

Original Research Article

Genetic Algorithm Based Chaotic Search for Constrained Nonlinear Optimization Problems

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].

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 OPTIMAZATION PROBLEMS

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

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

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 represents a vector of decision variables which take real values, and each decision variable x_n is constrained by its lower and upper boundaries $[x_n^l, x_n^u]$; N is the total number of decision variables x_n . The above problem must be solved for the values of the variables $x_1, \dots, x_N$ 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 $g_j(x) \leq 0$ for $j = 1, 2, \dots, p$ is called an inequality constraint, and each of the constraints $h_i(x) = 0$ for $i = 1, 2, \dots, q$ 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.

126 4. CHAOTIC MAPS

127

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

135

136 - **Chebyshev map**

137 Chebyshev map is represented as [36]:

$$138 \quad x_{t+1} = \cos\left(t \cos^{-1}(x_t)\right). \quad (2)$$

139 - **Circle map**

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

$$141 \quad x_{t+1} = x_t + b - (a - 2\pi)\sin(2\pi x_t) \bmod(1); \quad (3)$$

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

143 - **Gauss/mouse map**

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

$$145 \quad x_{t+1} = \begin{cases} 0 & \text{if } x_t = 0 \\ 1/x_t & \text{else} \end{cases} \bmod(1); \quad (4)$$

$$146 \quad \text{where } \frac{1}{x_t} \bmod(1) = \frac{1}{x_t} - \left\lfloor \frac{1}{x_t} \right\rfloor.$$

147 - **Intermittency map**

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

$$149 \quad x_{t+1} = \begin{cases} \varepsilon + x_t + cx_t^n & \text{if } 0 < x_t \leq p \\ \frac{x_t - p}{1 - p} & \text{elseif } p < x_t < 1 \end{cases}; \quad (5)$$

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

151 - **Iterative map**

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

$$153 \quad x_{t+1} = \sin\left(\frac{a\pi}{x_t}\right); \quad (6)$$

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

155 - **Liebovitch map**

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

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

158

$$159 \quad \text{where } \alpha = \frac{p_2(1 - (p_2 - p_1))}{p_1} \text{ and } \beta = \frac{((p_2 - 1) - p_1(p_2 - p_1))}{p_2 - 1}.$$

160 - **Logistic map**

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

$$164 \quad x_{t+1} = cx_t(1 - x_t); \quad (8)$$

165 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
166 sequence is generated by the Logistic map.

167 - **Piecewise map**

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

$$169 \quad x_{t+1} = \begin{cases} \frac{x_t}{p} & 0 < x_t < p \\ \frac{x_t - p}{0.5 - p} & p \leq x_t < 0.5 \\ \frac{(1 - p - x_t)}{0.5 - p} & 0.5 \leq x_t < 1 - p \\ \frac{(1 - x_t)}{p} & 1 - p < x_t < 1 \end{cases}; \quad (9)$$

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

171 - **Sine map**

172 Sine map [43] can be described as:

$$173 \quad x_{t+1} = \frac{a}{4} \sin(\pi x_t); \quad (10)$$

174 where $0 < a < 4$.

175 - **Singer map**

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

$$177 \quad x_{t+1} = \mu(7.86x_t - 23.31x_t^2 + 28.75x_t^3 - 13.302875x_t^4); \quad (11)$$

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

179 - **Sinusoidal map**

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

$$181 \quad x_{t+1} = ax_t^2 \sin(\pi x_t); \quad (12)$$

182 where $a = 2.3$.

183 - **Tent map**

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

$$185 \quad x_{t+1} = \begin{cases} x_t / 0.07 & x_t < 0.7 \\ \frac{10}{3}(1.0 - x_t) & x_t \geq 0.7 \end{cases} \quad (13)$$

186

187

188 **5. THE PROPOSED ALGORITHM**

189

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

196

197 **Phase I: GA**

198

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

$$202 \quad \text{Each individual position } |_i = L + (U - L) \times \text{rand}; \quad (14)$$

203 where $i = 1, \dots, N_{pop}$, and N_{pop} is the size of the population; L is the lower bound; U is the
204 upper bound, and rand is random numbers uniformly distributed within the range $[0, 1]$.
205

206 **Step 2. Initial reference point:** The algorithm needs at least one feasible reference point
207 (i.e., feasible point) to enter the evolution process (i.e., complete the algorithm procedure),
208 the reader is referred to Osman et al. [46].
209

210 **Step 3. Repairing:** Repair the infeasible individuals of the population to be feasible. The
211 idea of this technique is to separate any feasible individuals in a population from those that
212 are infeasible by repairing infeasible individuals. This approach co-evolves the population of
213 infeasible individuals until they become feasible, the reader is referred to [34].
214

215 **Step 4. Evaluation:** Evaluate the desired optimization fitness function in n variables for each
216 individual.
217

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

- 220 • **Ranking:** Ranks individuals according to their fitness value, and returns a column
221 vector containing the corresponding individual fitness value, in order to establish later
222 the probabilities of survival that are necessary for the selection process [20].
- 223 • **Selection:** There are several techniques of selection. The commonly used techniques
224 for selection of individuals are roulette wheel selection, rank selection, steady state
225 selection, stochastic universal sampling, etc.. Here we will use Stochastic Universal
226 Sampling (SUS) [47] where, the most important concern in a stochastic selection is to
227 prevent loss of population diversity due to its stochastic aspect.
- 228 • **Crossover:** In GAs, crossover is used to vary individuals from one generation to the
229 next; where it combines two individuals (parents) to produce a new individuals
230 (offspring) with probability (Pc). There are several techniques of crossover, one-point

231 crossover, two-point crossover, cut and splice, uniform crossover and half uniform
232 crossover, etc.[34,48]. Here we will use One-point crossover involving splitting two
233 individuals and then combining one part of one with the other pair. This method
234 performs recombination between pairs of individuals and returns the new individuals
235 after mating, and gives offspring the best possible combination of the characteristics of
236 their parents.

237 • **Mutation:** Premature convergence is a critical problem in most optimization techniques,
238 which occurs when highly fit parent individuals in the population breed many similar
239 offspring in early evolution time. Mutation is used to maintain genetic diversity from one
240 generation of a population to the next. In addition, Mutation is an operator to change
241 elements in a string which is generated after crossover operator [48]. In this study, we
242 will use real valued mutation; which means that randomly created values are added to
243 the variables with a low probability (Pm).
244

245 **Step 6. Migration:** In this step, a migration of the new offspring with the old population to
246 create the new population by taking the best individuals of parents and offspring population
247 [48,49].
248

249 **Step 7. Termination test:** The algorithm is terminated either when the maximum number of
250 generations is achieved, or when the individuals of the population coverages, convergence
251 occur when all individuals positions in the population are identical. In this case, crossover will
252 have no further effect. otherwise, return to step 3.
253

254 Phase 2: Chaos search

255
256 **Step 1. Determine variance range of chaos search boundary:** The range of chaotic local
257 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
258 genetic result, and ε is specified radius of chaos search.
259

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

$$263 \quad z^{k+1} = \mu z^k (1 - z^k), \quad z^0 \in (0,1), \quad z^0 \notin \{0.0, 0.25, 0.50, 0.75, 1.0\}, \quad k = 1, 2, \dots \quad (15)$$

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

$$266 \quad x_i = x_i^* - \varepsilon + 2\varepsilon z^k \quad \forall i = 1, \dots, n \quad (16)$$

267 **Step 4. Update the best value:** Set chaotic iteration number as $k = 1 \rightarrow$ Do
268 $x_i^k = x_i^* - \varepsilon + 2\varepsilon z^k \quad \forall i = 1, \dots, n$.

269 If $f(x^k) < f(x^*)$ then set $x^* = x^k$.

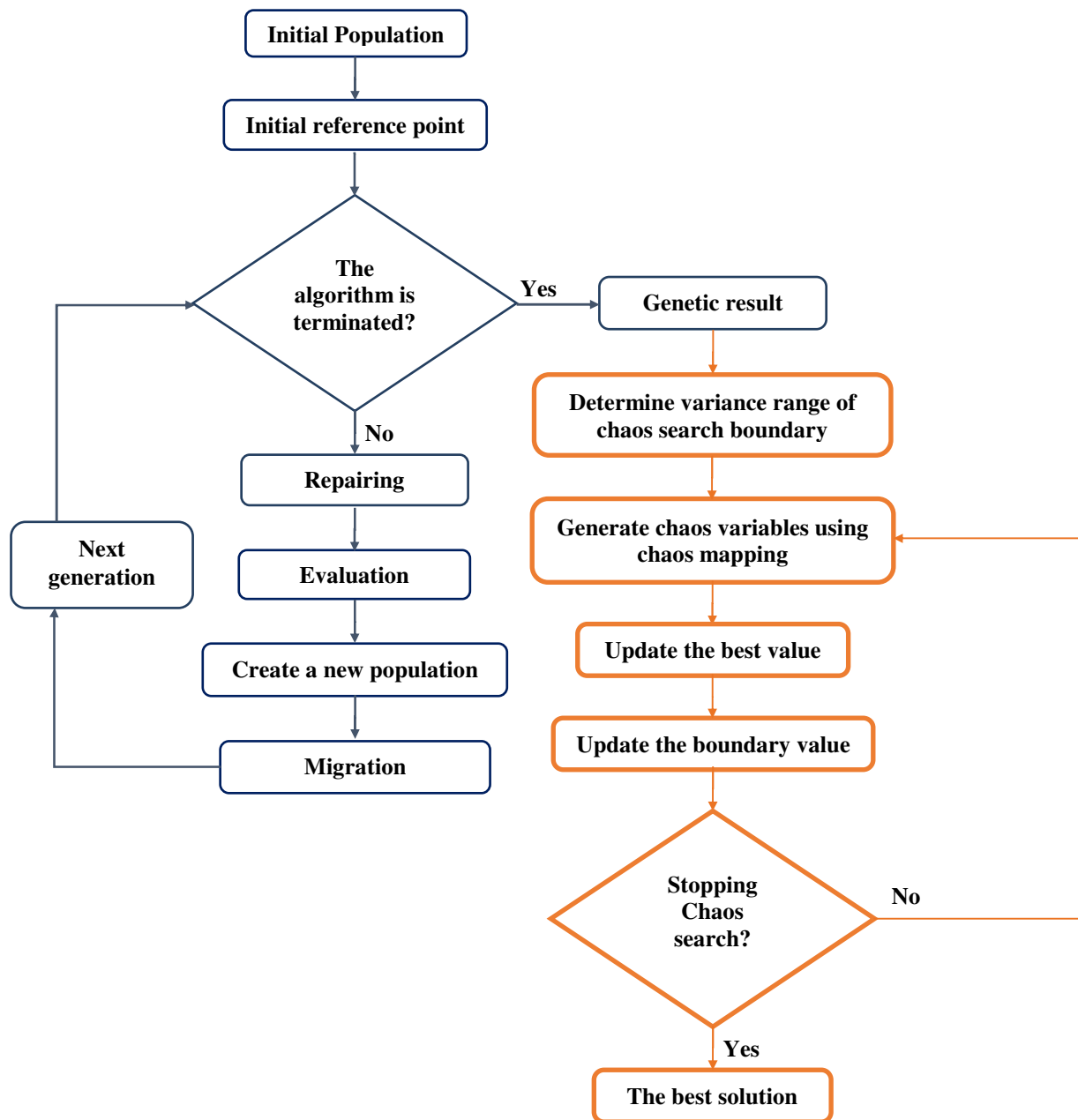
270 Else if $f(x^k) \geq f(x^*)$ then give up the kth iterated.

271 Result x^* .

272 Loop runs until $f(x^*)$ is not improved after k searches.
273

274 **Step 5. Update the boundary value:** Choose the boundary value $[a, b]$ of the new optimal
 275 point x_i^* as the new chaos search range. Chaos variable is mapped into the new search
 276 range by (16), then go to step 2.
 277

278 **Step 6. Stopping Chaos search:** If $f(x^*)$ is not improved for all k searches, stop Chaos
 279 search process and put out x^* as the best solution. The flow chart of the proposed algorithm
 280 showing in Fig. 2.
 281



322
 323 **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	$x_1 \in [78, 100]$	$f_1(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$
	$x_2 \in [33, 45]$	$C_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0$
	$x_3 \in [27, 45]$	$C_2(x) = -85.334407 - 0.0056858x_2x_5 - 0.006262x_1x_4 + 0.0022053x_3x_5 \leq 0$
	$x_4 \in [27, 45]$	$C_3(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0$
	$x_5 \in [27, 45]$	$C_4(x) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0$
		$C_5(x) = 9.300961 + 0.0047026x_3x_5 + 0 - 0.012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0$
		$C_6(x) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0$

340

341

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

343

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

344

345 6.1. The results

346

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

352

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

361

362

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

365

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	-30665.8

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.

REFERENCES

- [1] B.-B. Michael, Nonlinear Optimization with Engineering Applications, Springer Optimization and Its Applications, 19 (2008).
- [2] S.S. Rao, Engineering Optimization: Theory and Practice, 3rd ed., Wiley, NY, USA, 2009.
- [3] Z. Michalewicz, Evolutionary Computation Techniques for Nonlinear Programming Problems, International Transactions in Operational Research, 1(2) (1994) 223–240.
- [4] G.C. Onwubolu and B.V. Babu, New optimization Techniques in Engineering, Springer Science & Business Media, 141 (2004).
- [5] K. Gallagher and M. Sambridge, Genetic algorithms: A powerful tool for large-scale nonlinear optimization problems, Compute and Geosciences, 20(7\8) (1994) 1229-1236. Geosciences
- [6] N. Neves, A.-T. Nguyen and E.L. Torres, A study of nonlinear optimization problems using a distributed genetic algorithm, Proceedings of the 1996 International Conference on on Parallel Processing, 2 (1996) 29-36.
- [7] M.H. Alrefaei and A.H. Diabat, A simulated annealing technique for multi-objective simulation optimization, Applied Mathematics and Computation, 215 (2009) 3029–3035.
- [8] S.P. Brooks and B.J.T Morgan, Optimization using simulated annealing, The statistician, 44(2)(1995) 241-257.

- 419 [9] A.A. Mousaa, M.A. El-Shorbagy and W.F. Abd-El-Wahed, Local search based hybrid
420 particle swarm optimization algorithm for multiobjective optimization, *Swarm and*
421 *Evolutionary Computation*, 3 (2012) 1–14.
- 422 [10] X. Yang, J. Yuan, J. Yuan and H. Mao, A modified particle swarm optimizer with
423 dynamic adaptation, *Applied Mathematics and Computation*, 189 (2007) 1205–1213.
- 424 [11] A.A. Mousa, W.F. Abd El-Wahed and R. M. Rizk-Allah, A hybrid ant colony optimization
425 approach based local search scheme for multiobjective design optimizations, *Electric Power*
426 *Systems Research*, 81 (2011) 1014–1023,.
- 427 [12] J. Dréo and P. Siarry, An ant colony algorithm aimed at dynamic continuous
428 optimization. *Appl. Math. Comput.*, 181(1) (2006) 457-467.
- 429 [13] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, London, 2004.
- 430 [14] G. Oltean, *Fuzzy Techniques in Optimization-Based Analog Design*, 9th WSEAS
431 *International Conference on fuzzy systems (FS'08)* which was held in Sofia, Bulgaria, (2008)
432 178-191.
- 433 [15] M. Fares and B. Kaminska, A Fuzzy Nonlinear Programming Approach to Analog Circuit
434 Design, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*,
435 14(17) (1995) 785-793.
- 436 [16] J.J. Hopfield and D.W. Tank, “Neural” Computation of Decisions in Optimization
437 Problem, *Biological Cybernetics*, 52 (1985) 141-152.
- 438 [17] A. Zhou, A Genetic-Algorithm-Based Neural Network Approach for Short-Term Traffic
439 Flow Forecasting, *Advances in Neural Networks*, 3498 (2005) 965-969.
- 440 [18] S.M. Elsayed, R.A. Sarker and D.L. Essam, A new genetic algorithm for solving
441 optimization problems, *Engineering Applications of Artificial Intelligence*, 27 (2014) 57–69.
- 442 [19] K. Deb, Optimal design of a welded beam via genetic algorithms, *AIAA Journal*, 29(11)
443 (1991) 2013-2015.
- 444 [20] R. Malhotra, N. Singh and Y. Singh, Genetic Algorithms: Concepts, Design for
445 Optimization of Process Controllers, *Computer and Information Science*, 4(2) (2011) 39-54.
- 446 [21] I.G. Tsoulos, Solving constrained optimization problems using a novel genetic algorithm,
447 *Applied Mathematics and Computation*, 208(1) (2009) 273-283.
- 448 [22] W. Juan and W. Ping, Optimization of Fuzzy Rule Based on Adaptive Genetic Algorithm
449 and Ant Colony Algorithm, *International Conference on Computational and Information*
450 *Sciences (ICCIS)*, (2010) 359-362.
- 451 [23] F.-j. Sun and Y. Tian, Transmission Line Image Segmentation Based GA and PSO
452 Hybrid Algorithm, 2010 *International Conference on Computational and Information*
453 *Sciences (ICCIS)*, (2010) 677-680.
- 454 [24] M. Hénon, A two-dimensional mapping with a strange attractor, *Commun. Math. Phys.*,
455 50 (1976) 69-77.
- 456 [25] E. Lorenz, *The Essence of Chaos*, University of Washington Press, 1996.
- 457 [26] B. Liu, L. Wang, Y.H. Jin, F. Tang and D.X. Huang, Improved particle swarm
458 optimization combined with chaos, *Chaos, Solitons and Fractals*, 25 (2005) 1261–1271.
- 459 [27] T. Xiang, X. Liao and K.W. Wong, An improved particle swarm optimization algorithm
460 combined with piecewise linear chaotic map, *Applied Mathematics and Computation*, 190
461 (2007) 1637–1645.
- 462 [28] L. Wang, D.Z. Zheng and Q.S. Lin, Survey on chaotic optimization methods, *Computing*
463 *Technology and Automation*, 20(1) (2001) 1–5.
- 464 [29] Leandro dos Santos Coelho and V.C. Mariani, A novel chaotic particle swarm
465 optimization approach using Hénon map and implicit filtering local search for economic load
466 dispatch, *Chaos, Solitons and Fractals*, 39(2) (2009) 510–518.
- 467 [30] J. Chuanwen, E. Bompard, A self-adaptive chaotic particle swarm algorithm for short
468 term hydroelectric system scheduling in deregulated environment, *Energy Conversion and*
469 *Management*, 46 (2005) 2689–2696.
- 470 [31] W. Hu, H. Liang, C. Peng, B. Du and Q. Hu, Hybrid Chaos-Particle Swarm Optimization
471 Algorithm for the Vehicle Routing Problem with Time Window, *Entropy*, 15(2013) 1247-1270.

- 472 [32] M. Jampour, Chaotic Genetic Algorithm based on Lorenz Chaotic System for
473 Optimization Problems, *Intelligent Systems and Applications*, 5(5) (2013) 19-24.
- 474 [33] K. Sedlaczek and P. Eberhard, Constrained Particle Swarm Optimization of Mechanical
475 Systems, 6th World Congresses of Structural and Multidisciplinary Optimization, (2005).
- 476 [34] W.F. Abd-El-Wahed, A.A. Mousa and M.A. El-Shorbagy, Integrating particle swarm
477 optimization with genetic algorithms for solving nonlinear optimization problems, *Journal of*
478 *Computational and Applied Mathematics*, 235 (2011) 1446–1453.
- 479 [35] P. Ammaruekarat and P. Meesad, A Chaos Search for Multi-Objective Memetic
480 Algorithm, *International Conference on Information and Electronics Engineering*, 6 (2011)
481 140-144.
- 482 [36] M.S. Tavazoei and M. Haeri, Comparison of different one-dimensional maps as chaotic
483 search pattern in chaos optimization algorithms, *Appl. Math. Comput.*, 187 (10) (2007) 76–
484 85.
- 485 [37] R.C. Hilborn, *Chaos and Nonlinear Dynamics: An Introduction for Scientists and*
486 *Engineers*, second ed., Oxford Univ. Press, New York, 2004.
- 487 [38] D. He, C. He, L. Jiang, H. Zhu and G. Hu, Chaotic characteristic of a one dimensional
488 iterative map with infinite collapses, *IEEE Trans. Circuits Syst.*, 48(7) (2001) 900–906.
- 489 [39] A. Erramilli, R.P. Singh and P. Pruthi, *Modeling Packet Traffic With Chaotic Maps*, Royal
490 Institute of Technology, Stockholm-Kista, Sweden, 1994.
- 491 [40] R.M. May, Simple mathematical models with very complicated dynamics, *Nature*, 261(4)
492 (1976) 59–67.
- 493 [41] J.S. Arora, O.A. Elwakeil, A.I. Chahande and C.C. Hsieh, Global optimization methods
494 for engineering application: a review, *Struct. Optim.*, 9 (1995) 137–159.
- 495 [42] Y. Li, S. Deng and D. Xiao, A novel Hash algorithm construction based on chaotic
496 neural network, *Neural Comput. Appl.*, 20 (2011) 133–141.
- 497 [43] R.L. Devaney, *An Introduction to Chaotic Dynamical Systems*, Addison-Wesley, 1987.
- 498 [44] H. Peitgen, H. Jurgens and D. Saupe, *Chaos and Fractals*, Springer-Verlag, Berlin,
499 Germany, 1992.
- 500 [45] E. Ott, *Chaos in Dynamical Systems*, Cambridge University Press, UK, Cambridge,
501 2002.
- 502 [46] M.S. Osman, M.A. Abo-Sinna and A.A. Mousa, A solution to the optimal power flow
503 using genetic algorithm, *Applied Mathematics and Computation*, 155 (2004) 391–405.
- 504 [47] J.E. Baker, Reducing Bias and Inefficiency in the Selection Algorithm, *Proceedings of*
505 *the Second International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers,
506 Inc., San Mateo, CA, (1987) 14-21.
- 507 [48] A. Popov, *Genetic algorithm for optimization*, TU-Sofia, 2003.
- 508 [49] N. Neves, A.-T. Nguyen and E.L. Torres, A study of nonlinear optimization problem
509 using a distributed genetic algorithm, *25th International Conference on Parallel Processing*,
510 (1996).