

1 **Original Research Article**
2 **Hybrid Genetic Algorithm for Constrained**
3 **Nonlinear Optimization Problems**

4 **Author details:**

5 **S.M. Nasr¹, M.A. El-Shorbagy^{1*}, I.M. El_Desoky¹, Z. M. Hendawy¹, A.A.**
6 **Mousa^{1,2}**

7
8 **1Department of Basic Engineering Science, Faculty of Engineering, Shebin**
9 **El-Kom Menoufia University, Egypt.**

10 **2Department of Mathematics, Faculty of Sciences, Taif University, Saudi**
11 **Arabia.**

12 **Communication id:**

13 **sarah nasr**

14 **sarah.nasr.eid@gmail.com**

15 **sarah.nasr.eid@gmail.com**

16
17
18 **ABSTRACT**
19

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.

20
21 **Keywords:** *Constrained Nonlinear Optimization Problems, Optimization Algorithm, Genetic*
22 *Algorithm, Chaos Theory.*

23
24
25 **1. INTRODUCTION**
26

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

31
32 Unfortunately, there is no known method of determining the global maximum (or minimum)
33 to the general constrained nonlinear optimization problem. The algorithms for constrained
34 nonlinear optimization problems are classified to an indirect and direct methods. All of these
35 methods are called traditional optimization techniques, which are local in scope, depending
36 on the existence of derivatives, and they are insufficiently robust in discontinuous, vast
37 multimodal, and noisy search spaces [3].
38

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.

102
103

104 2. CONSTRAINED NONLINEAR OPTIMIZATION PROBLEMS

105
106

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

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

108

109 where $f, g_1, \dots, g_p, h_1, \dots, h_q$ are functions defined on R_n , X is a subset of R_n , and x
110 represents a vector of decision variables which take real values, and each decision variable
111 x_n is constrained by its lower and upper boundaries $[x_n^l, x_n^u]$; N is the total number of
112 decision variables x_n . The above problem must be solved for the values of the variables
113 $x_1, \dots, x_N$ that satisfy the restrictions and mean while minimize the function f . The function f is
114 usually called the objective function, or the criterion function. Each of the constraints
115 $g_j(x) \leq 0$ for $j = 1, 2, \dots, p$ is called an inequality constraint, and each of the constraints
116 $h_i(x) = 0$ for $i = 1, 2, \dots, q$ is called an equality constrain. If some of the constraints or the
117 objective function is nonlinear, the optimization problem called nonlinear optimization
118 problem.

119
120

121 3. THE BASICS OF GENETIC ALGORITHM

122

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

131
132
133
134

```

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]:

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

- Circle map

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

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

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

- Gauss/mouse map

The Gauss map consists of two sequential parts defined as [38]:

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

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

- Intermittency map

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

$$x_{t+1} = \begin{cases} \varepsilon + x_t + cx_t^n & \text{if } 0 < x_t \leq p \\ \frac{x_t - p}{1 - p} & \text{elseif } p < x_t < 1 \end{cases}; \quad (5)$$

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

163 - **Iterative map**

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

$$165 \quad x_{t+1} = \sin\left(\frac{a\pi}{x_t}\right); \quad (6)$$

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

167 - **Liebovitch map**

168 The proposed chaotic map [36] can be defined as:

$$169 \quad x_{t+1} = \begin{cases} \alpha x_t & 0 < x_t \leq p_1 \\ \frac{p_2 - x_t}{p_2 - p_1} & p_1 < x_t \leq p_2 ; \\ 1 - \beta(1 - x_t) & p_2 < x_t \leq 1 \end{cases} \quad (7)$$

170

171 where $\alpha = \frac{p_2(1-(p_2-p_1))}{p_1}$ and $\beta = \frac{((p_2-1)-p_1(p_2-p_1))}{p_2-1}$.

172 - **Logistic map**

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

$$176 \quad x_{t+1} = cx_t(1-x_t); \quad (8)$$

177 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
178 sequence is generated by the Logistic map.

179 - **Piecewise map**

180 Piecewise map [40] can be formulated as follows:

$$181 \quad x_{t+1} = \begin{cases} \frac{x_t}{p} & 0 < x_t < p \\ \frac{x_t - p}{0.5 - p} & p \leq x_t < 0.5 \\ \frac{(1-p-x_t)}{0.5-p} & 0.5 \leq x_t < 1-p \\ \frac{(1-x_t)}{p} & 1-p < x_t < 1 \end{cases}; \quad (9)$$

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

183 - **Sine map**

184 Sine map [43] can be described as:

$$185 \quad x_{t+1} = \frac{a}{4} \sin(\pi x_t); \quad (10)$$

186 where $0 < a < 4$.

187 - **Singer map**

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

$$189 \quad x_{t+1} = \mu(7.86x_t - 23.31x_t^2 + 28.75x_t^3 - 13.302875x_t^4); \quad (11)$$

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

191 - **Sinusoidal map**

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

$$193 \quad x_{t+1} = ax_t^2 \sin(\pi x_t); \quad (12)$$

194 where $a = 2.3$.

195 - **Tent map**

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

$$197 \quad x_{t+1} = \begin{cases} x_t / 0.07 & x_t < 0.7 \\ \frac{10}{3}(1.0 - x_t) & x_t \geq 0.7 \end{cases} \quad (13)$$

198

199

200 5. THE PROPOSED ALGORITHM

201

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

208

209 Phase I: GA

210

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

$$214 \quad \text{Each individual position } |_i = L + (U - L) \times \text{rand}; \quad (14)$$

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

217

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

221

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

226

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

229

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 $x_i^* = (x_1^*, x_2^*, \dots, x_n^*)$ 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.

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

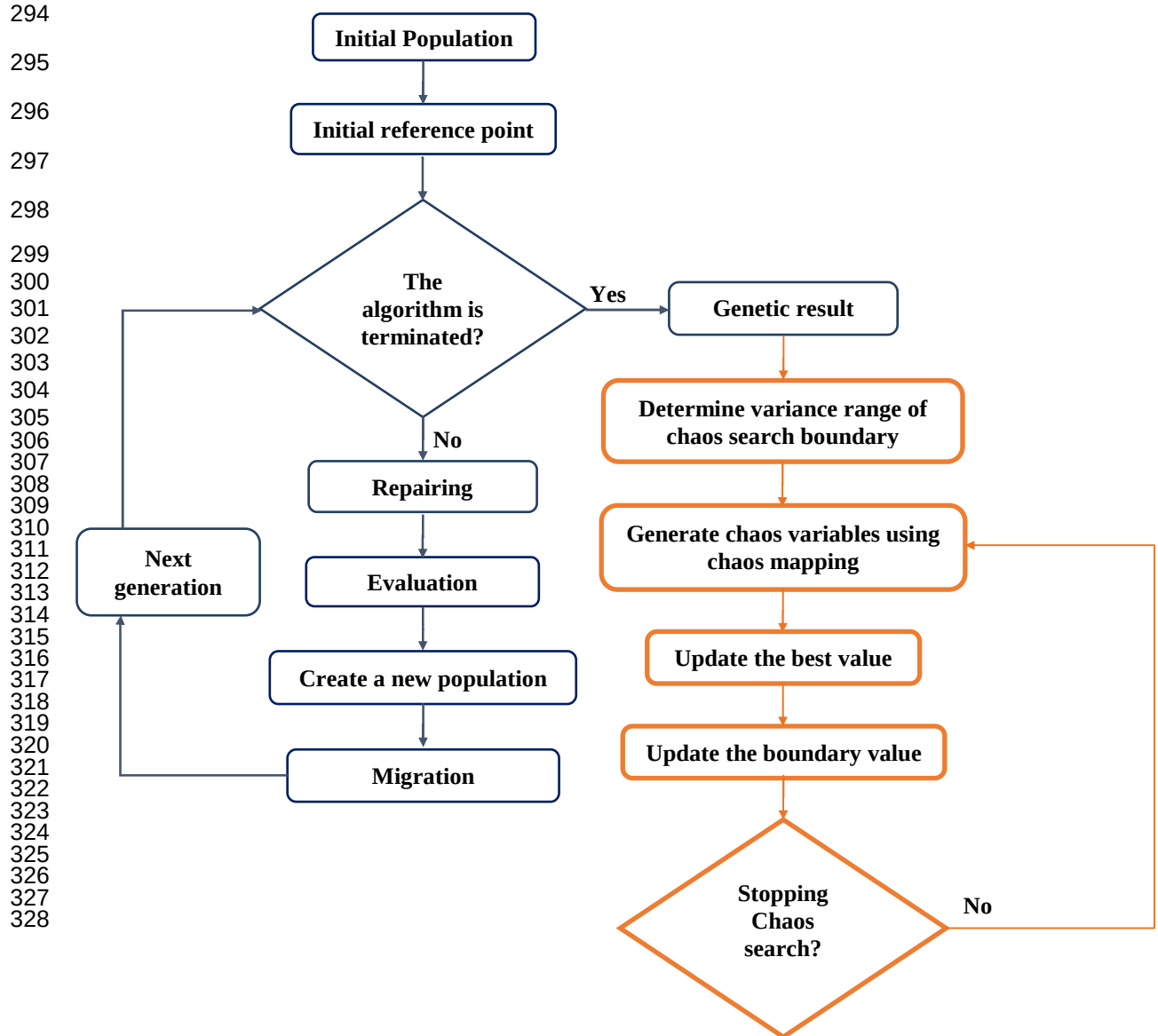
Step 3. Mapping chaos variable into the variance range: Chaos variable z^k is mapped into the variance range of optimization valuable $[a,b]$ by:

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

279 **Step 4. Update the best value:** Set chaotic iteration number as $k = 1 \rightarrow$ Do
 280 $x_i^k = x_i^* - \varepsilon + 2\varepsilon Z \quad \forall i = 1, \dots, n$.
 281 If $f(x^k) < f(x^*)$ then set $x^* = x^k$.
 282 Else if $f(x^k) \geq f(x^*)$ then give up the kth iterated.
 283 Result x^* .
 284 Loop runs until $f(x^*)$ is not improved after k searches.

285
 286 **Step 5. Update the boundary value:** Choose the boundary value $[a, b]$ of the new optimal
 287 point x_i^* as the new chaos search range. Chaos variable is mapped into the new search
 288 range by (16), then go to step 2.
 289

290 **Step 6. Stopping Chaos search:** If $f(x^*)$ is not improved for all k searches, stop Chaos
 291 search process and put out x^* as the best solution. The flow chart of the proposed algorithm
 292 showing in Fig. 2.
 293



329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351

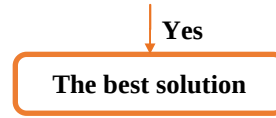


Fig. 2 The flow chart of the proposed algorithm
6. NUMERICAL STUDIES

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.

Table 1. Constrained benchmark problems

Problem	Variable bounds	Objective function $f(x)$ and constraints $C(x)$
P1	$x_1 \in [-10,10]$ $x_2 \in [-10,10]$	$f_1(x) = x_1^2 + x_2^2$ $C_1(x) = x_1 - 3 = 0$ $C_2(x) = 2 - x_2 \leq 0$
P2	$x_1 \in [-10,10]$ $x_2 \in [-10,10]$	$f_1(x) = \frac{1}{4000}(x_1^2 + x_2^2)^2 - \cos(\frac{x_1}{\sqrt{1}})\cos(\frac{x_2}{\sqrt{2}}) + 1$ $C_1(x) = x_1 - 3 = 0$ $C_2(x) = 2 - x_2 \leq 0$
P3	$x_1 \in [0.1,10]$ $x_2 \in [0,10]$	$f_1(x) = \frac{-\sin(2\pi x_1)^3 \sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$ $C_1(x) = x_1^2 - x_2 + 1 \leq 0$ $C_2(x) = 1 - x_1 + (x_2 - 4)^2 \leq 0$
P4	$x_1 \in [13,100]$ $x_2 \in [0,100]$	$f_1(x) = (x_1 - 10)^3 + (x_2 - 20)^3$ $C_1(x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$ $C_2(x) = (x_1 - 6) + (x_2 - 5)^2 - 82.81 \leq 0$

P5	$x_1 \in [-1,1]$ $x_2 \in [-1,1]$	$f_1(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$ $C_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0$ $C_2(x) = -85.334407 - 0.0056858x_2x_5 - 0.006262x_1x_4 + 0.0022053x_3x_5 \leq 0$ $C_3(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0$ $C_4(x) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0$ $C_5(x) = 9.300961 + 0.0047026x_3x_5 + 0 - 0.012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0$ $C_6(x) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0$
P6	$x_1 \in [78,100]$ $x_2 \in [33,45]$ $x_3 \in [27,45]$ $x_4 \in [27,45]$ $x_5 \in [27,45]$	$f_1(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$ $C_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0$ $C_2(x) = -85.334407 - 0.0056858x_2x_5 - 0.006262x_1x_4 + 0.0022053x_3x_5 \leq 0$ $C_3(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0$ $C_4(x) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0$ $C_5(x) = 9.300961 + 0.0047026x_3x_5 + 0 - 0.012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0$ $C_6(x) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0$

352

353

354 **Table 2. The proposed algorithm parameters**

355

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

356

357 **6.1. The results**

358

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

364

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

372

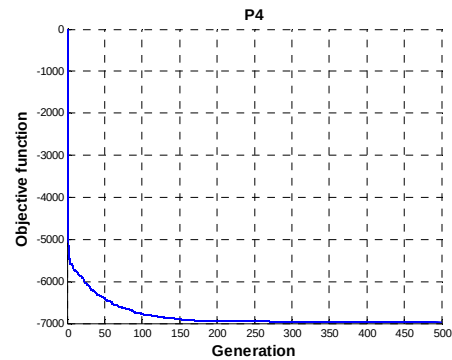
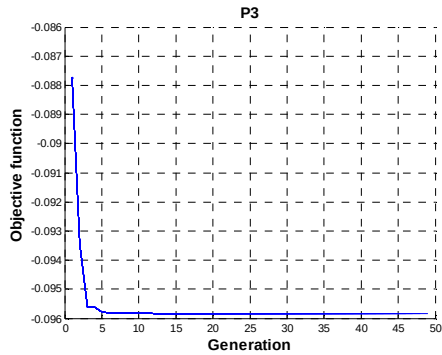
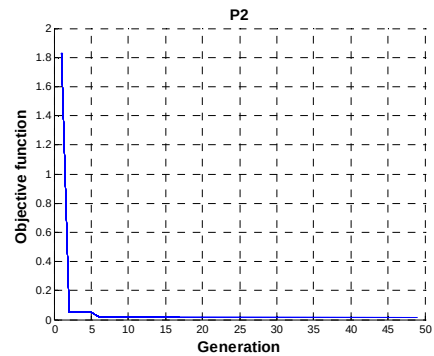
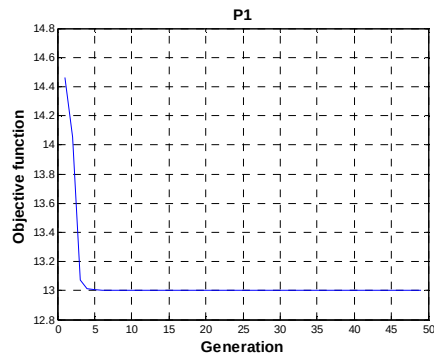
373

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

376

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

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).



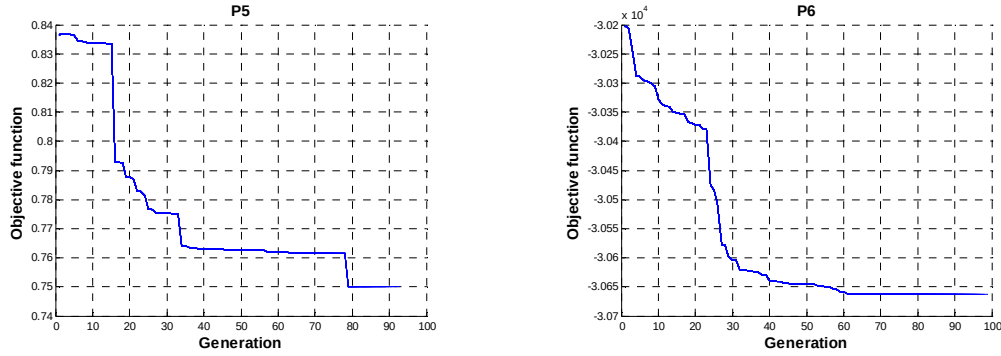


Fig. 3 The convergence rate of the proposed approach on different Constrained benchmark problems

7. CONCLUSIONS

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

420

421 REFERENCES

422

423 [1] Michae BB. Nonlinear Optimization with Engineering Applications, Springer Optimization
424 and Its Applications, 19 (2008).

425 [2] Rao SS. Engineering Optimization: Theory and Practice, 3rd ed., Wiley, NY, USA, 2009.

426 [3] Michalewicz Z. Evolutionary Computation Techniques for Nonlinear Programming
427 Problems, International Transactions in Operational Research, 1(2) (1994) 223–240.

428 [4] Onwubolu GC, Babu BV. New optimization Techniques in Engineering, Springer Science
429 & Business Media, 141 (2004).

430 [5] Gallagher K, Sambridge M. Genetic algorithms: A powerful tool for large-scale nonlinear
431 optimization problems, Compute and Geosciences, 20(7\8) (1994) 1229-1236.
432 Geosciences

433 [6] Neves N, Nguyen AT, Torres EL. A study of nonlinear optimization problems using a
434 distributed genetic algorithm, Proceedings of the 1996 International Conference on
435 Parallel Processing, 2 (1996) 29-36.

436 [7] Alrefaei MH, Diabat AH. A simulated annealing technique for multi-objective simulation
437 optimization, Applied Mathematics and Computation, 215 (2009) 3029–3035.

438 [8] Brooks SP, Morgan BJT. Optimization using simulated annealing, The statistician,
439 44(2)(1995) 241-257.

440 [9] Mousa AA, El-Shorbagy MA, Abd-El-WahedWF. Local search based hybrid particle
441 swarm optimization algorithm for multiobjective optimization, Swarm and Evolutionary
442 Computation, 3 (2012) 1–14.

443 [10] Yang X, Yuan J, Mao H. A modified particle swarm optimizer with dynamic adaptation,
444 Applied Mathematics and Computation, 189 (2007) 1205–1213.

445 [11] Mousa AA, AbdEl-Wahed WF, Rizk-Allah RM. A hybrid ant colony optimization
446 approach based local search scheme for multiobjective design optimizations, Electric
447 Power Systems Research, 81 (2011) 1014–1023,.

448 [12] Dréo J, Siarry P. An ant colony algorithm aimed at dynamic continuous optimization.
449 Appl. Math. Comput., 181(1) (2006) 457-467.

450 [13] Dorigo M, Stützle T. Ant Colony Optimization, MIT Press, London, 2004.

451 [14] Oltean G. Fuzzy Techniques in Optimization-Based Analog Design, 9th WSEAS
452 International Conference on fuzzy systems (FS'08) which was held in Sofia, Bulgaria,
453 (2008) 178-191.

454 [15] Fares M, Kaminska B. A Fuzzy Nonlinear Programming Approach to Analog Circuit
455 Design, IEEE Transactions on Computer-Aided Design of Integrated Circuits and
456 Systems, 14(17) (1995) 785-793.

457 [16] Hopfield JJ, Tank DW. "Neural" Computation of Decisions in Optimization Problem,
458 Biological Cybernetics, 52 (1985) 141-152.

459 [17] Zhou A. A Genetic-Algorithm-Based Neural Network Approach for Short-Term Traffic
460 Flow Forecasting, Advances in Neural Networks, 3498 (2005) 965-969.

461 [18] Elsayed SM, Sarker RA, Essam DL. A new genetic algorithm for solving optimization
462 problems, Engineering Applications of Artificial Intelligence, 27 (2014) 57–69.

463 [19] Deb K. Optimal design of a welded beam via genetic algorithms, AIAA Journal, 29(11)
464 (1991) 2013-2015.

465 [20] Malhotra R, Singh N, Singh Y. Genetic Algorithms: Concepts, Design for Optimization of
466 Process Controllers, Computer and Information Science, 4(2) (2011) 39-54.

467 [21] Tsoulos G. Solving constrained optimization problems using a novel genetic algorithm,
468 Applied Mathematics and Computation, 208(1) (2009) 273-283.

469 [22] Juan W, Ping W. Optimization of Fuzzy Rule Based on Adaptive Genetic Algorithm and
470 Ant Colony Algorithm, International Conference on Computational and Information
471 Sciences (ICIS), (2010) 359-362.

- 472 [23] Sun Fj, Tian Y. Transmission Line Image Segmentation Based GA and PSO Hybrid
473 Algorithm, 2010 International Conference on Computational and Information Sciences
474 (ICCCIS), (2010) 677-680.
- 475 [24] Hénon M. A two-dimensional mapping with a strange attractor, *Commun. Math. Phys.*,
476 50 (1976) 69-77.
- 477 [25] Lorenz E. *The Essence of Chaos*, University of Washington Press, 1996.
- 478 [26] Liu B, Wang L, Jin YH, Tang F, Huang DX. Improved particle swarm optimization
479 combined with chaos, *Chaos, Solitons and Fractals*, 25 (2005) 1261–1271.
- 480 [27] Xiang T, Liao X, Wong KW. An improved particle swarm optimization algorithm
481 combined with piecewise linear chaotic map, *Applied Mathematics and Computation*, 190
482 (2007) 1637–1645.
- 483 [28] Wang L, Zheng DZ, Lin QS. Survey on chaotic optimization methods, *Computing
484 Technology and Automation*, 20(1) (2001) 1–5.
- 485 [29] Coelho LS, Mariani VC. A novel chaotic particle swarm optimization approach using
486 Hénon map and implicit filtering local search for economic load dispatch, *Chaos, Solitons
487 and Fractals*, 39(2) (2009) 510–518.
- 488 [30] Chuanwen J, Bompard E. A self-adaptive chaotic particle swarm algorithm for short term
489 hydroelectric system scheduling in deregulated environment, *Energy Conversion and
490 Management*, 46 (2005) 2689–2696.
- 491 [31] Hu W, Liang H, Peng C, Du B, Hu Q. Hybrid Chaos-Particle Swarm Optimization
492 Algorithm for the Vehicle Routing Problem with Time Window, *Entropy*, 15(2013) 1247-
493 1270.
- 494 [32] Jampour M. Chaotic Genetic Algorithm based on Lorenz Chaotic System for
495 Optimization Problems, *Intelligent Systems and Applications*, 5(5) (2013) 19-24.
- 496 [33] Sedlacek K, Eberhard P. Constrained Particle Swarm Optimization of Mechanical
497 Systems, 6th World Congresses of Structural and Multidisciplinary Optimization, (2005).
- 498 [34] Abd-El-Wahed WF, Mousa AA, El-Shorbagy MA. Integrating particle swarm optimization
499 with genetic algorithms for solving nonlinear optimization problems, *Journal of
500 Computational and Applied Mathematics*, 235 (2011) 1446–1453.
- 501 [35] Ammaruekarat P, Meesad P. A Chaos Search for Multi-Objective Memetic Algorithm,
502 *International Conference on Information and Electronics Engineering*, 6 (2011) 140-144.
- 503 [36] Tavazoei MS, Haeri M. Comparison of different one-dimensional maps as chaotic
504 search pattern in chaos optimization algorithms, *Appl. Math. Comput.*, 187 (10) (2007)
505 76–85.
- 506 [37] Hilborn RC. *Chaos and Nonlinear Dynamics: An Introduction for Scientists and
507 Engineers*, second ed., Oxford Univ. Press, New York, 2004.
- 508 [38] He D, He C, Jiang L, Zhu H, Hu G. Chaotic characteristic of a one dimensional iterative
509 map with infinite collapses, *IEEE Trans. Circuits Syst.*, 48(7) (2001) 900–906.
- 510 [39] Erramilli A, Singh RP, Pruthi P. Modeling Packet Traffic With Chaotic Maps, Royal
511 Institute of Technology, Stockholm-Kista, Sweden, 1994.
- 512 [40] May RM. Simple mathematical models with very complicated dynamics, *Nature*, 261(4)
513 (1976) 59–67.
- 514 [41] Arora JS, Elwakeil OA, Chahande AI, Hsieh CC. Global optimization methods for
515 engineering application: a review, *Struct. Optim.*, 9 (1995) 137–159.
- 516 [42] Li Y, Deng S, Xiao D. A novel Hash algorithm construction based on chaotic neural
517 network, *Neural Comput. Appl.*, 20 (2011) 133–141.
- 518 [43] Devaney RL. *An Introduction to Chaotic Dynamical Systems*, Addison-Wesley, 1987.
- 519 [44] Peitgen H, Jürgen H, Saupe D. *Chaos and Fractals*, Springer-Verlag, Berlin, Germany,
520 1992.
- 521 [45] Ott E. *Chaos in Dynamical Systems*, Cambridge University Press, UK, Cambridge,
522 2002.
- 523 [46] Osman MS, Abo-Sinna MA, Mousa AA. A solution to the optimal power flow using
524 genetic algorithm, *Applied Mathematics and Computation*, 155 (2004) 391–405.

- 525 [47] Baker JE. Reducing Bias and Inefficiency in the Selection Algorithm, Proceedings of the
526 Second International Conference on Genetic Algorithms, Morgan Kaufmann Publishers,
527 Inc., San Mateo, CA, (1987) 14-21.
- 528 [48] Popov A. Genetic algorithm for optimization, TU-Sofia, 2003.
- 529 [49] Neves N, Nguyen A.-T, Torres EL. A study of nonlinear optimization problem using a
530 distributed genetic algorithm, 25th International Conference on Parallel Processing,
531 (1996).