British Journal of Mathematics & Computer Science
7(6): 466-480, 2015, Article no.BJMCS.2015.139
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

Hybrid Genetic Algorithm for Constrained Nonlinear Optimization Problems

S. M. Nasr[1*], M. A. El-Shorbagy[1], I. M. El-Desoky[1], Z. M. Hendawy[1] and A. A. Mousa[1,2]

[1]Department of Basic Engineering Science, Faculty of Engineering, Shebin El-Kom, Menoufia University, Egypt.
[2]Department of Mathematics, Faculty of Sciences, Taif University, Saudi Arabia.

Article Information

DOI: 10.9734/BJMCS/2015/16193
Editor(s):
(1) Doina Bein, Applied Research Laboratory, Pennsylvania State University, USA.
Reviewers:
(1) José Luis López-Bonilla, Department of Mathematics, National Polytechnic Institute, Mexico City, Mexico.
(2) Anonymous, China.
Complete Peer review History: http://www.sciencedomain.org/review-history.php?iid=937&id=6&aid=8332

Original Research Article

Received: 14 January 2015
Accepted: 18 February 2015
Published: 03 March 2015

Abstract

In this paper we present a hybrid optimization algorithm for solving constrained nonlinear optimization problems. The hybrid algorithm is a combination between one of the intelligence techniques (genetic algorithm) and chaos theory to enhance the performance and to reach the optimal solution. The proposed algorithm is operates in two phases: in the first one, genetic algorithm is implemented to solve nonlinear optimization problem. Then, in the second phase, local search referred to chaos theory is introduced to improve the solution quality and find the optimal solution. The results of numerical studies have been demonstrated the superiority of the proposed approach to finding the global optimal solution.

Keywords: Constrained nonlinear optimization problems; optimization algorithm; genetic algorithm; chaos theory.

1 Introduction

Optimization problems, especially constrained nonlinear optimization problems, are very important and frequently appear in the real world, such as structural optimization, engineering design, very-large-scale integration design, economics, allocation and location problems [1,2].

**Corresponding author: sarah.nasr.eid@gmail.com;*

Unfortunately, there is no known method of determining the global maximum (or minimum) to the general constrained nonlinear optimization problem. The algorithms for constrained nonlinear optimization problems are classified to an indirect and direct methods. All of these methods are called traditional optimization techniques, which are local in scope, depending on the existence of derivatives, and they are insufficiently robust in discontinuous, vast multimodal, and noisy search spaces [3].

Some optimization methods that are conceptually different from the traditional techniques have been appeared labeled as modern or non-traditional optimization techniques and are emerging as popular methods for the solution of complex engineering problems. These methods are based on certain characteristics and behavior of biological, molecular, swarm of insects, and neurobiological systems. Furthermore, non-traditional optimization techniques overcome difficulties and limitations of traditional techniques and are less susceptible to getting 'stuck' at local optimal. In addition they require fewer parameters without requiring the objective function to be derivable or even continuous [4].

Among the existing non-traditional techniques, well-known algorithms are Genetic algorithms (GA) [5,6], Simulated annealing (SA) [7,8], Particle swarm optimization (PSO) [9,10], Ant colony optimization (ACO) [11-13], Fuzzy optimization [14,15] and Neural-network-based methods [16,17], etc.

Genetic Algorithm (GA) is one of this non-traditional algorithms and is presented as an efficient global method for constrained nonlinear optimization problems. GAs are well suited for solving such problems and it enjoys an increasing interest in the optimization community and many industrial applications. For instance, Elsayed et al. [18] presented a new genetic algorithm for solving optimization problems, and successfully solving a set of constrained optimization problems. Furthermore, genetic algorithms was applied for optimal design of a welded beam in [19]. In addition, genetic algorithms concepts and design for optimization of process controllers is proposed in [20].

The fundamental of genetic algorithms is based on ‘Random’. This randomness may make premature convergence to reach to the global optimal solution taking a large number of iterations to reach to it and the optimization may get stuck at a local optimum. New researchers introduced improved methods based on the hybridizing algorithms with genetic algorithms to improve its results trying to find global optimal solution. For instance, Tsoulos [21] introduced a heuristic modified method based on the genetic algorithm for solving constrained optimization problems. Juan and Ping [22] optimized the fuzzy rule base with combination of the GA and Ant Colony; their results show that the hybrid method can be more useful than the basic GA. Additionally, Sun and Tian [23] developed an efficient hybrid method for image classification with PSO and GA; where the authors used features of fast convergence of PSO and diversity of GA to improve their method.

Chaos theory was initially described by Henon [24] and was summarized by Lorenz [25]. It is study in mathematics that has applications in several discipline: meteorology, sociology, physics, engineering, economics, biology, and philosophy. Chaos is a common non-linear phenomenon in nature, which fully reflects the complexity of the system that will be useful in optimization. Chaotic maps (including logistic maps) can easily be implemented and avoid entrapment in local optimal [26-30]. The inherent characteristics of chaos can enhance genetic solution by enabling it to escape from local solutions and increase the convergence to reach to the global solution.

Many researchers proposed a combination between chaos theory and optimization algorithms to improve the solution quality. For instance, in [31] the authors presented hybrid chaos-particle swarm optimization algorithm for the vehicle routing problem with time window. While, in [32] chaotic genetic algorithm based on Lorenz Chaotic System for optimization problems is proposed.

In this paper, we present a hybrid optimization algorithm for solving constrained nonlinear optimization problems. The hybrid optimization algorithm is a combination between genetic algorithm and chaos theory. The proposed algorithm is operates in two phases: in the first one, genetic algorithm is implemented to solve constrained nonlinear optimization problem. Then, in the second phase, local search referred to chaos theory is introduced to improve the solution quality and find the optimal solution. The results of various numerical studies have been demonstrated the superiority of the proposed approach to finding the global optimal solution [33].

The paper will be as follows. In section 2, we will define constrained nonlinear optimization problems. In sections 3, genetic algorithm is briefly introduced. In section 4, chaotic maps is briefly introduced. Proposed approach is proposed and explained in detail in section 5. Numerical studies and discussions are presented in section 6. Finally, we conclude the paper in section 7.

2 Constrained Nonlinear Optimization Problems

Constrained nonlinear optimization problem can be formulated as follows [2]:

$$\begin{aligned} \text{NLPP: Min} \quad & f(x), \\ \text{s.t.} \quad & g_j(x) \le 0, \quad \textit{for} \quad j = 1,2,...,p, \\ & h_i(x) = 0, \quad \textit{for} \quad i = 1,2,...,q; \\ & x_n^l \le x_n \le x_n^u, \quad n = 1,2,...,N, \end{aligned} \tag{1}$$

where $f, g_1,...,g_p, h_1,...,h_q$ are functions defined on R_n, $\mathcal{X}$ is a subset of R_n, and x represents a vector of decision variables which take real values, and each decision variable x_n is constrained by its lower and upper boundaries $\left[x_n^l, x_n^u\right]$; N is the total number of decision variables x_n. The above problem must be solved for the values of the variables $x_1,...,x_N$ that satisfy the restrictions and mean while minimize the function *f*. The function *f* is usually called the objective function, or the criterion function. Each of the constraints $g_j(x) \le 0$ for $j = 1,2,...,p$ is called an inequality constraint, and each of the constraints $h_i(x) = 0$ for $i = 1,2,...,q$ is called an equality constrain. If some of the constraints or the objective function is nonlinear, the optimization problem called nonlinear optimization problem.

3 The Basics of Genetic Algorithm

The genetic algorithms were introduced by Holland in 1970s as optimization approaches to find a global or near-global optimal solution [5]. Genetic algorithms start with a set of potential solutions (chromosomes). Next, genetic search operators such as selection, mutation and crossover are then applied one after another to obtain a new generation of chromosomes in which the expected quality over all the chromosomes is better than that of the initial generation [20]. This process is repeated until the termination criterion is met, and the best chromosome of the last generation is reported as the final solution [34]. Fig. 1 shows the pseudo code of the general GA algorithm.

Generate an initial population Check chromosome in constrains and repair out constrain values Evaluate chromosome in the objective function Do: Children population [Select parents from population and recombine parents (Crossover and mutation operators)] Evaluate children in the objective function Construct best population of parents and children population While satisfactory solution has been found

Fig. 1. The pseudo code of the general GA algorithm

4 Chaotic Maps

Chaos theory studies the behavior of systems that follow deterministic laws but appear random and unpredictable. Chaos being radically different from statistical randomness, especially the inherent ability to search the space of interest efficiently, could improve the performance of optimization procedure. It could be introduced into the optimization strategy to accelerate the optimum seeking operation and find the global optimal solution [35]. In this section, one dimensional and non-invertible maps are used to build up chaotic sequences. Here we offer some well-known chaotic maps found in the literature.

- **Chebyshev map**
  Chebyshev map is represented as [36]:

$$x_{t+1} = \cos\left(t \cos^{-1}(x_t)\right); \tag{2}$$

- **Circle map**
  Circle map is defined as the following representative equation [37]:

$$x_{t+1} = x_t + b - (a - 2\pi)\sin(2\pi x_t) \ \mathrm{mod}(1); \tag{3}$$

  where $a = 0.5$ and $b = 0.2$.

- **Gauss/mouse map**
  The Gauss map consists of two sequential parts defined as [38]:

$$x_{t+1} = \begin{cases} 0 & \text{if} \quad x_t = 0 \\ 1/x_t & \text{else} \quad \mathrm{mod}(1) \end{cases}; \tag{4}$$

  where $\frac{1}{x_t}\mathrm{mod}(1) = \frac{1}{x_t} - \left\lfloor \frac{1}{x_t} \right\rfloor$.

- **Intermittency map**
  The intermittency map [39] is formed with two iterative equations and represented as:

$$x_{t+1} = \begin{cases} \varepsilon + x_t + c x_t^n & \text{if} \quad 0 < x_t \le p \\ \dfrac{x_t - p}{1 - p} & \text{elseif} \quad p < x_t < 1 \end{cases} ; \tag{5}$$

where $c = \dfrac{1-\varepsilon-p}{p^2}$, $n = 2.0$ and ε is very close to zero.

- **Iterative map**
  The iterative chaotic map with infinite collapses [40] is defined with the following as:

$$x_{t+1} = \sin\left(\frac{a\pi}{x_t}\right); \tag{6}$$

where $a \in (0,1)$.

- **Liebovitch map**
  The proposed chaotic map [36] can be defined as:

$$x_{t+1} = \begin{cases} \alpha x_t & 0 < x_t \le p_1 \\ \dfrac{p_2 - x_t}{p_2 - p_1} & p_1 < x_t \le p_2 \\ 1 - \beta(1 - x_t) & p_2 < x_t \le 1 \end{cases} ; \tag{7}$$

where $\alpha = \dfrac{p_2\left(1-(p_2-p_1)\right)}{p_1}$ and $\beta = \dfrac{\left((p_2-1)-p_1(p_2-p_1)\right)}{p_2-1}$.

- **Logistic map**
  Logistic map [41] demonstrates how complex behavior arises from a simple deterministic system without the need of any random sequence. It is based on a simple polynomial equation which describes the dynamics of biological population [42].

$$x_{t+1} = c x_t (1 - x_t); \tag{8}$$

where $x_0 \in (0,\ 1)$, $x_0 \notin \{0.0,\ 0.25,\ 0.50,\ 0.75,\ 1.0\}$ and when $c = 4.0$ a chaotic sequence is generated by the Logistic map.

- **Piecewise map**
  Piecewise map [40] can be formulated as follows:

$$x_{t+1} = \begin{cases} \dfrac{x_t}{p} & 0 < x_t < p \\ \dfrac{x_t - p}{0.5 - p} & p \le x_t < 0.5 \\ \dfrac{(1 - p - x_t)}{0.5 - p} & 0.5 \le x_t < 1 - p \\ \dfrac{(1 - x_t)}{p} & 1 - p < x_t < 1 \end{cases}; \quad (9)$$

where $p \in (0, 0.5)$ and $x \in (0, 1)$.

- **Sine map**
  Sine map [43] can be described as:

$$x_{t+1} = \frac{a}{4}\sin(\pi x_t); \quad (10)$$

where $0 < a < 4$.

- **Singer map**
  One dimensional chaotic Singer map [44] is formulated as:

$$x_{t+1} = \mu\left(7.86x_t - 23.31x_t^2 + 28.75x_t^3 - 13.302875x_t^4\right); \quad (11)$$

where $\mu \in (0.9, 1.08)$.

- **Sinusoidal map**
  Sinusoidal map [42] is generated as the following equation:

$$x_{t+1} = ax_t^2\sin(\pi x_t); \quad (12)$$

where $a = 2.3$.

- **Tent map**
  Tent map [45] is defined by the following iterative equation:

$$x_{t+1} = \begin{cases} x_t / 0.07 & x_t < 0.7 \\ \dfrac{10}{3}(1.0 - x_t) & x_t \ge 0.7 \end{cases} \quad (13)$$

5 The Proposed Algorithm

In this section, we describe the proposed approach which is a combination between GA and chaos theory for solving constrained nonlinear optimization problems. The proposed algorithm is operates in two phases: in the first one, genetic algorithm is implemented to solve nonlinear optimization problem. Then, in the second phase, local search referred to chaos theory is introduced to improve the solution quality and find the optimal solution. The details of the proposed algorithm is described as follows:

Phase I: GA

Step 1. Initial Population: The population vectors in the first generation are initialized randomly satisfying the search space *S* (the lower and upper bounds for each variable), using the following equation.

$$\text{Each individual position}\Big|_i = \mathrm{L} + (\mathrm{U} - \mathrm{L}) \times \text{rand}; \tag{14}$$

where $i = 1,\ldots,N_{pop}$, and N_{pop} is the size of the population; *L* is the lower bound; *U* is the upper bound, and rand is random numbers uniformly distributed within the range [0,1].

Step 2. Initial reference point: The algorithm needs at least one feasible reference point (i.e., feasible point) to enter the evolution process (i.e., complete the algorithm procedure), the reader is referred to Osman et al. [46].

Step 3. Repairing: Repair the infeasible individuals of the population to be feasible. The idea of this technique is to separate any feasible individuals in a population from those that are infeasible by repairing infeasible individuals. This approach co-evolves the population of infeasible individuals until they become feasible, the reader is referred to [34].

Step 4. Evaluation: Evaluate the desired optimization fitness function in n variables for each individual.

Step 5. Create a new population: Creating a new population from the current generation by using the three operators (Ranking, selection, crossover, and mutation).

- **Ranking:** Ranks individuals according to their fitness value, and returns a column vector containing the corresponding individual fitness value, in order to establish later the probabilities of survival that are necessary for the selection process [20].
- **Selection:** There are several techniques of selection. The commonly used techniques for selection of individuals are roulette wheel selection, rank selection, steady state selection, stochastic universal sampling, etc.. Here we will use Stochastic Universal Sampling (SUS) [47] where, the most important concern in a stochastic selection is to prevent loss of population diversity due to its stochastic aspect.
- **Crossover:** In GAs, crossover is used to vary individuals from one generation to the next; where it combines two individuals (parents) to produce a new individuals (offspring) with probability (Pc). There are several techniques of crossover, one-point crossover, two-point crossover, cut and splice, uniform crossover and half uniform crossover, etc. [34,48]. Here we will use One-point crossover involving splitting two individuals and then combining one part of one with the other pair. This method performs recombination between pairs of individuals and returns the new individuals after mating, and gives offspring the best possible combination of the characteristics of their parents.
- **Mutation:** Premature convergence is a critical problem in most optimization techniques, which occurs when highly fit parent individuals in the population breed many similar offspring in early evolution time. Mutation is used to maintain genetic diversity from one generation of a population to the next. In addition, Mutation is an operator to change elements in a string which is generated after crossover operator [48]. In this study, we will use real valued mutation; which means that randomly created values are added to the variables with a low probability (Pm).

Step 6. Migration: In this step, a migration of the new offspring with the old population to create the new population by taking the best individuals of parents and offspring population [48,49].

Step 7. Termination test: The algorithm is terminated either when the maximum number of generations is achieved, or when the individuals of the population coverages, convergence occur when all individuals positions in the population are identical. In this case, crossover will have no further effect. otherwise, return to step 3.

Phase 2: Chaos search

Step 1. Determine variance range of chaos search boundary: The range of chaotic local search $[a,b]$ is determined by $x_i^* - \varepsilon < a_i$, $x_i^* + \varepsilon > b_i$; where $x_i^* = (x_1^*, x_2^*, ..., x_n^*)$ is the genetic result, and ε is specified radius of chaos search.

Step 2. Generate chaos variables: In this step, a chaotic random numbers z^k is generated by the Logistic map (Equation 8) which is used extensively by using the following equation.

$$z^{k+1} = \mu z^k \left(1 - z^k\right),\ z^0 \in (0,1),\ z^0 \notin \{0.0,\ 0.25,\ 0.50,\ 0.75,\ 1.0\},\ k = 1,2,.... \quad (15)$$

Step 3. Mapping chaos variable into the variance range: Chaos variable z^k is mapped into the variance range of optimization valuable $[a,b]$ by:

$$x_i = x_i^* - \varepsilon + 2\varepsilon z^k \quad \forall i = 1,...,n \quad (16)$$

Step 4. Update the best value: Set chaotic iteration number as $k = 1 \rightarrow$ Do $x_i^k = x_i^* - \varepsilon + 2\varepsilon z \quad \forall i = 1,...,n$.

If $f\left(x^k\right) < f\left(x^*\right)$ then set $x^* = x^k$.

Else if $f\left(x^k\right) \geq f\left(x^*\right)$ then give up the kth iterated.

Result x^* .

Loop runs until $f\left(x^*\right)$ is not improved after *k* searches.

Step 5. Update the boundary value: Choose the boundary value $[a,b]$ of the new optimal point x_i^* as the new chaos search range. Chaos variable is mapped into the new search range by (16), then go to step 2.

Step 6. Stopping Chaos search: If $f\left(x^*\right)$ is not improved for all *k* searches, stop Chaos search process and put out x^* as the best solution. The flow chart of the proposed algorithm showing in Fig. 2.

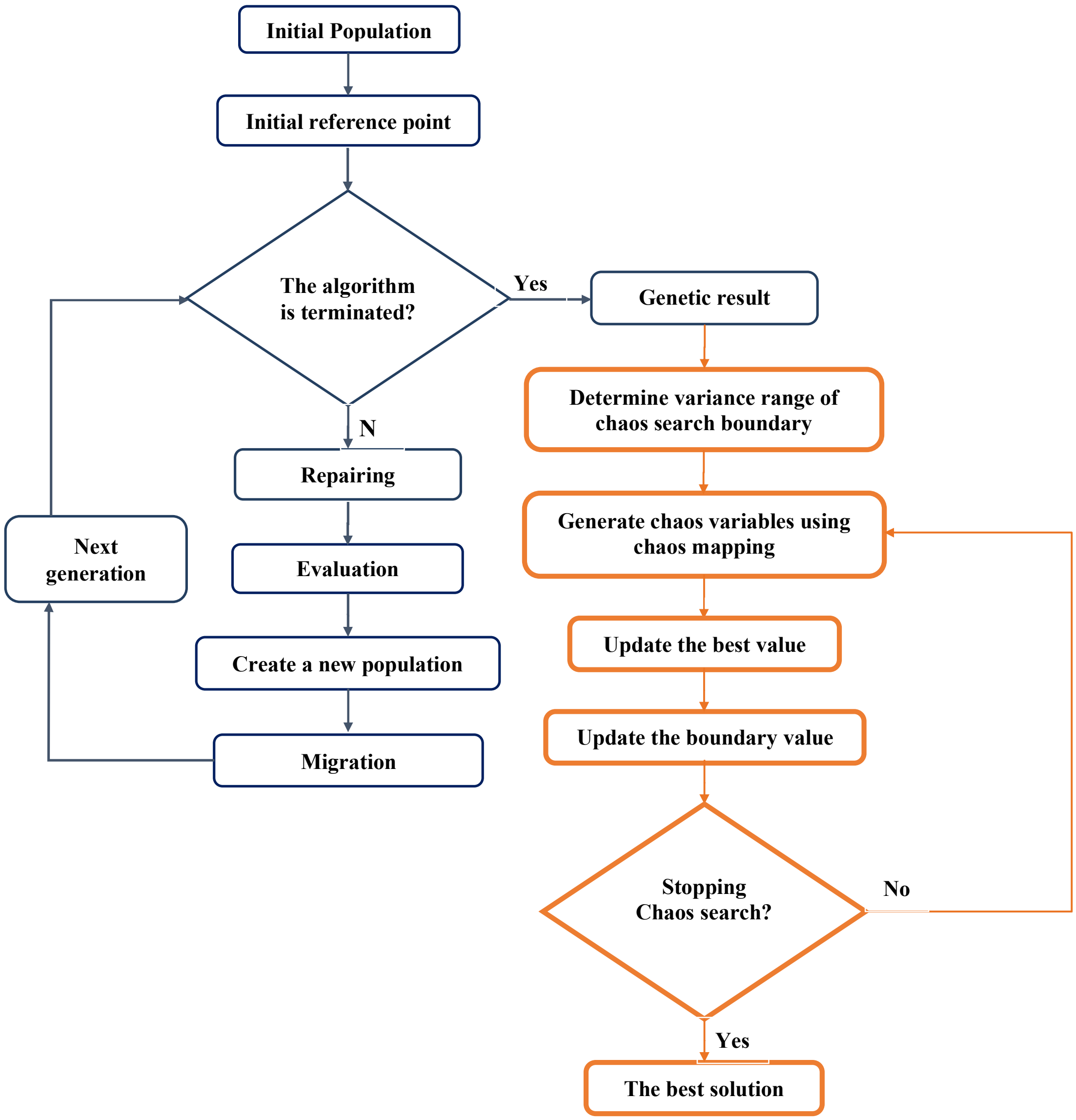

Fig. 2. The flow chart of the proposed algorithm

6 Numerical Studies

For evaluating the performance for global optimization, the proposed algorithm is tested by a set of constrained benchmark problems taken from the literature [33]. Table 1 lists the variable bounds, objective function and constraints for all these problems [33]. Our study consists of the comparison of performance with other optimization algorithms to demonstrate the efficiency and robustness of the proposed algorithm.

The proposed algorithm is coded in MATLAB 6.0 and the simulations have been executed on an Intel core (TM)i7-4500cpu 1.8GHZ 2.4 GHz processor. As any non-traditional optimization algorithms, the proposed algorithm, involves a number of parameters that affect the performance of algorithm. The parameters adopted in the implementation of the proposed algorithm are listed in Table 2.

6.1 The Results

The proposed algorithm and augmented Lagrange particle swarm optimization (ALPSO) [33] are applied to the set of Constrained benchmark problems. In [33] Sedlaczek and Eberhard applied their method making 30 independent runs. Table 3 illustrates the comparison between the optimal solution, the proposed algorithm result, and the best value obtained by ALPSO [33].

As a result from Table 3, for the problems (P1, P2, P3, P5 and P6), the proposed algorithm is found the optimal solution and near to the optimal solution for the problem P4. On the other hand, ALPSO is found the optimal solution for the problems (P3, P5 and P6), near to the optimal solution for the problems (P1 and P2) and smaller than the optimal solution for the problem P3. So, the proposed algorithm more converges to the optimal solution and exhibits a superior performance in comparison to ALPSO (i.e. the proposed algorithm found the better solutions than ALPSO on average).

Fig. 3 shows the convergence rate of the proposed approach on the several constrained benchmark problems. From these figures we can see that the proposed algorithm converge more quickly to the optimal solution in particular, in the early generation (iteration).

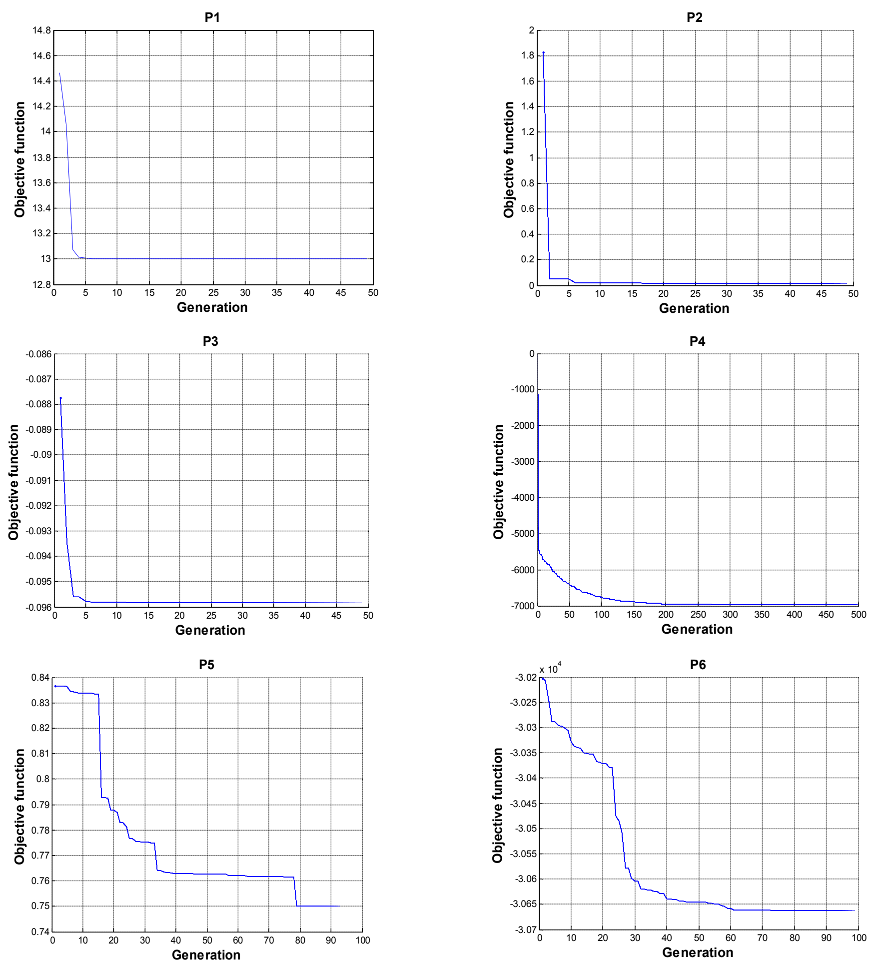

Fig. 3. The convergence rate of the proposed approach on different constrained benchmark problems

Table 1. Constrained benchmark problems

Problem	Variable bounds	Objective function $f(x)$ and constraints $C(x)$
P1	$x_1 \in [-10,10]$ $x_2 \in [-10,10]$	$f_1(x) = x_1^2 + x_2^2$ $C_1(x) = x_1 - 3 = 0$ $C_2(x) = 2 - x_2 \le 0$
P2	$x_1 \in [-10,10]$ $x_2 \in [-10,10]$	$f_1(x) = \frac{1}{4000}\left(x_1^2 + x_2^2\right)^2 - \cos(\frac{x_1}{\sqrt{1}})\cos(\frac{x_2}{\sqrt{2}}) + 1$ $C_1(x) = x_1 - 3 = 0$ $C_2(x) = 2 - x_2 \le 0$
P3	$x_1 \in [0.1,10]$ $x_2 \in [0,10]$	$f_1(x) = \frac{-\sin(2\pi x_1)^3 \sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$ $C_1(x) = x_1^2 - x_2 + 1 \le 0$ $C2(x) = 1 - x_1 + (x_2 - 4)^2 \le 0$
P4	$x_1 \in [13,100]$ $x_2 \in [0,100]$	$f_1(x) = (x_1 - 10)^3 + (x_2 - 20)^3$ $C_1(x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0$ $C2(x) = (x_1 - 6) + (x_2 - 5)^2 - 82.81 \le 0$
P5	$x_1 \in [-1,1]$ $x_2 \in [-1,1]$	$f_1(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$ $C_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.006262x_1x_4 - 0.0022053x_3x_5 - 92 \le 0$ $C_2(x) = -85.334407 - 0.0056858x_2x_5 - 0.006262x_1x_4 + 0.0022053x_3x_5 \le 0$ $C_3(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \le 0$ $C_4(x) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \le 0$ $C_5(x) = 9.300961 + 0.0047026x_3x_5 + 0 - 0012547x_1x_3 + 0.0019085x_3x_4 - 25 \le 0$ $C_6(x) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \le 0$
P6	$x_1 \in [78,100]$ $x_2 \in [33,45]$ $x_3 \in [27,45]$ $x_4 \in [27,45]$ $x_5 \in [27,45]$	$f_1(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$ $C_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.006262x_1x_4 - 0.0022053x_3x_5 - 92 \le 0$ $C_2(x) = -85.334407 - 0.0056858x_2x_5 - 0.006262x_1x_4 + 0.0022053x_3x_5 \le 0$ $C_3(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \le 0$ $C_4(x) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \le 0$ $C_5(x) = 9.300961 + 0.0047026x_3x_5 + 0 - 0012547x_1x_3 + 0.0019085x_3x_4 - 25 \le 0$ $C_6(x) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \le 0$

Table 2. The proposed algorithm parameters

Generation gap	0.9
Crossover rate	0.9
Mutation rate	0.7
Selection operator	Stochastic universal sampling
Crossover operator	Single point
Mutation operator	Real-value
GA generation	500-1000
Chaos generation	10000
Specified neighborhood radius	1E-6

Table 3. The comparison between the optimal solution, the best value of ALPSO, and the proposed algorithm result

Problem	Optimal solution	Best value of ALPSO	The proposed algorithm result
P1	13.0000	12.9995	13.0000
P2	0.01721	0.01719	0.01721
P3	-0.09583	-0.09583	-0.09583
P4	-6961.81	-6963.57	-6961.804
P5	0.75000	0.75000	0.750000
P6	-30665.5	-30665.5	-30665.5

7 Conclusion

In this paper we present a hybrid optimization algorithm for solving constrained nonlinear optimization problems. The proposed algorithm is a combination between one of the intelligence techniques (genetic algorithm) and chaos theory and it is operates in two phases: In the first one, genetic algorithm is implemented to solve constrained nonlinear optimization problems. Then, in the second phase, local search referred to chaos theory is introduced to find the optimal solution. The results of various numerical studies have been demonstrated the superiority of the proposed approach to finding the optimal solution.

A careful observation will reveal the following benefits of the proposed optimization algorithm:

1. It integrates the powerful global searching capability of the GA with the powerful local searching capability of the Chaos search.
2. Unlike classical techniques, the proposed algorithm search from a population of points, not single point. Therefore, it can provide a globally optimal solution.
3. It uses only the objective function information, not derivatives or other auxiliary knowledge. Therefore it can deal with the non-smooth, non-continues and non-differentiable functions which are actually existed in practical optimization problems.
4. It found better solutions than the other methods that reported in the literature.
5. It was demonstrated to be extremely effective and efficient at locating optimal solutions.
6. Due to simplicity of the proposed algorithm procedures, it can using to handle complex problems of realistic dimensions.

In our future works, the following will be researched:

a) Solving larger scale examples to demonstrate the efficiency of the proposed algorithm.
b) Updating the proposed algorithm to solve the multi-objective optimization problems.

Using another chaotic maps to accelerate the convergence property of the proposed algorithm and improve the solution quality.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Michae BB. Nonlinear optimization with engineering applications. Springer Optimization and Its Applications. 2008;19.

[2] Rao SS. Engineering Optimization: Theory and Practice, 3rd ed., Wiley, NY, USA; 2009.

[3] Michalewicz Z. Evolutionary computation techniques for nonlinear programming problems. International Transactions in Operational Research. 1994;1(2):223–240.

[4] Onwubolu GC, Babu BV. New optimization techniques in engineering. Springer Science & Business Media. 2004;141.

[5] Gallagher K, Sambridge M. Genetic algorithms: A powerful tool for large-scale nonlinear optimization problems. Compute and Geosciences. 1994;20(7\8):1229-1236. Geosciences.

[6] Neves N, Nguyen AT, Torres EL. A study of nonlinear optimization problems using a distributed genetic algorithm. Proceedings of the 1996 International Conference onon Parallel Processing. 1996;2:29-36.

[7] Alrefaei MH, Diabat AH. A simulated annealing technique for multi-objective simulation optimization. Applied Mathematics and Computation. 2009;215:3029–3035.

[8] Brooks SP, Morgan BJT. Optimization using simulated annealing. The Statistician. 1995;44(2):241-257.

[9] Mousa AA, El-Shorbagy MA, Abd-El-Wahed WF. Local search based hybrid particle swarm optimization algorithm for multiobjective optimization. Swarm and Evolutionary Computation. 2012;3:1–14.

[10] Yang X, Yuan J, Mao H. A modified particle swarm optimizer with dynamic adaptation. Applied Mathematics and Computation. 2007;189:1205–1213.

[11] Mousa AA, AbdEl-Wahed WF, Rizk-Allah RM. A hybrid ant colony optimization approach based local search scheme for multiobjective design optimizations, Electric Power Systems Research. 2011;81:1014–1023,.

[12] Dréo J, Siarry P. An ant colony algorithm aimed at dynamic continuous optimization. Appl. Math. Comput. 2006;181(1):457-467.

[13] Dorigo M, Stützle T. Ant Colony Optimization, MIT Press, London; 2004.

[14] Oltean G. Fuzzy Techniques in Optimization-Based Analog Design, 9th WSEAS International Conference on fuzzy systems (FS'08) which was held in Sofia, Bulgaria. 2008;178-191.

[15] Fares M, Kaminska B. A Fuzzy Nonlinear Programming Approach to Analog Circuit Design. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 1995;14(17):785-793.

[16] Hopfield JJ, Tank DW. "Neural" computation of decisions in optimization problem. Biological Cybernetics. 1985;52:141-152.

[17] Zhou A. A Genetic-Algorithm-based neural network approach for short-term traffic flow forecasting. Advances in Neural Networks. 2005;3498:965-969.

[18] Elsayed SM, Sarker RA, Essam DL. A new genetic algorithm for solving optimization problems. Engineering Applications of Artificial Intelligence. 2014;27:57–69.

[19] Deb K. Optimal design of a welded beam via genetic algorithms, AIAA Journal. 1991;29(11):2013-2015.

[20] Malhotra R, Singh N, Singh Y. Genetic algorithms: Concepts, design for optimization of process controllers. Computer and Information Science. 2011;4(2):39-54.

[21] Tsoulosl G. Solving constrained optimization problems using a novel genetic algorithm. Applied Mathematics and Computation. 2009;208(1):273-283.

[22] Juan W, Ping W. Optimization of Fuzzy rule based on adaptive genetic algorithm and ant colony algorithm. International Conference on Computational and Information Sciences (ICCIS). 2010;359-362.

[23] Sun Fj, Tian Y. Transmission line image segmentation based GA and PSO Hybrid Algorithm, 2010 International Conference on Computational and Information Sciences (ICCIS). 2010;677-680.

[24] Hénon M. A two-dimensional mapping with a strange attractor, Commun. Math. Phys. 1976;50:69-77.

[25] Lorenz E. The essence of chaos. University of Washington Press; 1996.

[26] Liu B, Wang L, Jin YH, Tang F, Huang DX. Improved particle swarm optimization combined with chaos, Chaos, Solitons and Fractals. 2005;25:1261–1271.

[27] Xiang T, Liao X, Wong KW. An improved particle swarm optimization algorithm combined with piecewise linear chaotic map. Applied Mathematics and Computation. 2007;190:1637–1645.

[28] Wang L, Zheng DZ, Lin QS. Survey on chaotic optimization methods. Computing Technology and Automation. 2001;20(1):1–5.

[29] Coelho LS, Mariani VC. A novel chaotic particle swarm optimization approach using Hénon map and implicit filtering local search for economic load dispatch, Chaos, Solitons and Fractals. 2009;39(2):510–518.

[30] Chuanwen J, Bompard E. A self-adaptive chaotic particle swarm algorithm for short term hydroelectric system scheduling in deregulated environment. Energy Conversion and Management. 2005;46:2689–2696.

[31] Hu W, Liang H, Peng C, Du B, Hu Q. Hybrid Chaos-Particle Swarm Optimization Algorithm for the Vehicle Routing Problem with Time Window, Entropy. 2013;15:1247-1270.

[32] Jampour M. Chaotic genetic algorithm based on Lorenz Chaotic system for optimization problems. Intelligent Systems and Applications. 2013;5(5):19-24.

[33] Sedlaczek K, Eberhard P. Constrained particle Swarm optimization of mechanical systems, 6^{th} World Congresses of Structural and Multidisciplinary Optimization; 2005.

[34] Abd-El-Wahed WF, Mousa AA, El-Shorbagy MA. Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems. Journal of Computational and Applied Mathematics. 2011;235:1446–1453.

[35] Ammaruekarat P, Meesad P. A chaos search for multi-objective memetic algorithm. International Conference on Information and Electronics Engineering. 2011;6:140-144.

[36] Tavazoei MS, Haeri M. Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. Appl. Math. Comput. 2007;187(10):76–85.

[37] Hilborn RC. Chaos and nonlinear dynamics: An introduction for scientists and engineers, second ed., Oxford Univ. Press, New York; 2004.

[38] He D, He C, Jiang L, Zhu H, Hu G. Chaotic characteristic of a one dimensional iterative map with infinite collapses. IEEE Trans. Circuits Syst. 2001;48(7):900–906.

[39] Erramilli A, Singh RP, Pruthi P. Modeling packet traffic with chaotic maps, Royal Institute of Technology, Stockholm-Kista, Sweden; 1994.

[40] May RM. Simple mathematical models with very complicated dynamics, Nature. 1976;261(4):59–67.

[41] Arora JS, Elwakeil OA, Chahande AI, Hsieh CC. Global optimization methods for engineering application: A review, Struct. Optim. 1995;9:137–159.

[42] Li Y, Deng S, Xiao D. A novel Hash algorithm construction based on chaotic neural network, Neural Comput. Appl. 2011;20:133–141.

[43] Devaney RL. An Introduction to Chaotic Dynamical Systems, Addison-Wesley; 1987.

[44] Peitgen H, Jurgen H, Saupe D. Chaos and Fractals, Springer-Verlag, Berlin, Germany; 1992.

[45] Ott E. Chaos in dynamical systems, Cambridge University Press, UK, Cambridge; 2002.

[46] Osman MS, Abo-Sinna MA, Mousa AA. A solution to the optimal power flow using genetic algorithm. Applied Mathematics and Computation. 2004;155:391–405.

[47] Baker JE. Reducing bias and inefficiency in the selection algorithm, proceedings of the second international conference on genetic algorithms. Morgan Kaufmann Publishers, Inc., San Mateo, CA. 1987;14-21.

[48] Popov A. Genetic algorithm for optimization, TU-Sofia; 2003.

[49] Neves N, Nguyen AT, Torres EL. A study of nonlinear optimization problem using a distributed genetic algorithm, 25th International Conference on Parallel Processing; 1996.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=937&id=6&aid=8332