Engineering Science and Technology, an International Journal 17 (2014) 131–136

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: http://www.elsevier.com/locate/jestch

Full length article

Differential evolution and simulated annealing algorithms for mechanical systems design

H. Saruhan*

University of Düzce, Faculty of Engineering, Department of Mechanical Engineering, 81620, Konuralp Campus, Turkey

ARTICLE INFO

Article history:
Received 23 January 2014
Received in revised form
10 April 2014
Accepted 23 April 2014
Available online 20 May 2014

Keywords:
Differential evolution
Simulated annealing
Genetic algorithm
Design optimization

ABSTRACT

In this study, nature inspired algorithms – the Differential Evolution (DE) and the Simulated Annealing (SA) – are utilized to seek a global optimum solution for ball bearings link system assembly weight with constraints and mixed design variables. The Genetic Algorithm (GA) and the Evolution Strategy (ES) will be a reference for the examination and validation of the DE and the SA. The main purpose is to minimize the weight of an assembly system composed of a shaft and two ball bearings. Ball bearings link system is used extensively in many machinery applications. Among mechanical systems, designers pay great attention to the ball bearings link system because of its significant industrial importance. The problem is complex and a time consuming process due to mixed design variables and inequality constraints imposed on the objective function. The results showed that the DE and the SA performed and obtained convergence reliability on the global optimum solution. So the contribution of the DE and the SA application to the mechanical system design can be very useful in many real-world mechanical system design problems. Beside, the comparison confirms the effectiveness and the superiority of the DE over the others algorithms – the SA, the GA, and the ES – in terms of solution quality. The ball bearings link system assembly weight of 634,099 gr was obtained using the DE while 671,616 gr, 728213.8 gr, and 729445.5 gr were obtained using the SA, the ES, and the GA respectively.

1. Introduction

Mechanical system design problems are complex activity in which computing capabilities are more and more required. Most mechanical systems are essential to modern development processes, where the efficient optimization methods are used for ever-increasing demands for high quality, low cost, and compact size. Developments in computer technology have proved to be a great chance to the world of mechanical systems design optimization. Any efficient optimization algorithm explores to investigate new and unknown areas in search space and exploit to make use of knowledge found at point previously visited to help find better solution point. Nature inspired algorithms can provide a remarkable balance between exploration and exploitation of the search space. Most of real world engineering problems characteristics are mixed discrete, integer, and continuous variables. It is required to satisfy a set of inequality and equality constraints imposed on the problems. Gradient based optimization methods are not suitable to be used in such problems. Although there are many studies such as [15,16] on optimizing mechanical system design problems, there are a few studies reported in the literature using discrete design variables whose values may be taken in normalized tables. Some considerable study on using discrete design variables references [7,19,20] can be cited. From this point of view, this study provides use of Differential Evolution (DE) and Simulated Annealing (SA) to seek a global optimum solution to problem in hand. The problem contains non-linear equations, equality constraints and mixed variables. Also there are independent discrete parameters taken from standardized tables. The DE and the SA imitate optimization processes found in nature. The DE is a population based direct search method that utilizes a set of parameter vectors that interact in a way that is inspired by the evolution of living species [13]. The DE is motivated by genetic annealing [5]. The SA algorithm imitates the process of annealing in metals as they cool from liquid to solid states. These algorithms both employ the generation of random numbers when they search for the optimum solution. They do not require the evaluation of gradients of the objective function.

* Tel.: +90 380 542 10 36; fax: +90 380 542 10 37.
E-mail addresses: hamitsaruhan@duzce.edu.tr, hamitsaruhan@hotmail.com.
Peer review under responsibility of Karabuk University.

http://dx.doi.org/10.1016/j.jestch.2014.04.006

2. The optimization algorithms

In this section of the paper, the fundamental intuition of the SA and the DE and how they process are given. The SA was proposed by Kirkpatrick et al. [8] to deal with complex non-linear problems. It showed the analogy between simulating the annealing of solid as proposed by Metropolis et al. [9]. The SA is an iterative improvement algorithm for a global optimization. It is inspired from thermodynamic to simulate the physical process of annealing of molten metals [2,10]. The method attempts to model the behavior of atoms in the mechanical procedure of annealing of metals. It obtains the minimum value of energy by simulating annealing which is a process employed to obtain a perfect crystal by gradual cooling of molten metals [11] in order to keep the system of melt in a thermodynamic equilibrium at given temperature. Thus, it exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure. At high temperature, the atoms in the molten metal can move freely with respect to each other as the cooling proceeds, the atoms of metal become more ordered and the system naturally converges towards a state of minimal energy. This formation of crystal mostly depends on the cooling rate. If the metal is cooled at very fast rate, the atoms will form an irregular structure and the crystalline state may not be achieved. The SA makes use of the algorithm [9] which provides an efficient simulation according to a probabilistic criterion stated as:

$$P(\Delta E) = \begin{cases} 1, & \text{if } \Delta E < 0 \\ e^{(-\Delta E/Tk)}, & \text{otherwise} \end{cases} \quad (1)$$

Thus, if $\Delta E < 0$, the probability, P, is one and the change - the new point- is accepted. Otherwise, the modification is accepted at some finite probability. Each set of points of all atoms of a system is scaled by its Boltzmann probability factor $e^{(-\Delta E/Tk)}$ where ΔE is the change in the energy value from one point to the next, k is the Boltzmann's constant and T is the current temperature as a control parameter. The general procedure for employing the SA as follows; Step 1: Start with a random initial solution, x, and an initial temperature, T, which should be high enough to allow all candidate solutions to be accepted and evaluate the objective function. Step 2: Set $i = i + 1$ and generate new solution $(x_i^{\text{new}} = x_i + rSL_i)$ where r is random number and SL_i at each move should be decreased with the reduction of temperature. Evaluate$F_i^{\text{new}} = F(x_i^{\text{new}})$. Step 3: Choose accept or reject the move. The probability of acceptance (depending on the current temperature) if $F_i^{\text{new}} < F_{i-1}$, go to Step 5, else accept F_i as the new solution with probability $e^{(-\Delta E/T)}$, where $\Delta E = F_i^{\text{new}} - F_{i-1}$ and go to Step 4. Step 4: If F_i was rejected in Step 3, set $F_i^{\text{new}} - F_{i-1}$. Go to Step 5. Step 5: If satisfied with the current objective function value, F_i, stop. Otherwise, adjust the temperature $(T^{\text{new}} = Tr_T)$ where r_T is temperature reduction rate called cooling schedule and go to Step 2. The process is done until freezing point is reached. The major advantages of the SA are an ability to avoid becoming trapped in local optimum and dealing with highly nonlinear problem with many constraints.

The DE is an evolutionary optimization algorithm proposed by Storm and Price [17]. It has been successfully applied to wide range of engineering design problems. The DE is a population based direct search method that utilizes a set of parameter vectors that interact in a way that is inspired by the evolution of living species [13]. The population based search algorithms are becoming increasingly popular for solving highly nonlinear, complex, and constrained optimization problems. A number of parameters include the population size, step size parameter, and crossover rate play a key role for convergence of the DE. The DE algorithm is robust and well suited to handle non-differentiable functions. The DE algorithm works as follows: It starts with initial population generated randomly. For creating next generation, the population is then evolved by evolutionary operators including mutation, crossover, and selection, special kind of differential operators that differ from classical mutation and crossover of the Genetic Algorithm (GA). The role of the operators is to ensure better solutions from good ones (exploitation) and to cover sufficiently the solution space for discovering the global optimum (exploration) [14]. The mutation operator applies the vector differentials between the existing population members of determining both the degree and direction of perturbation applied to the individual [4]. This self-referential mutation operator allows a gradual exploration of the search space [18]. A mutation process begins by randomly selecting three individuals in the population to form a triplet [4]. The general procedure for employing the DE as follows; Step 1: Uniformly initialize the population of individuals with random positions in the search space and evaluate the objective values of all individuals in the population. The initial population is set as;

$$x_{i,j}(0) = x_j^L + \text{rand}(0,1)\left(x_j^U - x_j^L\right) \quad (2)$$

where i indicate design variable, j indicate the population member, rand(0,1) is a uniformly distributed random number lying between 0 and 1. U and L are the upper and lower limits of each design variable respectively. Step 2: Create the trial vector for each individual using mutation and crossover operators. Fig. 1 gives the DE mutation and crossover schema in 2-D search space. $x_{i,G}$ (i = 1, 2, 3,…NP) is target vector where NP is population size, i indexes the population, and G is the generation to which the population belongs. For each individual, three different individuals; one base vector,$x_{r1,G}$, and others randomly selected vectors $x_{r2,G}$ and $x_{r3,G}$ are chosen from the current population. Next, a differential variation vector is generated by subtracting vector $x_{r3,G}$ from vector$x_{r2,G}$. Then a mutation scale factors, F, is multiplied with difference term and added to the third term $x_{r1,G}$ to form a mutant vector, $v_{i,G}$. Following mutation process, crossover operator takes place. The crossover operators mixes the target vector $x_{i,G}$ of the current generation and mutant vector $v_{i,G}$ to form a trial vector $u_{i,G+1}$.

$$v_{i,G} = x_{r1,G} + F(x_{r2,G} - x_{r3,G}) \quad (3)$$

$$u_{i,G+1} = \begin{cases} v_{i,G} & \text{if } \text{rand}(0,1) \le CF \\ x_{i,G+1} & \text{else} \end{cases} \quad (4)$$

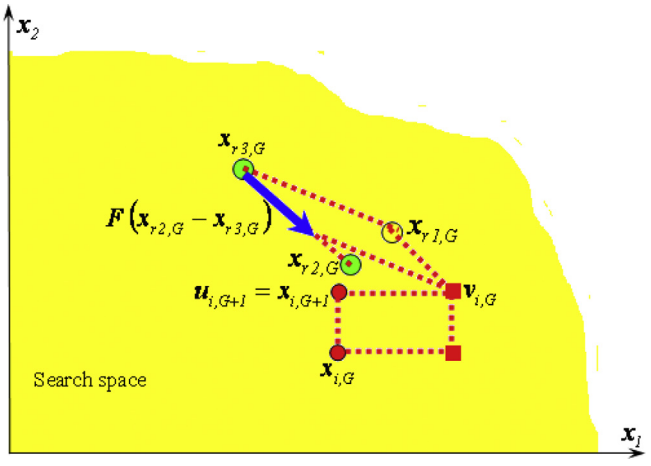

Fig. 1. The DE mutation and crossover schema in 2-D search space.

where r_1, r_2, and r_3 are random indexes which are mutually different integer. F is a real and constant mutation factor which controls the application of the differential variation $(x_{r2,G} - x_{r3,G})$. CF is another control parameter called crossover factor lies in the range of 0.0–1.0. F and CF affect the robustness of search process. The robustness of the process guarantees a high probability of getting the optimum result. Typical value of F is in the range of 0.4–1.0 [1]. Thus, the mutation factor scales the difference of any two of the three vectors. Step 3: Evaluate the objective values of all points in the trial vector. Step 4: Select the better vector from the competitor using selection operator. The selection operator is used to determine which one of the individual will survive for the next generation. The objective function value of each trial vector, $F(u_i(t))$, is composed to that of its corresponding target vector, $F(x_i(t))$, in the current population. The selection operator process can be expressed as:

$$x_i(t+1) = \begin{cases} u_i(t) & \text{if} \quad F(u_i(t)) \le F(x_i(t)) \\ x_i(t) & \text{if} \quad F(u_i(t)) > F(x_i(t)) \end{cases} \tag{5}$$

Step 5: If satisfied with current objective value stop. Otherwise go back to Step 2.

3. Problem statement

The design optimization of a ball bearing pivot link, using Genetic Algorithm (GA) and Evolution Strategy (ES), which is performed in the study by Moreau and Lafon [12]; will be a reference for the examination and validation of the DE and the SA performed in this study. Fig. 2 shows the ball bearing pivot link system which was taken partially from Ref. [12]. It is desired to size the lengths x_1 and x_2 and the two ball bearings R_i (numbered 1) and R_j (numbered 2) for minimizing the weight of the assembly system composed of a shaft and two ball bearings.

Table 1 gives the constant parameters which are defining and selecting choice of the problem at hand. R_i and R_j represent the selection choice of the two ball bearings. This selection of the ball bearings having a diameter of 30 mm to45 mm, from 1 to 28 in the same order as standardized table. The constant parameters which define the problem are $d_o = 28$ mm, $d_5 = 28$ mm, $b_o = 30$ mm, $b_3 = 6$ mm, and mass $\rho = 7800$ kg/m^3.

For the formulation of the problem in hand, the list of functional relations and conditions to describe the behavior of the linkage system are given as geometrical conditions, stress conditions on the shaft, and conditions on the bearing life span. The formulations are simplified in order to deal with numerical structure of the problem. The formulations of the problem are given in the following:

Table 1
Parameters choice of the two ball bearing system.

R_i, R_j	d_i (mm)	D_i (mm)	b_i (mm)	C_i (N)	m_i (gr)
1	30	42	7	4490	27
2	30	47	9	7280	51
3	30	55	9	11,200	85
4	30	55	13	13,300	120
5	30	62	16	19,500	200
6	30	72	19	28,100	350
7	30	90	23	43,600	740
8	35	47	7	4750	30
9	35	55	10	9560	80
10	35	62	9	12,400	110
11	35	62	14	15,900	160
12	35	72	17	25,500	290
13	35	80	21	33,200	460
14	35	100	25	55,300	950
15	40	52	7	4940	34
16	40	62	12	13,800	120
17	40	68	9	13,300	130
18	40	68	15	1600	190
19	40	80	18	30,700	370
20	40	90	23	41,000	630
21	40	110	27	63,700	1250
22	45	58	7	6050	40
23	45	68	12	14,000	140
24	45	75	10	15,600	170
25	45	75	16	20,800	250
26	45	85	19	33,200	410
27	45	100	25	52,700	830
28	45	120	29	76,100	1550

$$\begin{aligned} F(x_1, x_2, R_i, R_j) \;=\; & m_1 + m_2 + \rho\frac{\pi}{4}\Big[0.5\Big(d_1^2(b_1 - b_o) \\ & - (d_2 + 2((d_4 - d_2)/2))^2(b_1 + b_2)\Big)\Big] \\ & + \rho\frac{\pi}{4}\Big[(d_1 + 2((d_3 - d_4)/2))^2 b_3 + d_2^2 L_2 + x_1 d_1^2 \\ & + d_2^2(x_2 - b_3)\Big] \end{aligned} \tag{6}$$

Subject to

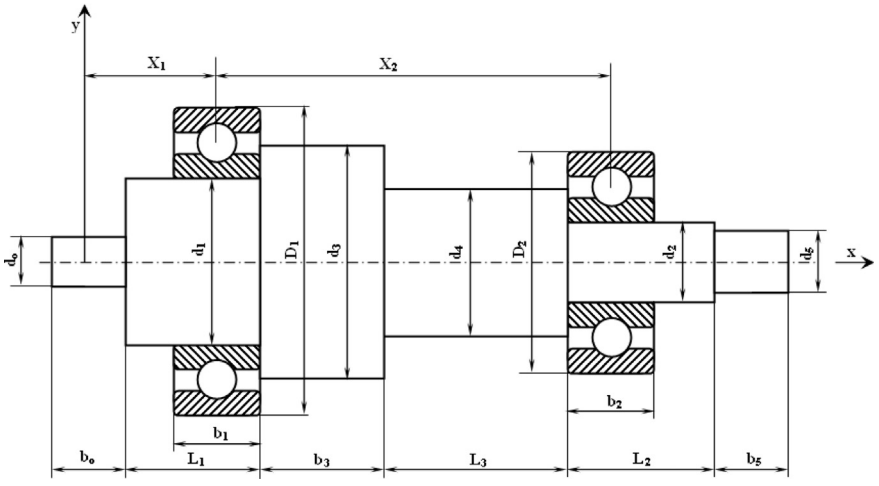

Fig. 2. Ball bearing pivot link system.

$$5 \le x_1 \le 200 \tag{7}$$

$$10 \le x_2 \le 200 \tag{8}$$

$$g_1(x_1, x_2, R_i, R_j) = 0.5b_1 - x_1 + (0.5b_o + 8) \le 0 \tag{9}$$

$$g_2(x_1, x_2, R_i, R_j) = 0.5b_1 + 0.5b_2 - x_2 + 13 \le 0 \tag{10}$$

$$g_3(x_1, x_2, R_i, R_j) = D_2 - D_1 \le 0 \tag{11}$$

$$g_4(x_1, x_2, R_i, R_j) = d_o - d_1 \le 0 \tag{12}$$

$$g_5(x_1, x_2, R_i, R_j) = d_5 - d_2 \le 0 \tag{13}$$

$$g_6(x_1, x_2, R_i, R_j) = D_1 - 100 \le 0 \tag{14}$$

$$g_7(x_1, x_2, R_i, R_j) = x_2 + x_1 + 0.5b_2 - 177 \le 0 \tag{15}$$

$$g_8(x_1, x_2, R_i, R_j) = (615.51x_1 + 3930)^{1/3} - d_1 \le 0 \tag{16}$$

$$g_9(x_1, x_2, R_i, R_j) = 29216\left(1 + \frac{x_1}{x_2}\right) - C_1 \le 0 \tag{17}$$

$$g_{10}(x_1, x_2, R_i, R_j) = 29216\left(\frac{x_1}{x_2}\right) - C_2 \le 0 \tag{18}$$

Handling of constraints is an important issue when solving complex real world problems. In this study, penalty function is used to improve convergence of the algorithm performance. In case of any violation of a constraint boundary, the fitness of corresponding solution is penalized by penalty function, *PF*, and thus kept within feasible regions of the design space by increasing the value of the objective function. A unique static penalty function developed by Homaifar et al. [6] is used with multiple violation levels set for each constraint in order to maintain a feasible solution. Each constraint is defined by the relative degree of constraint penalty coefficient, r_j, for the j-th constraints have to be judiciously selected.

$$\mathrm{PF} = \sum_{j=1}^{\mathrm{NC}} r_j \left(\max\left[0, g_j\right]\right)^2 \quad NC \text{ is number of constraints} \tag{19}$$

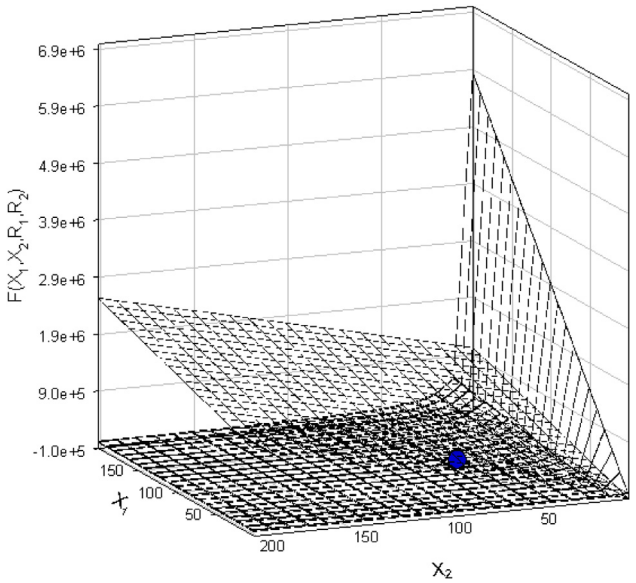

Fig. 3. The plot of the design objective function and constraints with global optimum solution point.

4. Employing the algorithms

In the optimization problem in hand, the design variables vectors,x_1 and x_2, R_i, and R_j, represent a solution that minimizes the objective function for weight of the assembly system. By employing the SA, a random initial point is selected at high temperature and a series of moves are made according to defined annealing schedule. The change in the objective function values, ΔE, is computed at each move. A new solution is generated in the neighborhood of the current configuration in each of iteration. This new solution is automatically accepted with probability of 1, if it results in decreased objective function value. Otherwise, if the new solution is increased the objective function value, the acceptance is given with a small probability, $e^{(-\Delta E/Tk)}$, where T is the current temperature and k is Boltzmann's constant. The probability expression suggests that if the temperature of the system is large, the probability of accepting the solution increases. Otherwise, if T is low, the probability of accepting solution decreases. Therefore, the temperature needs to be high at the beginning. As the iteration proceeds, the temperature is gradually decreased until the stopping condition is met. There are many ways to determine when to stop running the algorithm: One is the temperature when reduced to a threshold. Another is to reach a pre-specified number of temperature transitions. All the generating and acceptance depend on the temperature. The global optimum point can be converged by carefully controlling the rate of cooling of the temperature. The important setting parameters of the SA for this study are chosen as follows: Initial temperature $T = 10{,}000$, temperature reduction rate $r_T = 0.6$, and number of iterations performed at a particular temperature $n = 5$.

By employing the DE, the DE has several variant versions: DE/rand/1/bin, DE/best/1/bin, DE/rand/1/exp, DE/best/1/exp, DE/rand/2/bin, DE/best/2/bin, DE/rand/2/exp, DE/best/2/bin, DE/rand−to−best/1/bin, DE/rand−tp−best/1/exp which are in general form DE/x/y/z where DE stands for the DE, x is a string denoting the vector to be perturbed, y is the number of difference vectors considered for perturbation of x, z is for the type of crossover used (exp: exponential; bin: binomial) [3].The version used in this study is the DE/rad/1/bin which appears to be the most frequently used variant. Thus, "DE" stands for differential evolution, "rand" stands for the randomly selected individuals to compute the mutation values, "1" is the number of pairs of chosen solutions, and "bin" stands for binomial recombination. The DE initializes the population by assigning values from a uniform random distribution between the upper and the lower limits of each design variable. The population size remains the same through out the process. The parameter vectors are modified for locating better solution in each generation. Initial values selected for control in the DE algorithm are: population size = 100; number of generations = 1000, differential evolution factor ($F = 0.5$), crossover rate = 0.9.

5. Results

In general design optimization, concerning the convergence reliability, the algorithms should find a design to the global

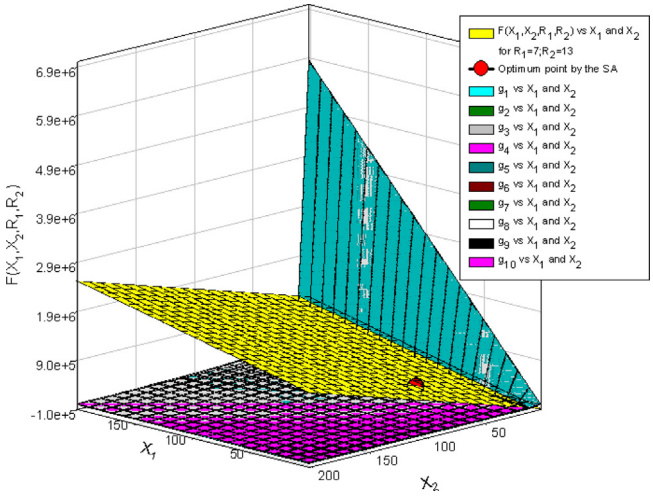

Fig. 4. The lengths x_1 and x_2 for the weight of the Ball bearing pivot link system versus the design constraints by the SA.

optimum solution within reasonable computational cost and time. The problem in hand is carried out for the best combination of the design variables to find the global optimum solution with no limits on the execution time. The algorithms are repeated until no search direction can be found that will improve objective function without violating the constraints. Thus, the algorithms are performed for complete search to get the best possible solution of the design. See Fig. 3. So, the comparison mainly focuses on the empirical solution. Figs. 4 and 5 give plots of the design spaces for weight of assembly system when two ball bearings R_7(numbered 1) and R_{13} (numbered 2)-obtained as optimum selection- versus design constraints, g_j. The plots show how the objective function varies for different design variables by visualizing the design space. The optimal design solution which satisfies the inequality constraints is marked on the plots. Since the algorithms conduct a search through search space of potential solutions to the problem, similar behavior of the algorithms is observed in the case of finding the optimum solution. Comparing the performance of the DE with the others algorithms for the problem in hand, the DE was more effective in obtaining better solutions, which are more stable with relatively smaller standard deviations, and higher success rates. Also, the DE has a

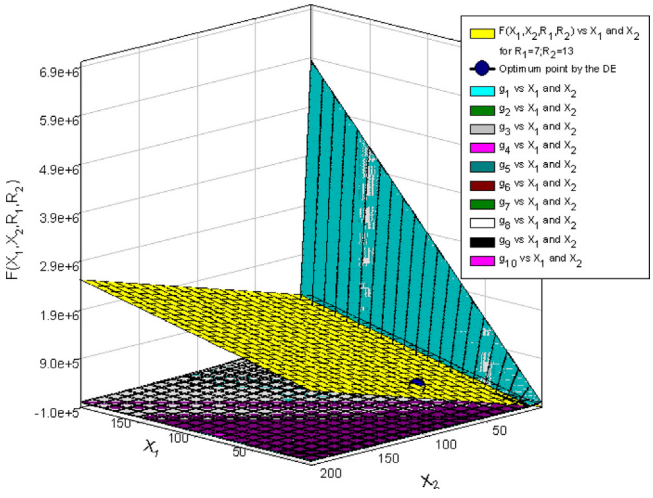

Fig. 5. The lengths x_1 and x_2 for the weight of the ball bearing pivot link system versus the design constraints by the DE.

Table 2
Comparison of the best overall solution found for assembly system by the DE, the SA, the GA, and the ES.

	DE	SA	GA	ES
Length (mm), x_1	30	30	35.5	35.5
Length (mm),x_2	75	79.99	65.03	68.84
Ball bearing numbered 1, R_i	7	7	14	14
Ball bearing numbered 2, R_j	13	13	6	4
Weight of the assembly (gr)	634094.0	671616.4	729445.5	728213.8

few parameters that they have a great impact on the performance of the algorithm from aspects of optimal value and convergence rate.

Table 2 shows a comparison of the best overall solution found for the weight of assembly system by the DE, the SA, the GA, and the ES. The ball bearings link system assembly weight of 634,099 gr was obtained using the DE while 671,616 gr, 728213.8 *gr*, and 729445.5 gr were obtained using the SA, the ES, and the GA respectively. It can be seen in Table 2 that the SA, the GA, and the ES give good approximation to the global optimum but not the exact solution. So, better convergence reliability is obtained with the DE. The assembly weight of 634,099 gr was found for the ball bearing numbered 1 (R_7) with the length $x_1 = 30$ mm, diameter $d_1 = 30$ mm, diameter $D_1 = 90$ mm, width $b_1 = 23$ mm, dynamic load capacity $C_1 = 43{,}600$ N, and mass $m_1 = 740$ gr while the length $x_2 = 35$ mm, diameter $d_2 = 35$ mm, diameter $D_2 = 80$ mm, width $b_2 = 21$ mm, dynamic load capacity $C_2 = 33{,}200$ N, and mass $m_2 = 460$ gr for the ball bearing numbered 2 (R_{13}).

6. Conclusion

In this study, nature inspired algorithms, the Differential Evaluation (DE), and the Simulated Annealing (SA), are employed to find the minimum weight of the ball bearings link system. The ball bearings link system is used extensively in many machinery applications. Among mechanical systems, designers pay great attention to the ball bearing link system because of its significant industrial importance. Nature inspired algorithms have been implemented successfully. They are very efficient and can be used to find out the global optimum solution with high probability. It is observed that the DE and the SA can be easily used to handling mixed design variables and constraints imposed on design optimization of mechanical systems. The DE and the SA are population based optimization algorithms which make them less prone to get trapped in local optima. It can be concluded that the DE and the SA are proven to be robust and have demonstrated their capability to produce an efficient solution to the problem. Beside, the comparison confirms the effectiveness and the superiority of the DE over the others algorithms in terms of solution quality.

References

[1] A.A. Abou El Ela, M.A. Abido, S.R. Spea, Optimal power flow using differential evolution algorithm, Electr. Power Syst. Res. 80 (7) (2010) 878–885.
[2] A. Corana, M. Marchesi, C. Martini, S. Ridella, Minimizing multimodal function of continuous variables with the simulated annealing algorithm, ACM Trans. Math. Software 13 (11) (1992) 87–100.
[3] S. Das, A. Abraham, A. Konar, Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives, Stud. Comput. Intell. 116 (2008) 1–38.
[4] H. Fan, J. Liu, J. Lampinen, Some improvement to the mutation donor of differential evolution, Int. J. Comput. Aided Eng. Software 27 (2) (2010) 225–242.
[5] J.A. Hernandez, J.D. Ospina, A multi dynamics algorithm for global optimization, Math. Comput. Modell. 52 (2010) 1271–1278.

[6] A. Homaifar, C.X. Qi, S.H. Lai, Constrained optimization via genetic algorithms, Simulation 62 (1994) 242–254.
[7] B.K. Kannan, S.N. Kramer, An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, ASME J. Mech. Des. 116 (1994) 318–320.
[8] S. Kirkpatric, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[9] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (1953) 1087–1090.
[10] M. Miki, T. Hiroyasu, O. Keiko, Simulated annealing with advanced adaptive neighborhood, J. Inf. Process. Soc. Jpn. 44 (1) (2003) 1–6.
[11] M. Miki, S. Hiwa, T. Hiroyasu, Simulated Annealing Using Adaptive Search Vector, IEEE, 2006.
[12] L.G. Moreau, P. Lafon, A comparison of evolutionary algorithms for mechanical design components, Eng. Optim. 34 (2002) 307–322.
[13] A.D. Olds, Interplanetary Trajectory Optimization with Differential Evolution (MSc. Thesis), University of Missouri-Columbia, 2005.
[14] A.R. Sa Angela, A.O. Andrade, A.B. Soares, S.J. Nasuto, Exploration vs. Exploitation in Differential Evolution, Proceedings of the 2008 Convention on Artificial Intelligence and Simulation of Behaviour AISB, 2008.
[15] H. Saruhan, Minimization of power loss in tilting-pad journal bearings via an evolutionary algorithm, Ind. Lubr. Tribol. 62 (3) (2010) 144–149.
[16] H. Saruhan, Pivoted-pad journal bearings lubrication design, Ind. Lubr. Tribol. 63 (2) (2011) 119–126.
[17] R. Storn, K. Price, Differential Evolution – a Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces, Technical Report TR-95–012, 1995.
[18] D. Zaharie, Influence of crossover on the behavior of differential evolution algorithms, Appl. Soft Comput. 9 (2009) 1126–1138.
[19] A.R. Yıldız, A novel hybrid immune algorithm for global optimization in design and manufacturing, Rob. Comput. Integr. Manuf. 25 (2009) 261–270.
[20] A.R. Yıldız, Hybrid Taguchi-differential evolution algorithm for optimization of multi-pass turning operations, Appl. Soft Comput. 13 (2013) 1433–1439.