Differential Evolution Algorithms to Solve Optimal Control Problems Efficiently

I.L. López-Cruz; A. Rojano-Aguilar
Postgrado en Ingeniería Agrícola y Uso Integral del Agua
Universidad Autónoma Chapingo
Chapingo, México
e-mail:ilopez@correo.chapingo.mx

Abstract. Optimal control of multimodal and singular problems of bioreactors has received considerable attention recently. Three main approaches have been attempted: deterministic methods like Iterative Dynamic Programming, stochastic methods like Adaptive Stochastic algorithms, and Evolutionary Algorithms. The aim of this research is to demonstrate that new evolutionary algorithms called generically Differential Evolution (DE) algorithms are efficient in solving both multimodal, and singular optimal control problems especially when a relatively greater number of variables (50-100) 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. It is shown that besides he three main operators of DE: mutation, crossover and selection, a filter operator is added in order to obtain smoother optimal trajectories of singular optimal control problems.

1 Introduction

During the last decade interest on the application of global optimization methods in optimal control has significantly increased. Evolutionary Algorithms are stochastic optimization methods that have shown several advantages as global optimization methods. They have been applied in the past basically to solve static optimization problems and only rarely to solve multimodal optimal control problems. It is well known that optimal control problems with singular arcs are very hard to solve by using 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 solving 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.

In spite of its reliability as a global optimization method, IDP is rather complex with several algorithm parameters, which require an expensive tuning, before the application of the algorithm to a new problem. Since many experiments are necessary IDP becomes deceptively inefficient recalling that the computation time is critical in solv-

© A. Gelbukh, R. Monroy. (Eds.)
Advances in Artificial Intelligence Theory
Research on Computing Science 16, 2005, pp. 271-280

ing optimal control problems. In dynamic optimization each evaluation of the cost function means running a long simulation (integration) of the dynamic model of the process. 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 many variables are optimized or these variables are correlated. Therefore, 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] which is not only simple but also remarkably efficient compared to other Evolutionary Algorithms, Simulated Annealing and Stochastic Differential equations. Recently, results were have presented that show DE are one of the most efficient evolutionary algorithms to solve optimal control problems efficiently [12, 13]. The present work illustrates that indeed DE algorithms are good candidates to solve multimodal optimal control problems. Also modified DE algorithms are evaluated in solving singular optimal control problems. The new operator is simple and does not add any additional algorithm parameter. The so-called median filter operator basically consists of a sliding window such as each control is replaced with the median of a few neighboring controls. The proposed operator is implemented on the *DE/rand/bin/1* algorithm, and tested on solving a dynamic optimization problem of a fed-batch bioreactor. In this work efficiency of algorithms is measured by counting the number of function evaluations required to solve a problem, which is a machine independent criterion. A comparison of the *DE/rand/bin/1* algorithm performance with and without the smoother operator is presented to illustrate the advantages of the proposed modified Differential Evolution algorithm.

2 The Optimal Control Problem

A continuous-time optimal control problem [12] 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 admissible 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 [13]. 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

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

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.
      Clip the new solution if necessary.
      Evaluate the new solution.
      Apply differential selection.
     end
     g=g+1;
end
```

Firstly, a population $P(0)$ of floating-point vectors $\bar{u}_i, i = 1,...,\mu$ is generated randomly from the domain of the variables to be optimized, where $\bar{u} = [u_1,...,u_d]$ 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:

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

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:

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

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 name 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 $z \in \{bin, \exp\}$.

Extensions of DE and a smoother operator

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:

$$u'_{ji}(g) = \begin{cases} \beta_j & if \;\; u'_{ji}(g) > \beta_j \\ \alpha_j & if \;\; u'_{ji}(g) < \alpha_j \end{cases} ; \;\; j = 1,2,...,d; i = 1,2,...,\mu \qquad \textbf{(7)}$$

where α_j and β_j represent the lower and upper boundaries of the control variables, respectively. A smoother operator is defined according to [8] as follows:

$$u_{j,i} = median(u_{j-F,i}, u_{j-F+1,i},...,u_{j,i},...,u_{j+F-1,i}, u_{j+F,i}) \qquad \textbf{(8)}$$
$$j = F+1, F+2,..., N-F \; ; i = 1,2,..., \mu$$

where $F = 1,2,..$ is the filtering radius. Both Differential Evolution algorithms and its extensions 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 modeled by a set of seven differential equations:

$$\dot{x}_1 = -k_1 x_1 \; . \qquad \textbf{(9)}$$

$$\dot{x}_2 = k_1 x_1 - (k_2 + k_3) x_2 + k_4 x_5 \; . \qquad \textbf{(10)}$$

$$\dot{x}_3 = k_2 x_2 \; . \qquad \textbf{(11)}$$

$$\dot{x}_4 = -k_6 x_4 + k_5 x_5 \; . \qquad \textbf{(12)}$$

$$\dot{x}_5 = k_3 x_2 + k_6 x_4 - (k_4 + k_5 + k_8 + k_9) x_5 + k_7 x_6 + k_{10} x_7 \; . \qquad \textbf{(13)}$$

$$\dot{x}_6 = k_8 x_5 - k_7 x_6 \; . \qquad \textbf{(14)}$$

$$\dot{x}_7 = k_9 x_5 - k_{10} x_7 \; . \qquad \textbf{(15)}$$

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 $u(t)$:

$$k_i = c_{i1} + c_{i2} u + c_{i3} u^2 + c_{i4} u^3 \; ; \; i = 1,...,10 \qquad \textbf{(16)}$$

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 $\mathbf{x}[0] = \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}^T$. This is a continuous process operated in steady state, so that 'time' in equations (9)-(16) 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 maximized: $J = x_7(t_f) \times 10^3$. Esposito and Floudas [16] found recently 300 local minima of this problem, so this is a challenging multimodal optimal control problem.

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:

$$\dot{x}_1 = g_1(x_2 - x_1) - \frac{x_1}{x_5}u \quad . \tag{17}$$

$$\dot{x}_2 = g_2 x_3 - \frac{x_2}{x_5}u \tag{18}$$

$$\dot{x}_3 = g_3 x_3 - \frac{x_3}{x_5}u \quad . \tag{19}$$

$$\dot{x}_4 = -g_4 g_3 x_3 - \frac{m - x_4}{x_5}u \quad . \tag{20}$$

$$\dot{x}_5 = u \quad . \tag{21}$$

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. 10-14) 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 Runge-Kutta integration method with a relative tolerance of $1e-8$ was applied. The DE algorithm was initialized randomly from the control's domain. Since DE algorithms are probabilistic methods the optimizations were repeated 10 times. The problem was solved for two number of variables N=50 and N=100.

6 Results and Discussion

Multimodal optimal control problem
Ten differential evolution algorithms were evaluated in solving the multimodal optimal control problem aforementioned. **Table 1** shows main results. NP is the population size used in each algorithm.

Table 1. Evaluation of several DE algorithms in solving a multimodal continuous-time optimal control problem

DE	CR	F	NP	Kp	F.E.	STD	J*	STD
$DE/best/2/bin$	0.0	0.9	15	-	2529	262.98	10.0942	0.0
$DE/best/2/\exp$	0.0	0.9	20	-	3426	388.85	10.0942	0.0
$DE/rand/1/\exp$	0.0	0.9	15	-	2289	295.31	10.0942	0.0
$DE/rand/1/bin$	0.0	0.9	20	-	3044	351.22	10.0942	4e-5
$DE/rand/2/bin$	0.0	0.9	20	-	3872	332.69	10.0942	4e-5
$DE/rand/2/\exp$	0.0	0.9	20	-	3882	440.44	10.0942	0.0
$DE/curr-to-rand/1/bin$	0.0	0.9	15	1	2100	346.04	10.0942	5e-5
$DE/curr-to-rand/1/\exp$	0.0	0.9	15	1	2257	202.63	10.0942	0.0
$DE/best/1/bin$	0.0	0.9	25	-	3112	324.30	10.0942	0.0
$DE/best/1/\exp$	0.0	1.0	25	-	3245	413.11	10.0941	2e-4

Results of table 1 represent the average of 10 runs regarding number of function evaluations (FE) and the objective function (J^*). A measure of population convergence was defined as a difference between worst and best solution satisfied a given value. In this case the accuracy required was 1e-3. Clearly all DE algorithms found

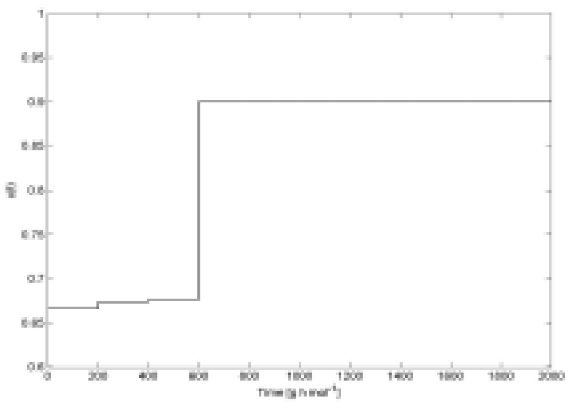

Figure 1. Optimal control trajectory of multimodal probem.

the global optimum with the given values of the parameters. Notice that because of the high multimodality of the problem the mutation parameter is greater and in some cases the population size was increased more than two times the size of the number of variables to be optimized. Price [11], suggests populations sizes between $2n$ and $20n$ but our results show that even with lower sizes DE algorithms can solve multimodal optimal control problems. Figure 1 shows the optimal control trajectory found by DE algorithms.

Singular optimal control of Park-Ramirez bioreactor

Since DE algorithms are very robust it is easy to determine a set of parameters that provides an acceptable solution. Furthermore, the solved optimal control problem has likely only one solution so it was found that an almost standard setting worked out properly. In contrast to the commonly applied approach, which is based on the use of a too large population size, in our situation population size was chosen equal to the dimension of the optimization (N) problem. Since we did not expect a multimodal problem then the mutation constant was kept reasonably small. However, the cross-over parameter was substantially increased in order to speed up the convergence of the algorithms. Table 2 and table 3 show the parameters settings (crossover constant, mutation parameter and population size) of $DE/rand/1/bin$ applied on optimal control problem using the Park-Ramirez bioreactor. Also the main results of the comparison regarding number of generations required, the number of function evaluations needed and the cost function value are presented.

Table 2. Results obtained by DE and smoother DE in solving a singular optimal control problem (Number of variables N=50)

	CR	F	μ	Generations	Function Evaluations	J^*
DE	0.9	0.6	50	5192	259600	32.41
SDE	0.9	0.6	50	932	46600	32.41

Table 3. Results obtained by DE and smoother DE in solving a singular optimal control problem (Number of variables N=100).

	CR	F	μ	Generations	Function Evaluations	J^*
DE	0.9	0.6	100	8251	825100	32.47
SDE	0.9	0.6	100	436	43600	32.47

Figure 2 and figure 3 show the optimal control trajectories calculated by both the DE and the smoother DE algorithms for N=50 and N=100 number of variables. In both cases the trajectories resulted on the same cost function value. Clearly, the trajectory generated by DE algorithm with a smoother operator has less oscillation than that obtained by DE. The oscillation of optimal control trajectory obtained by DE was as the control was parameterized more variables (N=100). A comparison of figures 2 and 3 makes apparent that only small differences can be distinguished between the optimal control trajectories calculated by the smother DE algorithm. The performance index values obtained for both situations N=50 and N=100 were exactly the same reported by [8] using Iterative Dynamic Programming. The improvement in efficiency according to the number of function evaluations as N=50 is used was 6 % in case of N=100

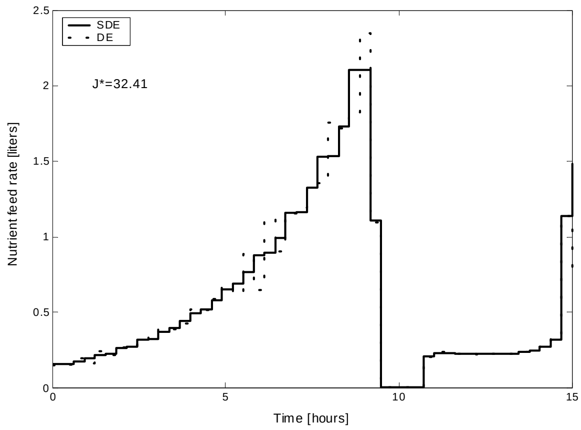

Figure 2. Optimal control trajectory of DE without and with smoother operator (N=50).

this was 19%. Similar percentages were obtained taking into consideration the number of generations. The explanation of this fact could be the increment of population size to $\mu = 100$ individuals as N=100. But also it is clear that avoiding the oscillation of the optimal trajectories speed up the convergence of the DE algorithm. Therefore, it is clear that using the smoother operator together with other DE operators, the performance of DE algorithms is improved considerably and also higher oscillation of optimal control trajectories can be avoided.

7 Conclusions

A highly multimodal optimal control problem was used to test the performance of several differential evolution algorithms. Results show that DE algorithms are good candidates to solve this class of problems since even using small populations they can find the global optimum trajectory. DE algorithms are robust and their parameters are chosen in a straightforward way. A smoother operator was proposed and evaluated in

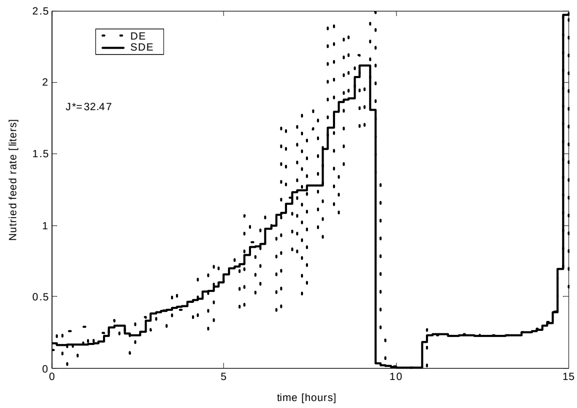

Figure 3. Optimal control trajectory without and with smoother operator N=100.

solving singular optimal control problems by Differential Evolution algorithms. The evaluation of the smoother operator on a dynamic optimization problem of a nonlinear bioreactor showed that the operator not only removed the oscillation of the optimal control trajectory, but also it speed up the convergence of the of a DE algorithm.

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. Lopez-Cruz, I.L. van Willigenburg, G., van Straten G. 2003. Efficient evolutionary algorithms for multimodal optimal control problems. Journal of applied soft computing 3(2): 97-122.
13. Moles, C.G., Banga, J.R., Keller, K. 2004. Solving nonconvex climate control problems: pitfalls and algorithm performances. Journal of applied soft computing 5:35-44.
14. Kirk, D.E.: Optimal control theory. An introduction. Dover Publications, Inc. New York. (1998).
15. Bryson, A.E.: Dynamic Optimization. Addison-Wesley. (1999).
16. Esposito W.R., Floudas, Ch.A.: Deterministic global optimisation in nonlinear optimal control problems. Journal of global optimisation 17 (2000) 97-126.