IJE TRANSACTIONS C: Aspects Vol. 25, No. 3, (September 2012) 197-204

International Journal of Engineering

Journal Homepage: www.ije.ir

Constrained Nonlinear Optimal Control via a Hybrid BA-SD

A. Alfi [a,*], A. Khosravi [b]

[a] *Faculty of Electrical and Robotic Engineering, Shahrood University of Technology, P.O. Box 36199-95161, Shahrood, Iran*
[b] *Department of Electrical and Computer Engineering, Babol University of Technology, P.O. Box 484, Babol, Iran*

PAPER INFO

Paper history:
Received 21 February 2012
Received in revised form 8 June 2012
Accepted 14 June 2012

Keywords:
Constrained Optimization
Swarm Intelligence
Bees Algorithm
Steepest Descent
Differential Evolution
Genetic Algorithm

ABSTRACT

The non-convex behavior presented by nonlinear systems limits the application of classical optimization techniques to solve optimal control problems for these kinds of systems. This paper proposes a hybrid algorithm, namely BA-SD, by combining Bee algorithm (BA) with steepest descent (SD) method for numerically solving nonlinear optimal control (NOC) problems. The proposed algorithm includes the merits of BA and SD simultaneously. The motivation of presenting the proposed algorithm includes that BA is showed to converge to the region that global optimum is settled, rapidly during the initial stages of its search. However, around global optimum, the search process will become slowly. In contrast, SD method has low ability to convergence to local optimum, but it can achieve faster convergent speed around global optimum and the convergent accuracy can be higher. In the proposed algorithm, at the beginning step of search procedure, BA is utilized to find a near optimum solution. In this case, the hybrid algorithm is used to enhance global search ability. When the change in fitness value is smaller than a predefined value, the searching procedure is switched to SD to accelerate the search procedure and find an accurate solution. In this way, the algorithm finds an optimum solution more accurately. Simulations demonstrate the feasibility of the proposed algorithm.

doi: *10.5829/idosi.ije.2012.25.03c.03*

1. INTRODUCTION

Since solving optimal control of nonlinear complex dynamics referred as nonlinear optimal control (NOC) lead to presenting multiple local optimums, global optimal control methods are required to find the global optimum or a sufficiently close approximation. To determine optimal control for a nonlinear system in classic optimal control theory, the nonlinear Hamilton–Jacobi–Bellman (HJB) equations have to be solved [1]. Nevertheless, in practice, the nonlinear HJB equations are very difficult to be solved. Hence, it is necessary to use numerical methods to solve NOC problems.

To handle this problem, some approaches introduced in the literatures numerically such as gradient descents. Although gradient descents such as steepest descent (SD) have shown important advances by providing high precision to solve NOC problems, but there exists possible trapping in local minima depending on the degree of nonlinearity and initial guess of solution. Based on this reason, population based search algorithms such as genetic algorithms (GA) [2]; particle swarm optimization (PSO) [3] and differential evolution (DE) [4] seem to be a more hopeful approach and provide a powerful means to solve this problem. These algorithms have promising advantages in compared with traditional techniques, including 1) They are free of model such that the gradient of objective function is not required; 2) They are not sensitive to initial guess of solution, and 3) They usually do not get stuck into a local optimum. For this reason, they have been successfully applied in many NOC problems [5-13].

Nowadays, bee algorithm (BA) as a new member of meta-heuristic algorithms is employed to solve several optimization problems such as training neural networks for pattern recognition, scheduling jobs for a production machine, and finding multiple feasible solutions to a preliminary design problem [14-18]. BA tries to model natural behavior of honey bees in food foraging. This makes them an excellent candidate for developing novel algorithms for solving optimization problems. Although BA is effective in the initial iterations of the search process, but in the neighborhood of the global optimal point the convergence speed is slow [15].

* Corresponding Author Email: *a_alfi@shahroodut.ac.ir* (A. Alfi)

Motivated by the aforementioned researches, the goal of this paper is to develop a new optimization algorithm to enhance global search ability, faster convergent speed around global optimum and also higher convergent accuracy for solving NOC problem. So, a hybrid algorithm by combining BA with SD, namely BA-SD, is introduced which combines merits of BA and SD methods for solving NOC. It is demonstrated that the proposed hybrid algorithm has the advantage of both BA and SD methods while it does not inherent their shortages. In the proposed hybrid algorithm, at the beginning step of searching process, BA is employed to find a neighborhood of the optimum solution. When the change in fitness value is smaller than a predefined value, the searching process is switched to SD searching to find an accurate solution.

To illustrate the performance of hybrid algorithm, a well-known system with nonlinear inequality constraint is employed. The simulation results confirm the superiority of BA-SD to some algorithms such as GA, DE, PSO, improved PSO algorithm with sequential quadratic programming algorithm (IPSO-SQP) in terms of the convergence speed and the accuracy without the premature convergence problem.

The reminder of paper is organized as follows: Section 2 summarizes the optimization algorithms briefly. The hybrid algorithm is introduced in Section 3. Section 4 contains the application of proposed algorithm to NOC problem. Simulation results are also obtained. Finally, conclusion is drawn in Section 5.

2. BRIEFLY DESCRIPTION OF OPTIMIZATION ALGORITHMS

The proposed hybrid algorithm is compared with frequently used algorithms in optimization problems, namely GA, PSO and DE in the problem in hand. This part describes these algorithms shortly.

2. 1. Standard GA GA is a population based optimization techniques that searches the best solution of a given problem based on the concepts of natural selection, genetics and evolution [2, 19]. The search is made starting from an initial population of individuals, often randomly generated. An individual is considered to be a possible candidate solution for the optimization problem in hand. At each evolutionary step, individuals are evaluated using a fitness function. The evolution (i.e., the generation of a new population) is made by means of three kind of operator: breeding, mutation and selection. Selection involves killing a given proportion of the population based on probabilistic "survival of the fittest". Killed individuals are replaced by children, which are created by breeding the remaining individuals in the population. For each child produced, breeding first requires probabilistic selection of two parent individuals, getting a more chance to fitter individuals to be chosen. Mutation allows new areas of the response surface to be explored by random alterations of optimization variables. GA iteratively improved the set of tentative solutions by applying the aforementioned stages to find a good solution.

In the traditional GA, all the variables of interest must be encoded as binary digits (genes) forming a string (chromosome). After a manipulation of binary-coded GA, the final binary digits are then decoded as original real numbers. On the other hand, in a real-coded GA, all genes in a chromosome are real numbers. To deal with practical engineering problems, the real-coded GA is more suitable than the binary-code GA [20, 21].

2. 2. Standard PSO Particle swarm optimization (PSO) is a kind of algorithm to search for the best solution by simulating the movement and flocking of birds [3]. Individual solutions in a population are viewed as "particles" that evolve or change their positions with time. Each particle in the swarm represents a candidate solution to the optimization problem. In the beginning, a population of candidate solutions is created. Each candidate solution is associated with a velocity. Then, the velocity of every particle is constantly adjusted according to the corresponding particle's experience and the particle's companions' experiences. At each iteration, the velocity of every particle is calculated as follows:

$$v_i^{t+1} = \omega\ v_i^t + c_1 r_1 (pbest_i^t - x_i^t) + c_2 r_2 (gbest^t - x_i^t) \quad (1)$$

where, x_i^t is the position of the particle *i* in *t*-th iteration, $pbest_i^t$ is the best previous position of this particle (memorized by every particle), $gbest^t$ is the best previous position among all the particles in *t*-th iteration (memorized in a common repository), ω is the inertia weight, c_1 and c_2 are acceleration coefficients and are known as the cognitive and social parameters, and finally r_1 and r_2 are two random number in the range [0,1]. After calculating the velocity, the new position of every particle can be worked out.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

The PSO algorithm performs repeated applications of the update equations above until a stopping criterion is reached.

2. 3. Differential Evolution (DE) Differential evolution (DE) is a population based stochastic function optimizer [4]. DE is an iterative optimization method which starts with the random initialization of a population of individuals in the search space. Then, it create new candidate solutions by combining existing ones according

to evolution mechanism, and then to the next iteration keeping whichever candidate solution has the best fitness on the optimization problem at hand. The evolution mechanism contains a mutation procedure in which consists of adding a weighted difference between two vectors while creating new one and comparing a newly created vector to the one from an existing population. If the new one is an improvement, it is accepted and forms part of the population, otherwise the new position is simply discarded. Another evolution mechanism is Crossover operator. Crossover procedure generates trial vector by randomly mixing the muted vector with the last target vector. If the trial vector produces a better performance for an objective function than that compared to, the new vector replaces it in the population. This procedure is called selection. DE iteratively improved the set of tentative solutions and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered. Refer to K. Price et al. and R. Storn et al. [4, 8] for more details.

3. THE PROPOSED HYBRID ALGORITHM

3. 1. Standard BA Bees Algorithm (BA) is a new member of meta-heuristics algorithm that tries to model natural behavior of honey bees in food foraging. Collecting, processing, and advertising of nectars are examples of intelligent behaviors of honey bees [17].

TABLE 1. The Pseudo Code of the Standard BA [18]

1. Initialize bee population with random solutions
2. Evaluate the fitness of the population
While (stopping criterion not met) //forming new bee population
3. Select elite bees
4. Select sites for neighborhood search
5. Recruit bees around the selected sites and evaluate fitness
6. Select the fittest bee from each site
7. Assign remaining bees to search randomly and evaluate their fitness
End while

One of the key differences of the bee swarm in comparison with other population-based methods is that it contains different types of bees such as scouts, onlookers, foragers and etc. Therefore, a bee swarm can provide different types of patterns which are utilized by the bees to adjust their flying trajectories. Consequently, this capability can result effective and robust algorithms to solve highly complex problems [14]. Basic bees algorithm entails several parameters to be initialized: number of scout bees (n), number of first part elites bees (m_1), number of second part elites bees (m_2), number of recruiting bees for first part elites bees (e_1), number of bees recruited for second part elites bees (e_2), number of rest bees for random search (r), size of patches and a stopping criterion. Pseudo code of the basic BA in its simplest form is presented in Table 1 [18].

BA starts with the n scout bees being placed randomly in the search space. The fitness of the sites visited by the scout bees are evaluated in step 2. In our study fitness function is performance index and our search space is determined with dimensions of control, so algorithm change the value of $\tilde{u}_i$ in each iteration to find the best solution for minimizing performance index. In step 3, bees that have highest fitness split to two parts. Each bee in the first part (i.e. upper part) use e_1 bees for searching in sites visited by them and its neighborhood. Their searching space is defined by fraction of a random function. This is the position of e_1 onlooker bees whit respect to m_1 elites bees:

$$x_i^j = x_i + 0.01 \times \text{rand}, \quad \begin{matrix} i = 1, 2, ..., m_1 \\ j = 1, 2, ..., e_1 \end{matrix} \tag{3}$$

where, x_i are position of m_1 first elite bees and x_i^j are position of e_1 onlooker bees. At the same time, each bee in the second part (i.e. lower part) recruits e_2 bees for searching in sites visited by them and its neighborhood. However, in this time their searching space is wider than first part. This is for preventing from trapping in local optimum.

$$x_k^l = x_k + 0.1 \times \text{rand}, \quad \begin{matrix} k = 1, 2, ..., m_2 \\ l = 1, 2, ..., e_2 \end{matrix} \tag{4}$$

In step 6, fitness of the employed bees calculated and again bees with highest fitness recognized and consider as a new elite bees. Existence of rest bees assured us that the final solution (i.e. position) is the global solution. The algorithm repeated until reach to stopping criterion. The readers who are interested in more detailed information on BA applications as well as the variants of the BA should refer to A. Baykasoglu et al. [22].

3. 2. Standard SD This method searches for the minimum of an *N*-dimensional objective function in the direction of a negative gradient,

$$-g(x) = -\nabla f(x) = -\left[\frac{\partial f(x)}{\partial x_1} \quad \frac{\partial f(x)}{\partial x_2} \quad \dots \quad \frac{\partial f(x)}{\partial x_N}\right]^T \tag{5}$$

with the step-size α_k (at iteration k) adjusted so that the function value is minimized along the direction by a (one-dimensional) line search technique like the quadratic approximation method. The pseudo code for SD is as follows:

***Step 1*:** with the iteration number k=0, find the function value $f_0 = f_0(\mathbf{x}_0)$ for the initial point $\mathbf{x}_0$.

***Step 2*:** Increment the iteration number k by one, find the step-size α_{k-1} along the direction of the negative gradient $-g_{k-1}$ by a (one-dimensional) line search like the quadratic approximation method.

$$\alpha_{k-1} = \text{ArgMin}_{\alpha} f(\mathbf{x}_{k-1} - \alpha \mathbf{g}_{k-1} / \|\mathbf{g}_{k-1}\|) \quad (6)$$

***Step 3*:** Move the approximate minimum by the step-size α_{k-1} along the direction of the negative gradient $-\mathbf{g}_{k-1}$ to get the next point.

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha \mathbf{g}_{k-1} / \|\mathbf{g}_{k-1}\|) \quad (7)$$

***Step 4*:** If $\mathbf{x}_k \approx \mathbf{x}_{k-1}$ and $f(\mathbf{x}_k) \approx f(\mathbf{x}_{k-1})$, then declare $\mathbf{x}_k$ to be minimum and terminate the procedure. Otherwise, go back to step 2.

A number of authors have successfully applied SD method to the solution of optimal control problems. In particular, [23] approximate the control variable as piecewise constant and solve the NLP problem using the SD technique.

3. 3. The Proposed Hybrid BA-SD In optimization algorithms, the exploration refers to the ability to investigate the various unknown regions in the solution space to discover the global optimum, while the exploitation refers to the ability of applying the knowledge of the previous good solutions to find better solutions.

The BA is a global optimization algorithm, which has a strong ability to find global optimistic result. However, it has a disadvantage which the search around global optimum is very slow. The SD, on the contrary, has a strong ability to find local optimistic result, but its ability to find the global optimistic result is weak. In this paper, a hybrid BA-SD optimization algorithm is proposed for solving NOC problems, which combines the merits of global search of BA and the local search of SD algorithm.

The BA-SD algorithm's searching process is also started from initializing a group of random bees. First, BA is run to search the global best position in the solution space. Then SD algorithm is used to search around the global optimum. In this way, this hybrid algorithm may find an optimum more quickly and accurately. The procedure for this BA-SD algorithm can be summarized as follows:

***Step 1*:** Initialize population with random solutions. Each individual in the population is a potential solution for NOC problem in hand.

***Step 2*:** Evaluate fitness of the population.

***Step 3*:** If the maximum iteration is reached go to Step 8, else forming new population.

***Step 4*:** Select sites for neighborhood search.

Step 5: Recruit bees for selected sites (more bees for best e sites) and evaluate fitnesses.

***Step 6*:** Select the fittest bee from each patch.

***Step 7*:** Assign remaining bees to search randomly and evaluate their fitnesses.

Step 8: Use the SD algorithm described in section 2 to search around global best which is founded by BD for some iteration. In this case, the best result obtained by BA is considered as the initial guess for the SD algorithm.

4. APPLICATION OF BA-SD

4. 1. Constrained Nonlinear Optimization Problem In this paper, the following nonlinear differential equation is considered:

$$\dot{x} = f(x(t), u(t), t) \quad (8)$$

where, $x(t) \in R^n$ is the state vector and the initial state $x(0)$ is given. $u(t) \in R^m$ is also the control vector limited by their lower and upper bounds as,

$$u_{min} < u_i(t) < u_{max}, \quad i = 1, 2, ..., m \quad (9)$$

In addition, it may be inequality constrains on state variables like,

$$\psi_i(X) \le 0, \quad i = 1, 2, ..., l \quad (10)$$

The performance index J associated with this system is a scalar function, which can be formulated as follows:

$$J(u(t)) = \phi(x(t_f), t_f) + \int_0^{t_f} L(x(t), u(t), t)\, dt \quad (11)$$

where, $\phi(.)$ is final state cost function and L(.) is interval cost function. $\phi(.)$ and L(.) functions are opted to achieve an appropriate design goal.

The aim is to determine the optimal control policy u(t) in the time interval $[t_0, t_f]$ such that performance index is minimized or maximized [24]. Indirect and direct methods are two general methods to solve this type of problems numerically [25]. Indirect method is based on the solution of a calculus of variations problem through the use of the Pontryagin's minimum principle [26] whereas in a direct method, the optimal control problem is approximated by a finite dimensional optimization problem, which can be cast in a nonlinear programming (NLP) form and solved accordingly [27]. This is achieved through control parameterization. In our study, the control u(t) is assumed to be a piecewise constant such as [2003]

$$u(t_k) = u_k, \quad t \in [t_k, t_{k+1}], \quad k = 0, 1, ..., N-1, \quad t_0 = 0, \quad t_N = t_f \quad (12)$$

Consequently, $N \times m$ parameters determine the control over $[0, t_f]$. The NLP problem is to find the stacked control vector defined by:

$\tilde{u} = [u_0^T, u_1^T, ..., u_{N-1}^T] = [\tilde{u}_1, ..., \tilde{u}_{N\times m}]$, where $\tilde{u}_i$, $i = 1, 2, ..., m \times N$ are scalars.

4. 2. Experimental Study and Discussion In this part, the proposed hybrid algorithm is applied to solving NOC of a mathematical system with nonlinear inequality constraint. This system involves a nonlinear inequality constraint and has been studied by several researchers [24, 28-33]. The state equations for the system are:

$$\dot{x}_1 = x_2 \quad (13)$$

$$\dot{x}_2 = -x_2 + u \quad (14)$$

$$\dot{x}_3 = x_1^2 + x_2^2 + 0.005u^2 \quad (15)$$

with initial condition $X(0) = [0 \ -1 \ 0]^T$. The nonlinear inequality constraint to be satisfied is

$$h(x) = x_2 + 0.5 - 8(t - 0.5)^2 \le 0 \quad (16)$$

The upper and lower bounds of control is as:

$$-20 \le u \le 20 \quad (17)$$

Moreover, the performance index to be minimized is

$$J = x_3(t_f) \quad (18)$$

where, $t_f = 1$. To solve this problem by means of the proposed method, the time interval $[t_0, t_f]$ is discretized in $N = 20$ time intervals as done by W. Mekarapiruk et al. [28].

In 1998 [32], this problem has been solved using the control parameterization technique, and the obtained results was J = 0.1816. In 1997, Mekarapiruk et al. [28] proposed a penalty function and solve this inequality state constraint. They obtained a result of J =0.1769.

In 2011, Modares et al. [24] obtained a result of J =0.1728 using a hybrid PSO algorithm with Sequential Quadratic Programming (SQP) algorithm.

In order to show the performance of the proposed algorithm in the problem in hand, it is compared to some frequently used optimization algorithms, including GA, PSO, BA, SD and IPSO-SQP [24], which is the latest research in NOC problem. For all simulations, for PSO algorithm both c_1 and c_2 are set to 2 [34] and the inertia weight ω decreases linearly from 0.9 to 0.4. For GA, the crossover and mutation rates are considered as 0.8 and 0.1, respectively [35]. For DE, all of its parameters are the same as in R. Storn et al. [8].

Simulation results are presented in Tables 2-7 for population size of 20, 40 and 60, respectively, where each algorithm is implemented 20 times independently. The results indicate in how many iterations as well as necessary time, the convergence of the solution or success is met. The average of elapsed time in 20 runs is considered as a criterion for computational time. From these tables, it is obvious that the hybrid BA-SD algorithm is more robust and accurate than other algorithms. In addition, comparing the results of the proposed method with Goh et al. [32] and Mekarapiruk et al. [28], it is concluded that the proposed method has better accuracy than others.

Finally, in order to show the feasibility of the proposed algorithm, the results of BA-SD are compared with IPSO-SQP [24] including optimal control policy and constraint trajectory. The optimal control policy is given in Table 8. Moreover, Figure 1 shows the optimum control trajectory obtained by BA-SD and IPSO-SQP whereas the trajectory of the function of state h(**X**) is shown in Figure 2. As can be seen, the constraint is satisfied throughout the time interval.

TABLE 3. Iterations and Elapsed Time Obtained By GA, DE, PSO, IPSO-SQP and BA-SD for Population Size=20

Criterion	GA	DE	PSO	IPSO-SQP	BA-SD
Iteration	317.56	189.42	252.94	170.45	168.92
Elapsed Time (Sec)	1129.1574	150.3624	170.6525	108.5012	105.9501

TABLE 5. Iterations and Elapsed Time Obtained By GA, DE, PSO, IPSO-SQP and BA-SD for Population Size=40

Criterion	GA	DE	PSO	IPSO-SQP	BA-SD
Iteration	311.28	184.95	249.64	169.95	154.57
Elapsed Time (Sec)	1107.9487	147.6557	168.4694	107.3312	107.3312

TABLE 7. Iterations and Elapsed Time Obtained By GA, DE, PSO, IPSO-SQP and BA-SD for Population Size=60

Criterion	GA	DE	PSO	IPSO-SQP	BA-SD
Iteration	294.95	169.95	231.63	145.29	133.73
Elapsed Time (Sec)	1721.9487	192.1882	219.1999	126. 2628	125.9989

TABLE 2. Results Obtained By GA, DE, PSO, IPSO-SQP and BA-SD for Population Size=20

Algorithm	Best	Mean	Worst
GA	0.22185	0.28691	0.31993
DE	0.20984	0.25347	0.30675
PSO	0.21892	0.26737	0.32745
IPSO-SQP	0.17279	0.17284	0.17289
BA-SD	0.17264	0.17269	0.17277

TABLE 4. Results Obtained By GA, DE, PSO, IPSO-SQP and BA-SD for Population Size=40

Algorithm	Best	Mean	Worst
GA	0.20885	0.27861	0.31069
DE	0.19854	0.23877	0.29838
PSO	0.20429	0.24731	0.30176
IPSO-SQP	0.17277	0.17283	0.17286
BA-SD	0.17261	0.17101	0.17066

TABLE 6. Results Obtained By GA, DE, PSO, IPSO-SQP and BA-SD for Population Size=60

Algorithm	Best	Mean	Worst
GA	0.20172	0.25496	0.30525
DE	0.19523	0.22106	0.26487
PSO	0.20334	0.23478	0.29451
IPSO-SQP	0.17276	0.17281	0.17284
BA-SD	0.17258	0.17265	0.17272

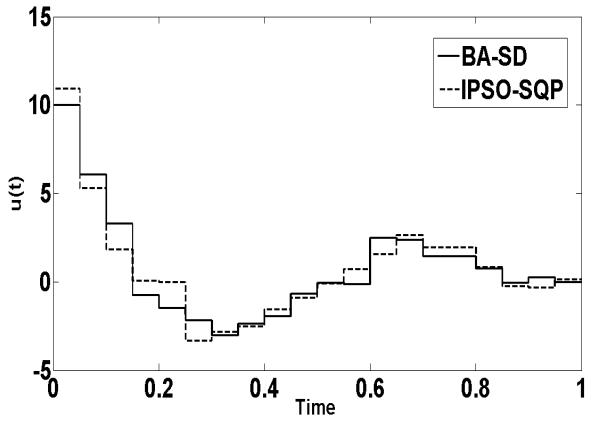

Figure 1. Optimum control trajectory for the system with nonlinear inequality constraint problem by BA-SD and IPSO-SQP

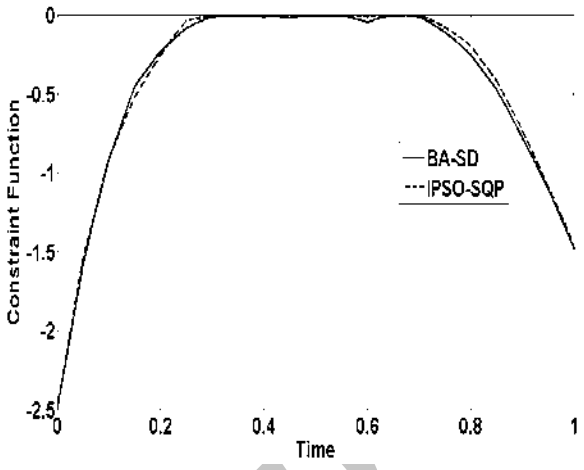

Figure 2. Trajectory of the function *h(x)* of the system with nonlinear inequality constraint after applying optimal control policy by BA-SD and IPSO-SQP

TABLE 8. Optimal Control Policy of the System with Nonlinear Inequality Constraint for 20 Time Intervals by BA-SD and IPSO-SQP

Time interval k	Time t_k	Control $u(t_{k-1})$ (BA-SD)	Control $u(t_{k-1})$ (IPSO-SQP)
1	0.05	10.0236	10.9455
2	0.10	6.0781	5.3248
3	0.15	3.3180	1.8326
4	0.20	-0.7540	0.0791
5	0.25	-1.4842	0.0013
6	0.30	-2.1701	-3.312
7	0.35	-3.0262	-2.8228
8	0.40	-2.3567	-2.5280
9	0.45	-1.9405	-1.5667
10	0.50	-0.6579	-0.9150
11	0.55	-0.0668	-0.0754
12	0.60	-0.1302	0.7194
13	0.65	2.5029	1.5740
14	0.70	2.3647	2.6541
15	0.75	1.4454	1.9597
16	0.80	0.2625	0.5037
17	0.85	-0.4269	-0.6190
18	0.90	-1.2140	-1.6743
19	0.95	-0.9107	-1.7820
20	1.00	-1.1981	-1.2969

5. CONCLUSION

This paper presented a hybrid BA-SD algorithm which combines merits of BA and SD methods for solving NOC. It was illustrated that the hybrid algorithm has the advantage of both BA and SD methods while it does not inherent their shortages. BA converges rapidly to a local optimum solution whereas SD can achieve faster convergent speed around global optimum and finally the convergent accuracy can be higher. The case of study is for system with nonlinear inequality constraint. The performance of the proposed algorithm was compared with those of the SD and some global search algorithms such as GA and PSO algorithm, in terms of the parameter accuracy and convergence speed. Simulation results illustrated that the proposed algorithm is more successful in comparison with others. Future works in this area contain the application of other inspired algorithms such as memetic algorithm in NOC problem.

6. REFERENCES

1. Bryson, J.A. and Ho, Y.C., "Applied Optimal Control: Optimization, Estimation and Control", Washington, DC: Hemisphere, (1975).
2. Goldberg, D., "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, Reading, MA, (1989).
3. Kennedy, J. and Eberhart, R.C., "Particle Swarm Optimization", ***IEEE International Conference on neural networks***, 1995, 1942-1948.
4. Price, K. and Storn, R., "Differential Evolution: A Simple Evolution Strategy for Fast Optimization", ***Dr. Dobb's Journal***, Vol. 22, (1997), 18-24.
5. Sewald, H. and Kumar, R.R., "Genetic Algorithm Approach For Optimal Control Problems with Linearity Appearing Controls", ***Journal of Guidance, Control and Dynamics***, Vol. 18, (1995), 177-182.
6. Yamashita, Y. and Shima, M., "Numerical Computational Method Using Genetic Algorithms for the Optimal Control Problem with Terminal Constraints and Free Parameters", *Nonlinear Analysis: **Theory, Methods and Applications***, Vol. 30, (1997), 2285-2290.
7. Lee, K.S. and Geem, Z.W., "A New Meta-heuristic Algorithm for Continuous Engineering Optimization: Harmony Search Theory and Practice", ***Computing Methods and Application in Mechanical Engineering***, Vol. 194, (2004), 3902-3933.
8. Storn, R., "System Design by Constraint Adaptation and Differential Evolution", *IEEE Transactions on Evolutionary Computation*, Vol. 3, (1999), 22-34.
9. Clerc, M. and Kennedy, J., "The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space", ***IEEE Transactions on Evolutionary Computation***, Vol. 6, (2002), 58-73.
10. Babu, B.V. and Angira, R., "Modified Differential Evolution (MDE) for Optimization of Non-Linear Chemical Processes", ***Computers and Chemical Engineering***, Vol. 30, (2006), 989-1002.
11. Varadarajan, M. and Swarup, K.S., "Differential Evolution Approach for Optimal Reactive Power Dispatch", ***Applied Soft Computing***, Vol. 8, (2008), 1549-1561.
12. Arumugam, M.S. and Rao, M.V.C., "On the Improved Performances of the Particle Swarm Optimization Algorithms with Adaptive Parameters, Cross-Over Operators and Root Mean Square (RMS) Variants for Computing Optimal Control of a Class of Hybrid Systems", ***Applied Soft Computing***, Vol. 8, (2008), 324-336.
13. Herrera, F. and Zhang, J., "Optimal Control of Batch Processes Using Particle Swam Optimization with Stacked Neural Network Models", ***Computers and Chemical Engineering***, Vol. 33, (2009), 1593-1601.
14. Akbari, R., Mohammadi, A. and Ziarati, K., "A Novel Bee Swarm Optimization Algorithm for Numerical Function Optimization", ***Communications in Nonlinear Science and Numerical Simulation***, Vol. 15, (2010), 3142-3155.
15. Karaboga, D. and Akay, B., "A Comparative Study of Artificial Bee Colony Algorithm", ***Applied Mathematics and Computation***, Vol. 214, (2009), 108-132.
16. Singh, A., "An Artificial Bee Colony Algorithm for the Leaf-Constrained Minimum Spanning Tree Problem", ***Applied Soft Computing***, Vol. 9, (2009), 625-631.
17. Karaboga, D. and Basturk, B., "On the Performance of Artificial Bee Colony (ABC) Algorithm", ***Applied Soft Computing***, Vol. 8, (2008), 687-697.
18. Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S. and Zaidi, M., "The Bees Algorithm: A Novel Tool for Complex Optimization Problems", ***International Virtual Conference on Intelligent Production Machines and Systems***, (2006), 454-461.
19. Holland, J.H., "Adaptation in Natural and Artificial Systems", The University of Michigan Press, Ann Arbor, Michigan, (1975).
20. Chambers, L., "Practical Handbook of Genetic Algorithms: New Frontiers", CRC Press, (1995).
21. Chang, W., "Nonlinear System Identification and Control Using a Real-Coded Genetic Algorithm", *Applied **Mathematical Modeling***, Vol. 31, (2007), 541-550.
22. Baykasoglu, A., Özbakır, L. and Tapkan, P., "Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem", Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, I-Tech Education and Publishing, Vienna, Austria, (2007), 113-144.
23. Kirk, D.E., "Optimal Control Theory: An Introduction", Dover Publications, Inc, (2004).
24. Modares, H. and Naghibi Sistani, M.B., "Solving Nonlinear Optimal Control Problems Using a Hybrid IPSO–SQP Algorithm", ***Engineering Applications of Artificial Intelligence***, Vol. 24, (2011), 476-484.
25. Stryk, O.V. and Bulirsch, R., "Direct and Indirect Methods for Trajectory Optimization", ***Annals of Operations Research***, Vol. 37, (1992), 357-373.
26. Bryson, A.E., "Dynamic Optimization", Addison-Wesley, Menlo Park, NY, (1999).
27. Agrawal, S.K. and Fabien, B.C., "Optimization of Dynamic Systems, Solid Mechanics and Its Applications", Kluwer Academic Publishers, Dordrecht, The Netherlands, (1999).
28. Mekarapiruk, W. and Luus, R., "Optimal Control of Inequality State Constrained Systems", ***Industrial & Engineering Chemistry Research***, Vol. 36, (1997), 1686-1694.
29. Elnagar, G., Kazemi, M.A. and Razzaghi, M., "The Pseudo-Spectral Legendre Method for Discretizing Optimal Control

Problems", ***IEEE Transactions on Automatic Control***, Vol. 40, (1995), 1793-1796.

30. Teo, K.L., Goh, C.J. and Wong, K.H., "A Unified Computational Approach to Optimal Control Problems", Wiley: New York, (1991), 146-147.
31. Vlassenbroeck, J., "A Chebyshev Polynomial Method for Optimal Control with State Constraints", ***Automatica***, Vol. 24, (1988), 499-506.
32. Goh, C.J. and Teo, L.K., "Control Parameterization: A Unified Approach to Optimal Control Problems with General Constraints", ***Automatica***, Vol. 24, (1988), 3-18.
33. Mehra, R. K. and Davis, R.E., "A Generalized Gradient Method for Optimal Control Problems with Inequality Constraints and Singular Arcs", ***IEEE Transactions on Automatic Control***, Vol. 17, (1972), 69-78.
34. Shi, Y. and Eberhart, R.C., "A Modified Particle Swarm Optimizer", ***In Proceedings of the Conference on Evolutionary Computation***, (1998), 69-73.
35. Grefenstette, J.J., "Optimization of Control Parameters for Genetic Algorithms", ***IEEE Transactions on Systems Man and Cybernetics***, Vol. 16, (1986), 122-128.

Constrained Nonlinear Optimal Control via a Hybrid BA-SD

A. Alfi [a], A. Khosravi [b]

[a] *Faculty of Electrical and Robotic Engineering, Shahrood University of Technology, P.O. Box 36199-95161, Shahrood, Iran*
[b] *Department of Electrical and Computer Engineering, Babol University of Technology, P.O. Box 484, Babol, Iran*

P A P E R I N F O

Paper history:
Received 21 February 2012
Received in revised form 8 June 2012
Accepted 14 June 2012

Keywords:
Constrained Optimization
Swarm Intelligence
Bees Algorithm
Steepest Descent
Differential Evolution
Genetic Algorithm

چکیده

رفتار غیرمحدب در سیستم‌های غیرخطی، کاربرد تکنیک‌های بهینه‌سازی کلاسیک را برای حل اینگونه سیستم‌ها محدود می‌سازد. در این مقاله الگوریتم ترکیبی به نام BA-SD از ترکیب الگوریتم زنبور (BA) با روش شدیدترین نزول (SD) برای حل مسائل کنترل بهینه غیرخطی پیشنهاد می‌شود. علت پیشنهاد این الگوریتم به‌واسطه سریع بودن همگرایی الگوریتم زنبور در مراحل اولیه و کند بودن آن در مرحله جستجو در اطراف نقطه بهینه است. در حالی که روش شدیدترین نزول دارای قابلیت پایین در همگرایی نقطه بهینه محلی و سرعت بالای همگرایی در اطراف نقطه بهینه کلی و در نتیجه دقت همگرایی بالا است. در الگوریتم پیشنهادی، در مرحله اولیه از فرایند جستجو، الگوریتم زنبور به منظور یافتن یک حل بهینه تقریبی استفاده می‌شود. در این حالت، توانایی جستجوی کلی افزایش می‌یابد. چنانچه مقدار تغییر ارزش کمتر از مقدار از پیش تعریف شده باشد، در فرایند جستجو روش شدیدترین نزول برای شتاب بخشیدن به فرایند جستجو و یافتن حل دقیق جایگزین می‌شود. بدین صورت یافتن حل بهینه از دقت بالاتری برخوردار است. شبیه‌سازی‌ها نشان‌دهنده کارایی الگوریتم پیشنهادی است.

doi: *10.5829/idosi.ije.2012.25.03c.03*