.9000	0.9000

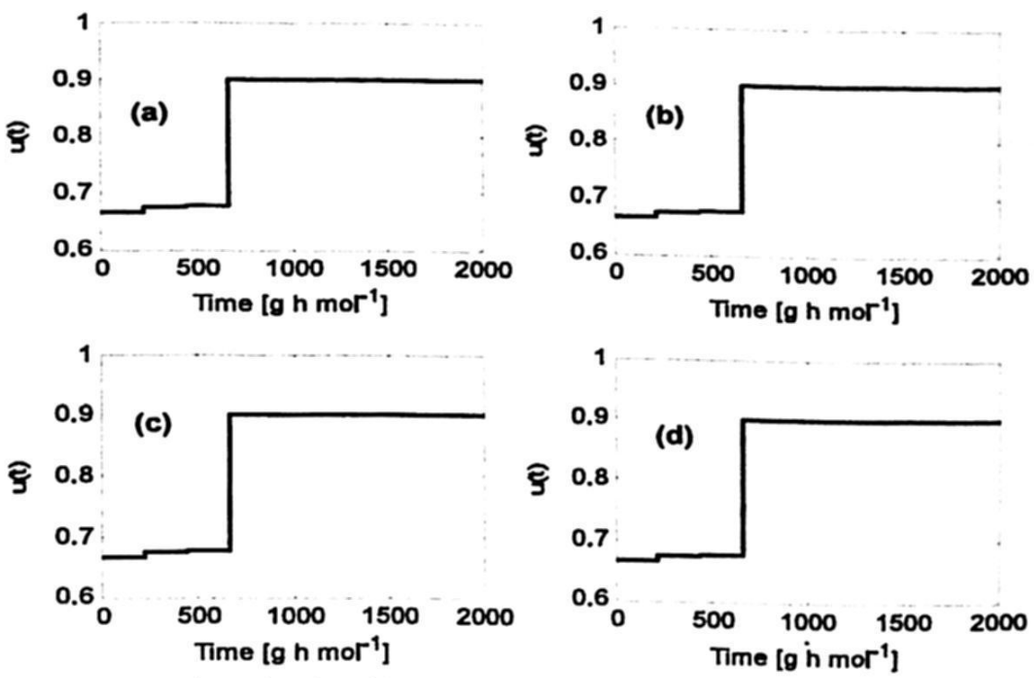

Fig. 1. Optimal trajectories found by the DE/rand/1/bin algorithm for the multimodal problem. a) Clipping technique. b) Bounce-back approach. c) Averaged bounce-back. d) Random reinitialization.

Singular Optimal Control of Park-Ramirez Bioreactor

The parameters for the standard DE algorithm were: population size 100 individuals, scaling factor for mutation F=0.6 and probability of crossover CR=0.9. The total number of trials was thirty. The converge criterion was the VTR=32.47, which has been reported before as the global optimum for this problem [8] using Iterative Dynamic Programming using in case of $N = 100$ parameters. Table 3 shows the performance of the DE/rand/1/bin algorithm using the four strategies regarding the number of function evaluations required to solve this problem.

Table 3. Statistics calculated to evaluate the four boundary constraints-handling strategies in the DE/rand/1/bin algorihtm on a singular optimal control problem.

	Clipping technique	Random reinitialization	Bounce-back	Averaged bounce-back
Mean	64607.0	139233.0	73610.0	69500.0
Std	7584.1	8417.0	6629.9	7399.4
ENES	64607.0	139233.0	73610.0	69500
AES	64607.0	139233.0	73610.0	69500
C	64607.0	139233.0	73610.0	69500
P_C	100%	100%	100%	100%
Q_m	646.0	1392.3	736.1	695.0

As in case of the multimodal problem again the clipping approach was the most efficient since it required lower Q_m-measure ENES and AES than the other strategies. The random reinitialization strategy was the worst method taking into account several statistics. An statistical test of significance among the four strategies was carried out. The null hypothesis: means are equal, was rejected in all cases at the significance level 0.05. Figure 2 shows the optimal control trajectory found by DE algorithm using the four handling constraints strategies. Only small differences in the optimal trajectories due to the effect of the smoother operator are observed.

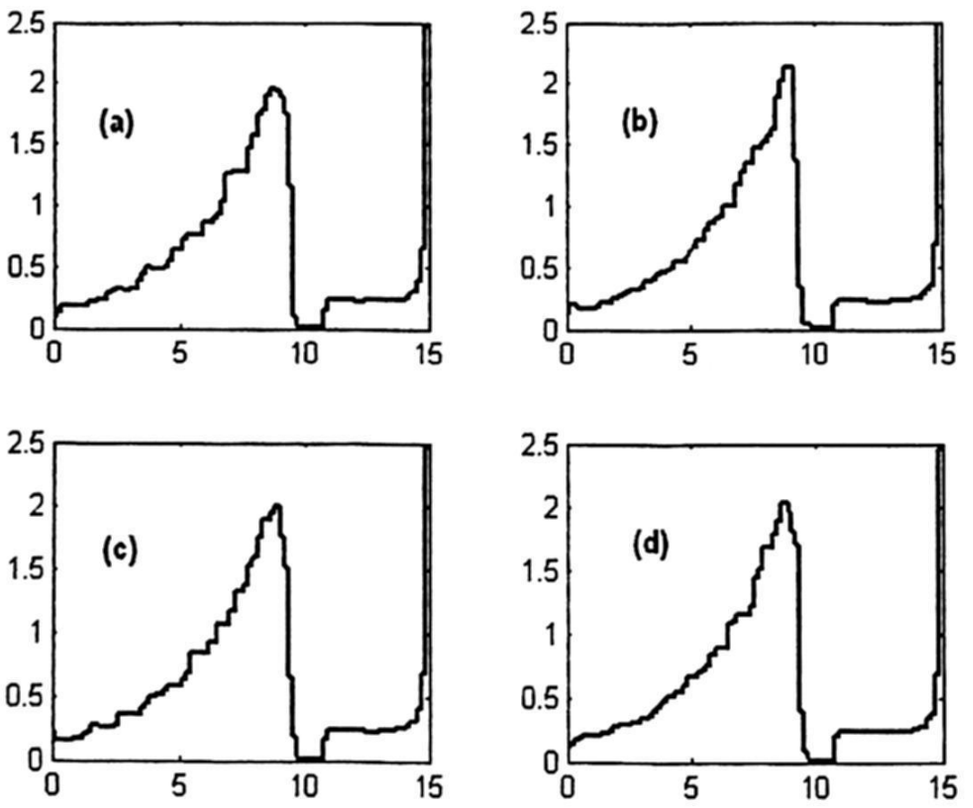

Fig. 2. Optimal trajectories found by the DE/rand/1/bin algorithm for the singular arc control problem. a) Clipping technique. b) Bounce-back approach. c) Averaged bounce-back. d) Random reinitialization.

7 Conclusions

A highly multimodal optimal control problem was used to test four boundary constraints strategies on the performance of the standard differential evolution algorithm (DE/rand/1/bin). Results showed although clipping technique, random reinitialization, bounce-back and averaged bounce back converged to the global optimum, the clipping and bounce-back approaches are more efficient regarding the required number of function evaluations. In addition an singular optimal control problem that appears in the dynamic optimization of bioreactors was solved. Again results showed that the four boundary constraints strategies worked out well, however clipping and bounce-back approaches solved the problem more efficiently.

References

1. Park, S., Ramirez, W. F.: Optimal production of secreted protein in fed-batch reactors. AIChE Journal 34 (9) (1988) 1550-1558.
2. Park, S., Ramirez, W. F.: Dynamics of foreign protein secretion from *Saccharomyces cerevisiae*. Biotechnology and Bioengineering 33 (1989) 272-281.
3. Menawat, A., Mutharasan, R., Coughanowr, D. R.: Singular optimal control strategy for a fed-batch bioreactor: numerical approach. AIChE Journal 33 (5)(1987) 776-783.
4. Roubus, J.A., de Gooijer, C.D., van Straten, G., van Boxtel, A.J.B.: Comparison of optimization methods for fed-batch cultures of hybridoma cells, Bioproc. Eng. 17 (1997) 99-102.
5. Roubos, J.A., van Straten, G., van Boxtel, A.J.B.. An evolutionary strategy for fed-batch bioreactor optimization: concepts and performance, Journal of Biotechnology 67 (1999) 173-187.
6. Luus, R.. On the application of Iterative Dynamic Programming to Singular Optimal Control problems. IEEE, Transactions on Automatic Control 37 (11) (1992)
7. Luus, R. Iterative Dynamic Programming. Chapman & Hall/CRC., Boca Raton, (2000).
8. Tholudur, A., Ramirez, W.F.: Obtaining smoother singular arc policies using a modified iterative dynamic programming algorithm. International Journal of Control, 68(5) (1997)1115-1128.
9. Salomon, R.: Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms. BioSystems 39 (1996) 263-278.
10. Storn, R., Price, K.: Differential Evolution- a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11 (1997) 341-359.
11. Price K., V.: An Introduction to Differential Evolution. In Corne, D., Dorigo, M., and Glover, F. (eds.). New Ideas in Optimization. Mc Graw Hill (1999).
12. Price K. V., Storn R.M., Lampinen J.A.: Differential Evolution, A practical approach to global optimization. Springer-Verlag. Berlin (2005).

13. Feoktistov V.: Differential Evolution, In search of solutions. Springer-Verlag. New York (2006).
14. Lopez-Cruz I., Rojano-Aguilar A.: Differential Evolution algorithms to solve optimal control problems efficiently. Research on Computing Science 16 (2005): 271-280.
15. Kirk, D.E.: Optimal control theory. An introduction. Dover Publications, Inc. New York. (1998).
16. Bryson, A.E.: Dynamic Optimization. Addison-Wesley. (1999).
17. Lopez-Cruz, I., van Willigenburg, G., van Straten, G. : Differential evolution algorithms for multimodal optimal control problems. Applied Soft Computing 3 (2) (2004) 97-122.
18. Esposito W.R., Floudas, Ch.A.: Deterministic global optimisation in nonlinear optimal control problems. Journal of global optimisation 17 (2000) 97-126.