2. CONSTRAINED NONLINEAR PROGRAMMING CASES (CNLC).

In the definition of the common nonlinear programming problem, the objective chosen functions $f(\bar{x})$ and control chosen functions $g_i(\bar{x})$ are nonlinear function. Finding the optimum solution of a nonlinear function is especially difficult; consequently, no one know an approach for locating the universal best solution to the common nonlinear programming case.

As stated in [3], let $\bar{x}$ be continuous variables, the general NLPC with respect to continuous variables is to select $\bar{x}$ such that

$$Min\ f(\bar{x}),$$
$$where \qquad \bar{x} = (x_1, x_2, \dots, x_n) \in R^n \qquad (1)$$

While $\bar{x} = (x_1, x_2, \dots, x_n) \in G$ (feasible space). The search space is the set $H \subseteq R^n$, and the feasible part of the search space is symbolized by the set $G \subseteq H$. Normally, the seeking universe is described by n-dimensional rectangular hypercube in R^n . Where the domains of used variables can be defined as lower and upper bounds:

$$Lift(i) \le x_i \le Right(i), \quad 1 \le i \le n \quad .$$

While, the suitable set F is realized as that part of the seeking universe S which satisfies an extra set of control restrictions:

$$g_i(\bar{x}) \le 0 \text{ and } i = 1,..,m \qquad (2)$$

The following is the definition of NLPC:

$$NLPP : \text{Max} f(\bar{x})$$
$$s.t.$$
$$G = \{\bar{x} \in R^n \mid \psi_i(\bar{x}) \le 0,\ i = 1,2\dots,k \ \ and \ \ \phi_j(\bar{x}) = 0,\ j = k+1,2,..,m\}$$
$$H = \{\bar{x} \in R^n \mid L_i \le \bar{x}_i \le U_i,\ i = 1,2,..,n\} \qquad (3)$$

All set of predefined nonlinear equations $\phi_j(\bar{x}) = 0,\ j = k+1,2,..,m$ are transformed to a set of pairs of related inequalities: $-\varepsilon \le \phi_j(\bar{x}) \le \varepsilon$ with additional parameter (ε) which define the degree of precision of the system. So, only nonlinear inequalities are used by our proposed approach.

$$NLPP : Max\ f(\bar{x})$$
$$s.t.$$
$$G = \{\bar{x} \in R^n \mid g_i(\bar{x}) \le 0,\ i = 1, 2\dots, 2m-k\}$$
$$H = \{\bar{x} \in R^n \mid L_i \le \bar{x}_i \le U_i,\ i = 1, 2, \dots, n\} \qquad (4)$$

3. BASIC CONCEPTS AND HISTORY OF GENETIC ALGORITHMS (GAs)

In 1975, based on Darwin's concept of Evolution, GA was initiated by Holland [29]. GA begins with a group of candidate solutions (called individuals) or randomly generated chromosome. The candidate solutions are evolved during several iterations seeking to converge to the fittest solution. New offspring are generated by applying GA operators (mutation and crossover). Crossover includes splitting of two or more chromosomes and combining them in order to produce new offspring. On the prime hand, mutation includes changing a single gene of a chromosome. Then, the individual chromosomes are evaluated using a certain criteria and the elite selections are saved to the following generation. This process is continued till the stopping rule is satisfied. The better fitness individual chromosome is taken as the end selection of the problem. Figure 1 shows a typical GA's flow chart.

GA has several advantages. The most important GA's advantages are as it doing well for universal seeking in optimization cases, particularly, when the goal function is non-convex, discrete or have many topical minima. These advantages lead to some potential disadvantages. GA disadvantages because an extra information like gradients are not used by GA and its rate of convergence is slow with well-behaved goal functions.

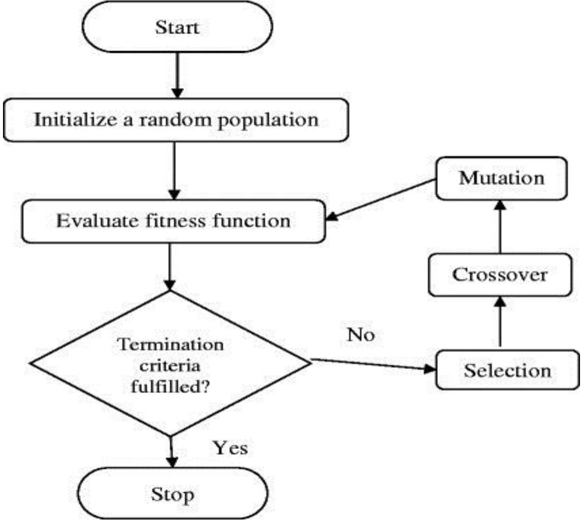
Start
Initialize a random population
Evaluate fitness function
Mutation
Crossover
Termination criteria fulfilled?
No
Selection
Yes
Stop

Fig. 1. Genetic Algorithm's flow chart

4. CHAOS THEORY

Chaos was presented by Henon (1976) [**23**] and it was summarized by Lorenz[24]. Chaos theory refers to the study of chaotic dynamical systems as water flows, weather patterns, and anatomical functions. Chaotic systems represented by nonlinear dynamical systems that are strongly sensible to the starting state, while tiny variations in starting state cause large variations in the end output of system. Chaos theory could be coupled with the theory of optimization to speed up the optimum search procedure in order to get the universal best selection.

Chaotic functions is defined as a certain map that introduces some type of chaotic behaviour. Those chaotic functions can be represented by a discrete-time type or a continuous-time type. In this research we concentrate on Discrete chaotic functions which usually represented in the form of iterated functions. For the interest reader certain well-defined chaotic functions introduced in the scientific search area can be found in [30].

5. THE NOVEL SUGGESTED ALGORITHM

Here, we have introduced the suggested novel algorithm for solving constrained NLP, Sinusoidal chaotic search based genetic algorithm (SCGA), which is hybridization between GAs and chaotic map based local search algorithm. The proposed algorithm consists of two stages (called phases). For the prime stage, GA is applied as universal optimization algorithm to get out an inexact optimal selection for the CNLP case. After that, in the second stage (phase), Sinusoidal chaotic search is executed to speed up the convergence rate and enhance the selection quality of the problem. The detailed description concept of the proposed system can be explained as shown in the next steps:

Stage I: GA

Step1. Initial Population

Firstly all search variables are initially randomly created satisfying the entire range of search variables (i.e., satisfying the search space).

Step2. Reference Common Point

Only one feasible reference common point is required by the proposed method (i.e., feasible point satisfying the set of constraints) to complete the evolution Mating, the interested reader may find additional information on [31-32].

Step 3. Modified Repairing

The main concept of this scheme is firstly, distinguish all possible solutions (individuals that are feasible) in each algorithm population from those that are infeasible. Secondly, repair each infeasible individual to become feasible. Using this method, the algorithm co-evolves all set of infeasible solutions till they turn to be feasible. The description of repair algorithm is as follow:

Suppose that we have a point satisfying the search universe $\boldsymbol{b} \in \boldsymbol{H}, \boldsymbol{b} \notin \boldsymbol{G}$ (while $\boldsymbol{H}$ is the seeking universe and $\boldsymbol{G}$ is the feasible space). The algorithm detect one reference point, say $\boldsymbol{a} \in \boldsymbol{G}$ and randomly generates points $\bar{\boldsymbol{A}}$ from the segment line between the two points $\boldsymbol{b}, \boldsymbol{a}$, where this segment line might be expanded on the two sides equally [31, 32], which determined by parameter $\mu \in [0,1]$. Then , new feasible individuals α_1, α_2 are written as follows:

$$\alpha_1 = \gamma.\boldsymbol{b} + (1-\gamma).\boldsymbol{a}, \quad \alpha_2 = (1-\gamma).\boldsymbol{b} + \gamma.\boldsymbol{a}$$

While $\gamma = (2\mu+1)\delta - \mu$, $\mu \in [0,1]$ is a user specified parameters, and $\delta \in [0,1]$ is a number that is generated randomly.

Step 4. Evaluation

In the fitness evaluation stage, for each individual, we determine a ranking function (ranking metric) of chromosomes fitness. This ranking metric determines which individual will be stil exist to the coming descent via detecting chromosomes that have higher fitness. **Step 5. Generating new population**

To generate a new proper population, we implement the genetic operators (crossover and mutation), the detailed description are as follows:

Ranking: Using fitness value for each individual, the population of individuals was ranked, which return a vector of scalar containing the individual fitness value, this vector is used to calculate their probabilities that are needed for the choice operation.

Selection: Roulette wheel election is an efficient genetic algorithm operator applied in GA for choosing useful candidates for recombination. The simulation may be considered by supposing a roulette wheel in which each potential selection corresponds to a cavity on the wheel, the volume of the cavity is proportional to the probability of choosing of the selected solution[14].

Crossover: crossover combines two or more parents to create a novel offspring with crossover prescribed probability P_c. In this research, we implement Multi-point crossover [33].

Mutation: Mutation is implemented to maintain genotype diversity during the algorithm evolution. Mutation is considered as a genetic algorithm agent, that change values of one or more gene in a chromosome from its starting values [34]. In this research, we implement real valued mutation; where we added a generated randomly values to each variable with a prescribed probability $P_m <<< 1$.

Step 6. Stopping rule

The optimization method is ended after predetermined number of generations

MaxGen (*MaxGen* is a user defined parameter) has been achieved.

Stage II : Sinusoidal map based local search

Chaotic mapping-based topical seeking improve the algorithm capability to disturb x^* in a specified local neighborhood of search space[35]. The procedure of chaotic Sinusoidal map based local search was described and implemented as following:

Step 1. Chaotic search boundary's variance range

The span of chaotic mapping-based-local search (i.e., upper bound and lower bound for each variable) $\{[L_i, U_i], \ i = 1, 2, \dots, N\}$ is determined by the relation $L_i > x_i^* - \xi, \ U_i < x_i^* + \xi$; where ξ is radius of chaos search scheme, see Figure 2 .

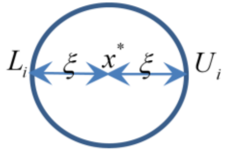

Fig 2: The range of chaotic mapping-based-local search

Step 2. Sinusoidal Chaotic Variables Generator

Perform chaos map by applying the Sinusoidal Chaotic map operator. Sinusoidal Chaotic map [30] is performed according to the following discrete dynamical system which defined by the iterative function, where Chaos variable z^k is created as described in the coming iterative formula:

$$z_{t+1} = \sin(\pi z_t); \tag{5}$$

Step 3. Mapping chaotic variable

Generated chaotic element z^k is transformed into the variance period $[L_i, U_i]$ of the selected optimization variable by the following relation:

$$x_i^k = L_i + (U_i - L_i) z^k \tag{6}$$

Step 4. Updating the better selected value

If $f\left(x^k\right) < f\left(x^*\right)$ then put $x^* = x^k$, else stop the iteration process.

Step 5. Chaotic Search Stopping Criteria

If the function value $f\left(x^*\right)$ has not get better for all predetermined *k* iteration, then end the iteration operation and set it out as the better reached solution value.

The flow chart of the proposed hybrid algorithm is illustrated in figure 3, while figure 4 declares the structure of the pseudo code of the proposed chaotic local search,

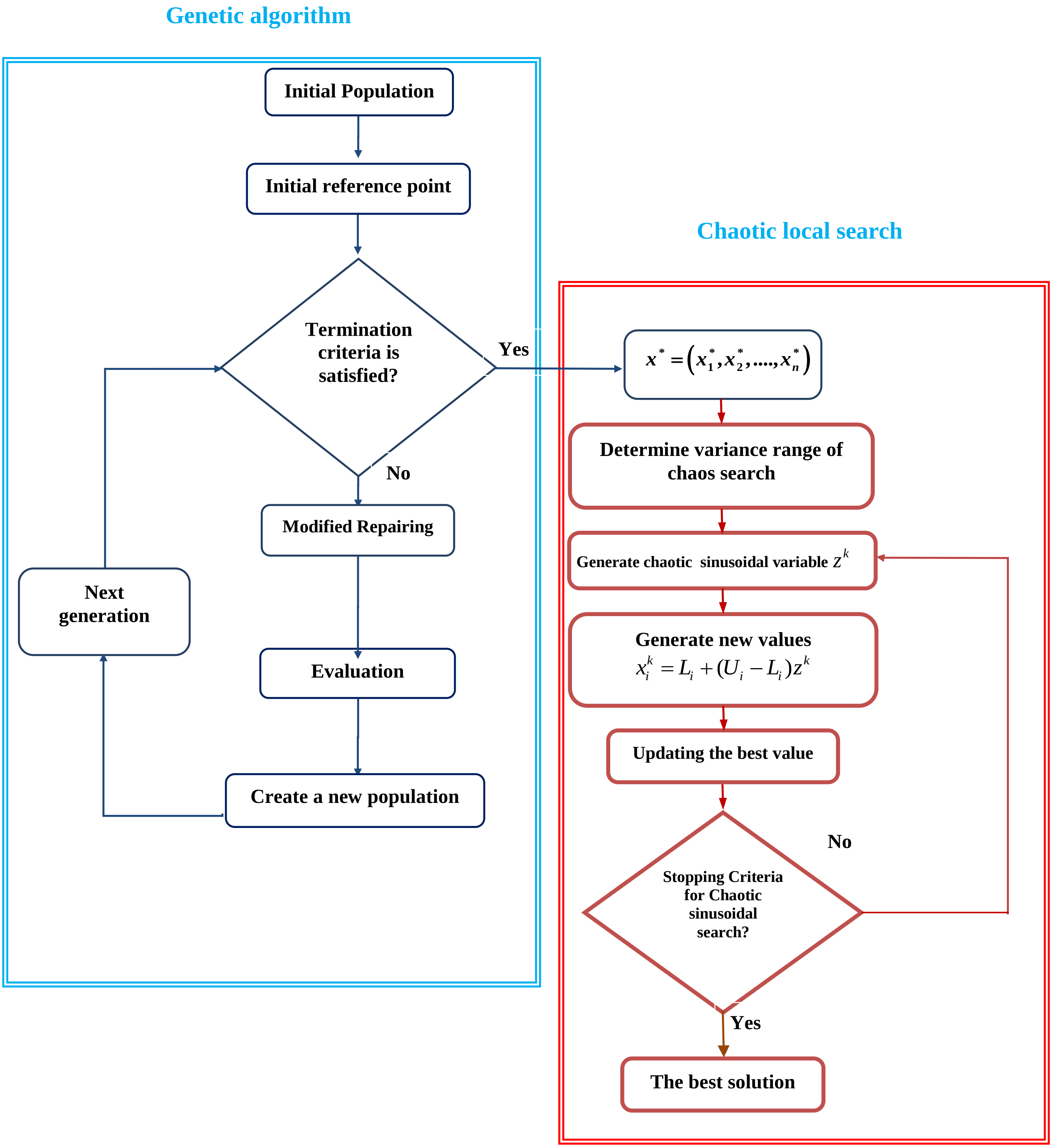
Genetic algorithm
Initial Population
Initial reference point
Termination criteria is satisfied?
Yes
No
Modified Repairing
Next generation
Evaluation
Create a new population
Chaotic local search
$x^* = \left(x_1^*, x_2^*, \ldots, x_n^*\right)$
Determine variance range of chaos search
Generate chaotic sinusoidal variable z^k
Generate new values
$x_i^k = L_i + (U_i - L_i)z^k$
Updating the best value
Stopping Criteria for Chaotic sinusoidal search?
No
Yes
The best solution

Fig. 3. SCGA procedure for nonlinear optimization problems

```
Given $x^* = (x_1^*, x_2^*, ..., x_n^*)$, $\varepsilon$ , and number of chaotic iteration $k$
      While: $f(x^*)$ is improved
          Begin
          $k \leftarrow 1$
                Generate $z^k$ using Sinusoidal map
                $x_i^k = L_i + (U_i - L_i) z^k$
                If $f(x^k) < f(x^*)$
                        then $x^* = x^k$
                    Else if $f(x^k) \geq f(x^*)$
                        continue,
                End if
                If termination criteria satisfied,
                    Break
                End if
          $k \leftarrow k + 1$
    End while
```

Fig. 4. Pseudo code of the algorithm of Sinusoidal map based local search

6. SIMULATION ANALYSIS

In this section, we investigate the validation of the proposed algorithm, where we present eight nonlinear constraint problems [36-40], which was solved using different evolutionary algorithms [14,37,41,42] by Intel Core i5 processors and implemented in MATLAB 12. Also the proposed algorithm was implemented to solve bi-objective real life supply production planning application of minimizing harmful pollution substance and maximizing revenue. The parameters and the operators that are implemented in the simulation run are declared in Table 1.

Table 1. The algorithm parameters/operators

Parameters/Operators	Value/Type
Population size	100
Number of GA iteration	100
Chaos search iteration	30
Chaotic map	Sinusoidal
Crossover type	Single point
Probability of Crossover	0.88
Mutation type	Polynomial mutation
probability of Mutation	0.03
Selection type	Roulette Wheel

M1[14,39]

$$Min \quad f(y) = 0.8356891 y_1 y_5 + 5.3578547 y_3^2$$

$$S.t.$$

$$C_1(y) = 80.51249 + 0.0071317 y_2 y_5 + 0.0029955 y_1 y_2 + 0.0021813 y_3^2$$

$$C_2(y) = 85.334407 + 0.0056858 y_2 y_5 + 0.00026 y_1 y_4 + 0.0022053 y_3 y_5$$

$$C_3(y) = 9.300961 + 0.0047026 y_3 y_5 + 0.0012547 y_1 y_3 + 0.0019085 y_3 y_4$$

$$90 \leq C_1(y) \leq 110, 0 \leq C_2(y) \leq 92, \;\; 20 \leq C_3(y) \leq 25$$

$$78 \leq y_1 \leq 102, \; 33 \leq y_2 \leq 45, \; 27 \leq y_3 \leq 45, \; 27 \leq y_4 \leq 45, \; 27 \leq y_5 \leq 45$$

M2[14,40]

$$\begin{aligned} Min\ \ f(x) = {} & y_1^2 + y_2^2 + y_1 y_2 - 14y_1 - 16y_2 + (y_3 - 10)^2 \\ & + 4(y_4 - 5)^2 + (y_5 - 3)^2 + 2(y_6 - 1)^2 + 5y_7^2 \\ & + 7(y_8 - 11)^2 + 2(y_9 - 11)^2 + (y_{10} - 7)^2 + 45, \end{aligned}$$

Subject to

$$C_1(x) = 105 - 4y_1 - 5y_2 + 3y_7 - 9y_8 \geq 0,$$
$$C_2(x) = -10y_1 + 8y_2 + 17y_7 - 2y_8 \geq 0,$$
$$C_3(x) = 8y_1 - 2y_2 + 5y_9 + 2y_{10} + 12 \geq 0,$$
$$C_4(x) = -3(y_1 - 2)^2 - 4(y_2 - 3)^2 - 2y_3^2 + 7y_4 + 120 \geq 0,$$
$$C_5(x) = -5y_1^2 - 8y_2 - (y_3 - 6)^2 + 2y_4 + 40 \geq 0,$$
$$C_6(x) = -y_1^2 - 2(y_2 - 2)^2 + 2y_1 y_2 - 14y_5 + 6y_6 \geq 0,$$
$$C_7(x) = -0.5(y_1 - 8)^2 - 2(y_2 - 4)^2 - 3y_5^2 + y_6 + 30 \geq 0,$$
$$C_8(x) = 3y_1 - 6y_2 - 12(y_9 - 8)^2 + 7y_{10} \geq 0,$$
$$-10 \leq y_i \leq 10, i = 1, \ldots 10.$$

M3[38]

$$Min\ \ f(x) = 5\sum_{i=1}^{4} y_i - 5\sum_{i=1}^{4} y^2{}_i - \sum_{i=5}^{13} y_i$$

Subject to

$$C_1(y) = 2y_1 + 2y_2 + y_{10} + y_{11} - 10 \leq 0,$$
$$C_2(y) = 2y_1 + 2y_2 + y_{10} + y_{12} - 10 \leq 0$$
$$C_3(y) = 2y_2 + 2y_3 + y_{11} + y_{12} - 10 \leq 0,$$
$$C_4(y) = -8y_1 + y_{10} \leq 0$$
$$C_5(y) = -8y_2 + y_{11} \leq 0, C_6(y) = -8y_3 + y_{12} \leq 0,\ C_7(y) = -2y_4 - y_5 + y_{10} \leq 0,$$
$$C_8(y) = -2y_6 - y_7 + y_{10} \leq 0,\ C_9(y) = -2y_8 - y_9 + y_{12} \leq 0,$$
$$0 \leq y_i \leq 1, i = 1, \ldots 9, 0 \leq y_i \leq 100, i = 10, 11, 12, 0 \leq y_{13} \leq 1.$$

M4[38]

$$\text{Min } f(y) = \left| \frac{\sum_{i=1}^{n} \cos^4 y_i - \prod_{i=1}^{n} \cos^2 y_i}{\sqrt{\sum_{i=1}^{n} i y_i^2}} \right|$$

Subject to $C_1(y) = 0.75 - \prod_{i=1}^{n} y_i \leq 0;\ C_2(y) = \sum_{i=1}^{n} y_i - 7.5n \leq 0;\ 0 \leq y_i \leq 10, i = 1, \ldots, 20.$

M5 [36,14]

$$\begin{aligned}
&Min\quad f_w(y) = 1.10471 y_1^2 y_2 + .04811 y_3 y_4 (14.0 + y_2)\\
&\textit{Subject to}\\
&g_1(y) = 13600 - r(y) \geq 0, g_2(y) = 30000 - \sigma(y) \geq 0,\\
&g_3(y) = x_4 - x_1 \geq 0, g_4(y) = p_c(y) - 6000 \geq 0,\\
&g_5(y) = 0.25 - \delta(y) \geq 0, 0.125 \leq y_1 \leq 10, 0.1 \leq y_2, y_3, y_4 \leq 10.\\
&\text{where,}\\
&r(y) = \sqrt{(r'(y))^2 + (r''(y))^2 + l(r'(y) r''(y)) / \sqrt{0.25 * (y_2^2 + (y_1 + y_3)^2)}}\\
&\sigma(y) = \frac{504000}{y_3^2 y_4}, p_c(y) = 64746.022(1 - 0.0282346t) y_3 y_4^3,\\
&\delta(y) = \frac{2.1952}{y_3^3 y_4}, r'(y) = \frac{6000}{\sqrt{2} y_1 y_2},\\
&r''(y) = \frac{6000(14.0 + 0.5x_2)\sqrt{0.25 * (y_2^2 + (y_1 + y_3)^2)}}{2\{0.707 y_1 y_2 (y_2^2 / 12 + 0.25(y_1 + y_3)^2)\}}.
\end{aligned}$$

M6 [36,14]

$$\begin{aligned}
&Min\quad f(y) = (y_1 - 10)^3 + (y_2 - 20)^3\\
&\textit{Subject to}\\
&g_1(x) = (y_1 - 5)^2 + (y_2 - 5)^2 - 100 \geq 0\\
&g_2(x) = -(y_2 - 5)^2 - (y_1 - 6)^2 + 82.81 \geq 0\\
&13 \leq y_i \leq 100, \quad 0 \leq y_2 \leq 100.
\end{aligned}$$

M7 [36,14]

$$\begin{aligned}
&Min\quad f(x) = (y_1 - 1)(y_1 - 2)(y_1 - 3) + y_3\\
&\textit{Subject to}\quad C_1(x) = y_3 \mathbin{^\wedge} 2 - y_2 \mathbin{^\wedge} 2 - x_1 \mathbin{^\wedge} 2 \geq 0,\ C_2(y) = y_1 \mathbin{^\wedge} 2 + y_2 \mathbin{^\wedge} 2 + y_3 \mathbin{^\wedge} 2 - 4 \geq 0\\
&0 \leq y_1, \ 0 \leq y_2, \quad 0 \leq y_3 \leq 5
\end{aligned}$$

M8 [36,14]

$$\begin{aligned}
&Min\quad f(y) = 5y_1 + 50000 / y_1 + 20y_2 + 75000 / y_2 + 10y_3 + 144000 / y_3\\
&\textit{Subject to}\quad C_1(y) = 1 - 4 / y_1 - 32 / y_2 - 120 / y_3 \geq 0, \quad 10^{-5} \leq y_i, i = 1, 2, 3
\end{aligned}$$

Table 2 summarize the results for different four evolutionary algorithms, where we list the best solution, worst solution, mean, and standard deviations after 20 independent runs for each problem, the obtained results by applying the proposed algorithm are better than the corresponding ones obtained from other algorithms.

Table 2: statistical analysis of the proposed algorithm versus other evolutionary algorithms

Function	Optimal	Status	Evolutionary Algorithms				
			Proposed algorithm	[37]	[12]	[14]	[18]
M1	-30665.55	Best solution	-30665.55	-30665.55	-30665.51	-30665.35	-30665.53
		Mean	-30666.51	-30666.31	-30666.26	-30666.31	-30666.53
		Worst solution	-30665.16	-30665.53	-30665.33	-30665.23	-30665.53
		St. Dev.	6.1E-06	5.1E-02	4.3E-04	4.2E-01	5.1E-09
M2	24.8641	Best solution	24. 64	24.8641	24.9631	24.9641	24.3062
		Mean	24.641	24.6621	24.7691	24.8621	24.4312
		Worst solution	24. 8641	24.6521	24.6601	24.6532	24.6721
		St. Dev.	2E-04	3.3E-02	4.3E-02	4.2E-03	5.1E-01
M3	-15	Best solution	-15	-15	-15	-15	-15
		Mean	-15	-15	-15	-15	-14.492
		Worst solution	-15	-15	-15	-15	-14.354
		St. Dev.	0	0	0	0	9E-01
M4	-0.80325	Best solution	-0.80325	-0.80325	-0.81325	-0.83215	-0.80315
		Mean	-0.80325	-0.81325	-0.81212	-0.84325	-0.81325
		Worst solution	-0.80325	-0.82325	-0.81532	-0.86312	-0.82335
		St. Dev.	0	5.3E-02	4.6E-04	4.7E-02	5.7E-03
M5	2.381021	Best solution	2.381021	2.381021	2.38302	2.381054	2.38104
		Mean	2.382041	2.392041	2.38405	2.381092	2.38710
		Worst solution	2.383061	2.393061	2.38602	2.381063	2.3900
		St. Dev.	4.1E-05	9.3E-06	5.3E-04	3.2E-03	3.2E-05
M6	-6961.81	Best solution	-6961.81	-6961.71	-6961.35	-6961.51	-6961.34
		Mean	-6961.61	-6961.53	-6961.21	-6961.42	-6961.21
		Worst solution	-6961.21	-6961.45	-6961.01	-6961.34	-6961.12
		St. Dev.	5E-04	3E-04	1.2E-03	2.2E-05	1.7E-04
M7	-4.5857	Best solution	-4.5857	-4.5807	-4.5432	-4.5800	-4.5850
		Mean	-4.5502	-4.5750	-4.5401	-4.5741	-4.5843
		Worst solution	-4.5403	-4.5634	-4.5390	-4.5707	-4.5821
		St. Dev.	2E-06	3.3E-04	3.1E-03	1.4E-02	2.2E-04
M8	6299.84	Best solution	6299.6	6299.84	6299.73	6300.10	6299.01
		Mean	6299.74	6299.70	6299.72	6300.04	6299.03
		Worst solution	6299.84	6299.04	6299.43	6301.21	6299
		St. Dev.	2.1E-04	4.1E-04	5.2E-03	5.1E-04	2.2E-03

Figures 5-12 illustrate the convergence analysis of the proposed algorithm for these eight problems

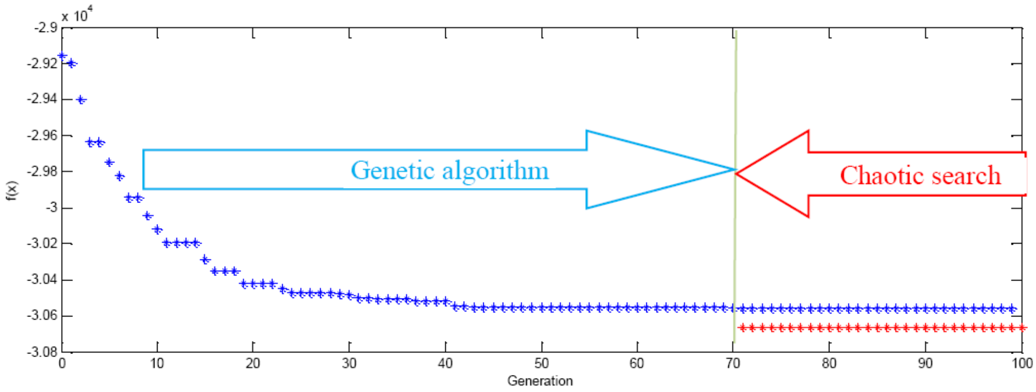

Fig. 5. Convergence analysis for problem M1

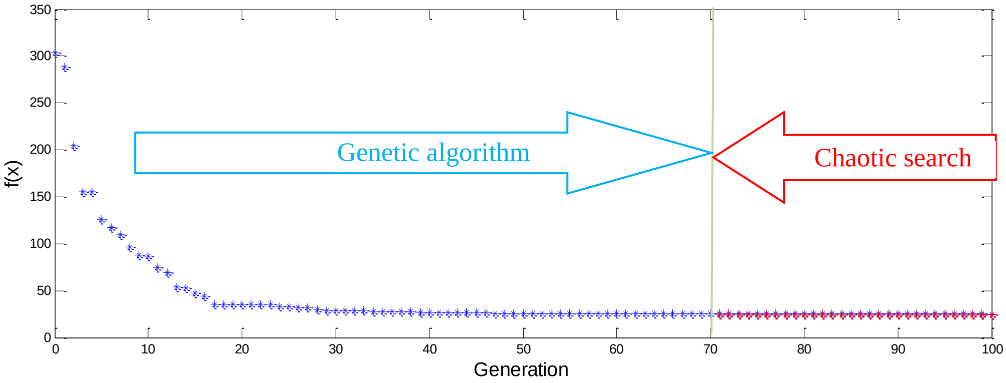

Fig. 6. Convergence analysis for problem M2

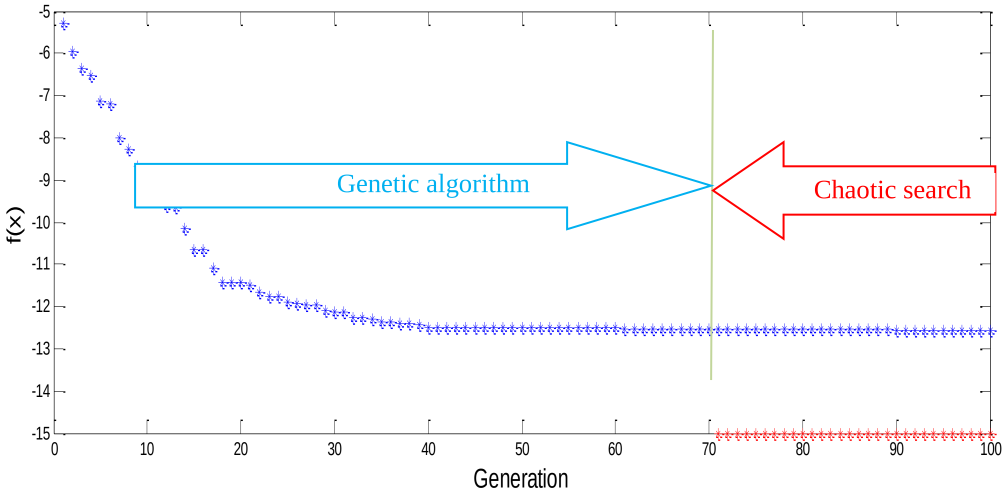

Fig.7. Convergence analysis for problem M3

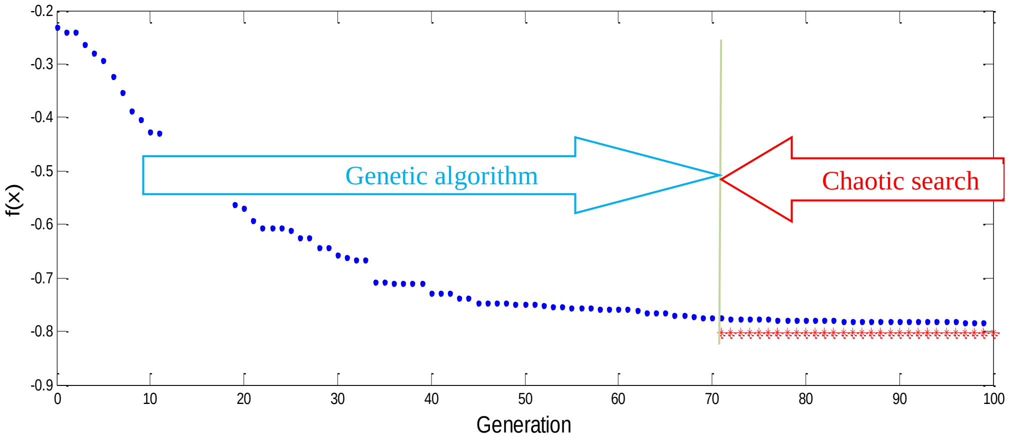

Fig.8. Convergence analysis for problem M4

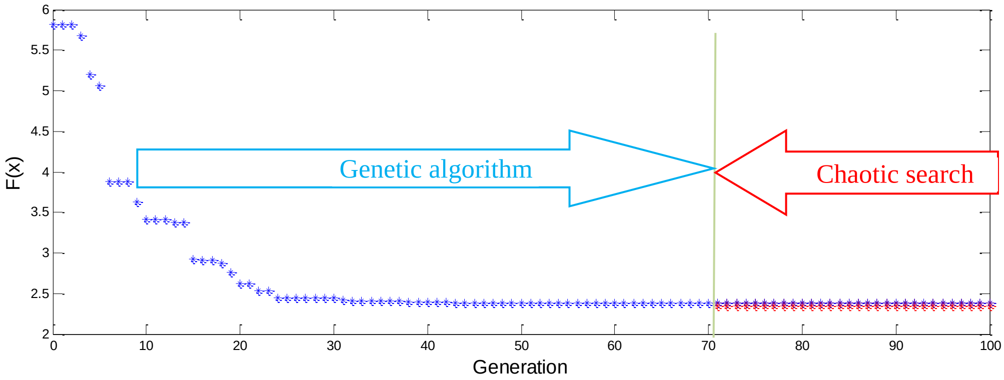

Fig. 9. Convergence analysis for problem M5

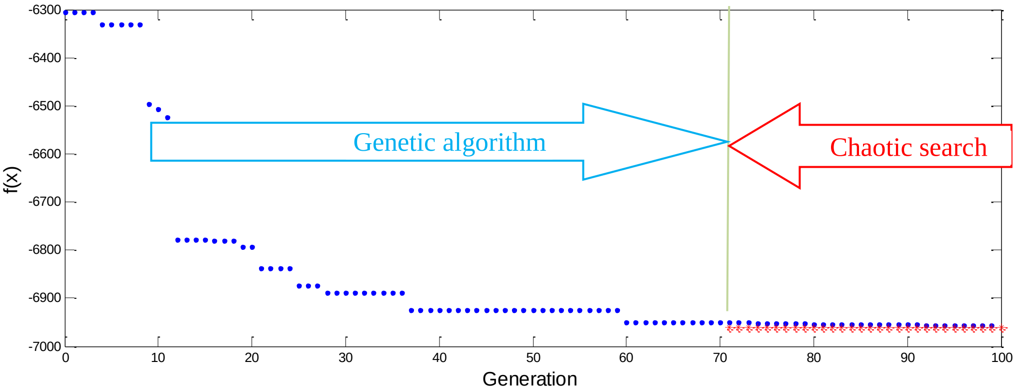

Fig. 10. Convergence analysis for problem M6

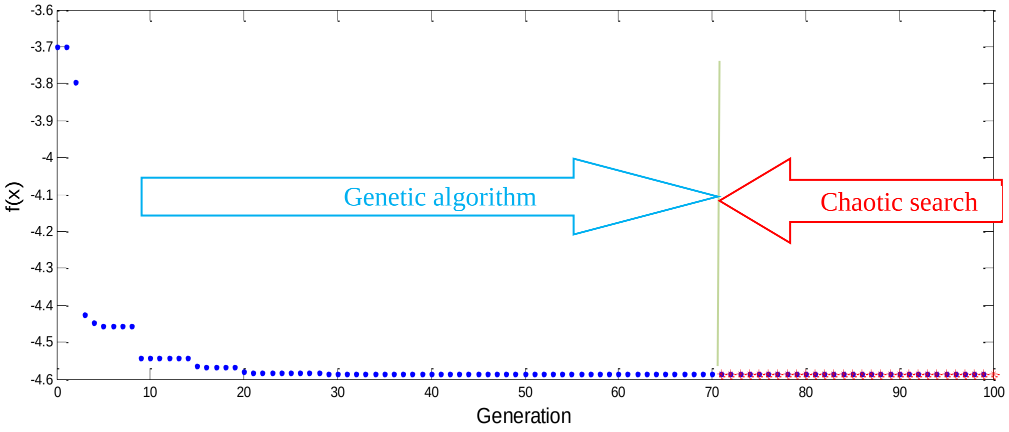

Fig. 11. Convergence analysis for problem M7

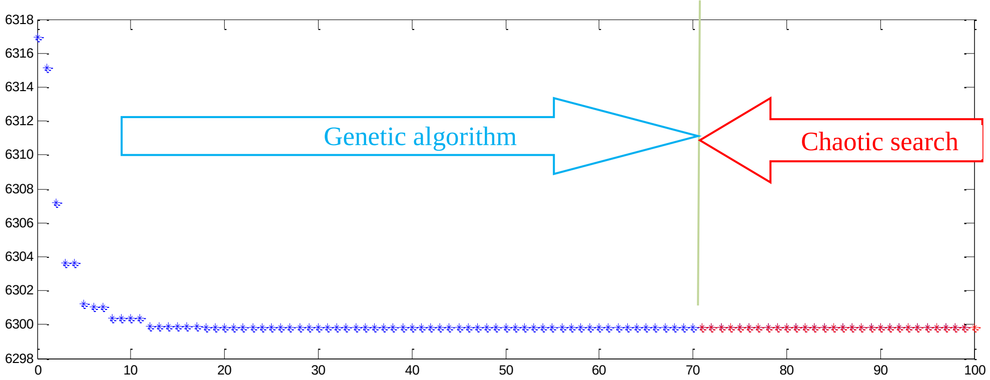

Fig. 12. Convergence analysis for problem M8

Table 3 gives the percentage saving of time computation due to using chaotic local search; the obtained results declare that using chaotic local search saving around 30% as an average for all eight problems. It was emphasized that significant savings can be achieved via the chaotic local search, which evolves the total performance of the proposed algorithm.

Table 3: Percentage saving for test problems

Problem	M1	M2	M3	M4	M5	M6	M7	M8
Percentage saving	70%	30%	47.5%	36%	36%	22%	0%	0%

Fuzzy Bi-Objective Linear Programming [34,44]:

A certain factory creates 3 different outputs A, B and C. To create 1 unit of A it requires 2 ton of raw matter I, 3 ton of raw matter II, and 4 ton of raw matter III To create 1 unit of output B it requires 8 ton of raw matter I, and 1 ton of raw matter II. To create 1 unit of output C it requires 4 ton of raw matter I, 4 ton of raw matter II, and 2 ton of raw matter III. At present prices, the company anticipates to market output A at a price of 5×10^6 \$/unit, output B at a price of 10×10^6 \$/unit, and output C at a price 12×10^6 \$/unit. However, over production operation, creating 1 unit of output A gives 1 ton of hurtful pollution material, creating 1 unit of output B gives 2 ton of hurtful pollution material, and creating 1 unit of output C gives 2 ton of hurtful pollution material. he goals are to maximize total income and to minimize total given hurtful pollution material. The availability of raw materials are 100 ton, 50 ton, and 50 ton for Raw materials I,II, and III respectively.

This real world problem can be mathematically formulated as bi-objective linear programming as following:

$$\begin{aligned}
&Max\ \ F_0 = 5y_1 + 10y_2 + 12y_3 (revenue)\\
&Min\ \ F_1 = y_1 + 2y_2 + 2y_3 (pollution)\\
&s.t.\\
&2y_1 + 8y_2 + 4y_3 \le 100,\ 3y_1 + y_2 + 4y_3 \le 50,\ 4y_1 + 2y_3 \le 50\\
&y_1, y_2, y_3 \ge 0
\end{aligned}$$

This can be written as follows:

$$Max\ \ F_0 = cy\,,\ Min\ \ F_1 = dy\,,\quad s.t.\ \ Ay \le b,\ \ y \ge 0$$

Where,

$$x = [y_1 \quad y_2 \quad y_3]^T,\ c^T = \begin{bmatrix} 5 \\ 10 \\ 12 \end{bmatrix},\ d^T = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$$

$$A = \begin{bmatrix} 2 & 8 & 4 \\ 3 & 1 & 4 \\ 4 & 0 & 2 \end{bmatrix},\ b = [100 \quad 50 \quad 50]^T$$

By solving each objective separately using Linear Programming Calculator [45], we get:

$$Case\ 1: Max\ \ F_0 = cy\ \ \ s.t.\ Ay \le b,\ \ y \ge 0.$$

$$y = (0, 7.14, 10.71),\ F_0 = 200,\ F_1 = 35.71$$

$$Case\ \ 2: Min\ F_1 = dy\ \ \ s.t.\ Ay \le b,\ \ y \ge 0.$$

$$y = (0, 0, 0),\ F_0 = 0,\ F_1 = 0$$

In problem (1), Maximum income is 200 million \$, but this solution gives 35.71 ton of hurtful pollution material.

In problem (2), Minimum hurtful pollution material is 0 ton, which would mean 0 \$ of total income. As we can observe, these two goals (Maximum income and Minimum hurtful pollution material) are in conflict with each other (i.e., when maximize total income, hurtful pollution material increases, also when Minimize hurtful pollution material, total income decreases).

To get an adjustment solution with respect to opacity and satisfaction degree, two aims were defined as follows. [43] :

Aim 1 : Must obtain at least 75% of maximum total income (150×10^6 \$), but we would prefer to get 100% of maximum total income (200×10^6 \$).

Aim 2 : The total pollution matter must not exceed 30 ton , but we would prefer to get ton of pollution matter.

These two aims can be mathematically modelled into fuzzy nonlinear programming using modified parabolic membership function as follows:

Let μ is the degree of membership function.

For goal 1, the value in the interval [150,200] would be described by a nonlinear function (Parabolic function) $(F_0 - 150)^2 / 2500$, thus we get the following membership function which declared in figure 13

$$\mu(F_0) = 0,\ \text{when}\ \ 0 \le F_0 \le 150\ \ \text{and}\ \mu(F_0) = \frac{(F_0 - 50)^2}{2500}\ \text{when}\ 150 \le F_0 \le 200$$

As declared in figure 13.

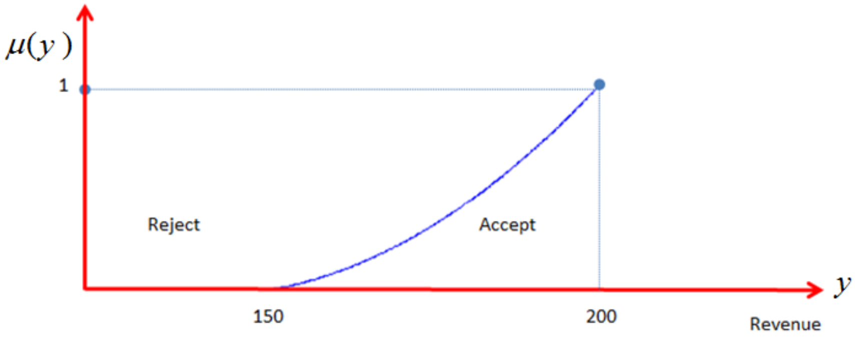

Fig. 13. Parabolic Membership Function for Revenue

For goal 2, the value in the interval [0,30] would be described by a nonlinear function (Parabolic function) $1-\sqrt{\frac{F_1}{30}}$, thus we get the following membership function as declared in figure 14

$$\mu(F_1)=0,\ \text{when}\ \ 30\le F_1\ \ \text{and}\ \mu(F_1)=1-\sqrt{\frac{F_1}{30}}\ \text{when}\ 0\le \text{F}_1\le 30$$

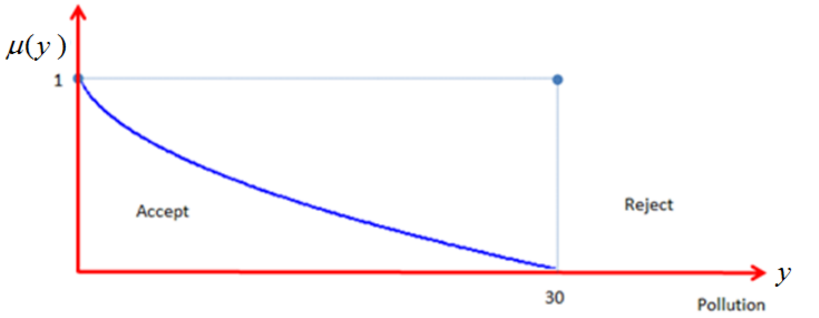

Fig. 14. Parabolic Membership Function for pollution

By implementing the concept of fuzzy membership function, the objectives of our multi-objective problem will be defined as follows

$$Max\ \{\min\{F_0,F_1\}\}s.t.\ \text{Ay}\le \text{b,y}\ge 0.$$

This would means that we want to find the value of which is dependent upon which would maximize membership to both sets, since $F_0=cy\,,F_1=dy$ we get the following nonlinear programming problem:

$$Max\ \ \lambda$$

$$s.t.$$

$$\lambda\le\frac{(5y_1+10y_2+12y_3-150)^2}{2500},$$

$$\lambda\le 1-\sqrt{\frac{y_1+2y_2+2y_3}{30}}$$

$$2y_1+8y_2+4y_3\le 100$$

$$3y_1+y_2+4y_3\le 50$$

$$4y_1+2y_3\le 50$$

$$y_1,y_2,y_3\ge 0$$

Solving this nonlinear programming problem using our proposed approach, we obtain:

$$\lambda = 0.0461, y_1 = 0, y_2 = 1.5331, y_3 = 12.1167$$

Figure 15 represent a convergence curve for the proposed approach

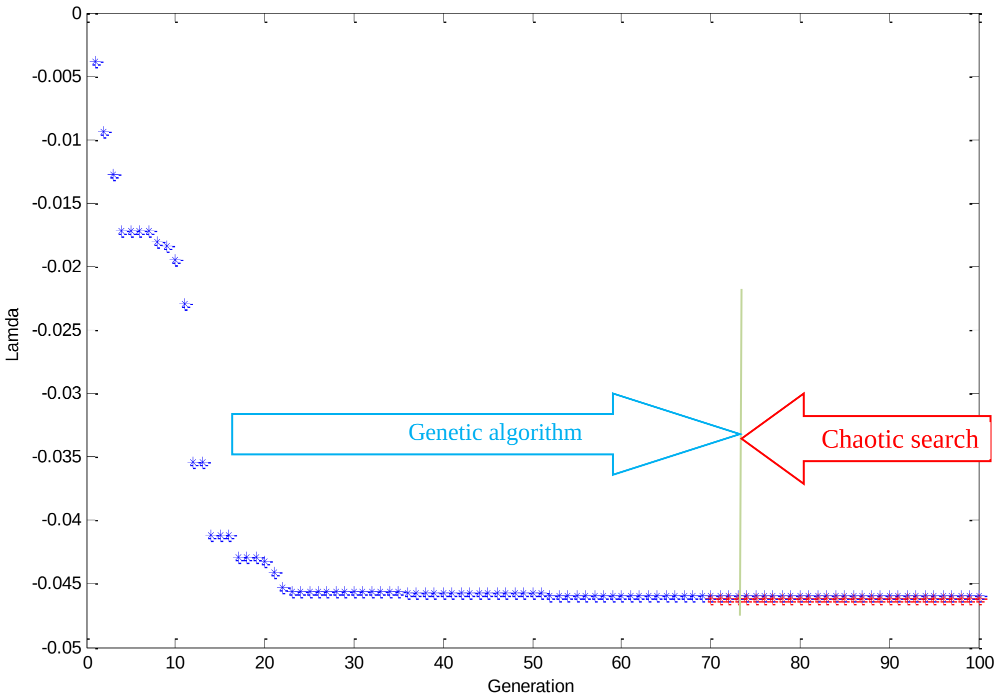

Fig. 15. Convergence analysis for real application

In general, this real life application has carried out its aims by forming mathematically new modified membership function (parabolic function) and in implementation it in a limited resources production planning problem. Based on the results analysis of bi-objective real life resource production planning application with minimizing harmful pollution substance and maximizing revenue, the following points are drawn:

1. The proposed mathematical model can be applied to higher dimension problem with many objectives by incorporating only one additional nonlinear constraint for each additional objective function.
2. The proposed model using modified membership function can be extended to any real application in any fields of engineering, and science, with no modifications.
3. Investigation of results refer to that the total income and the total hurtful pollution substance in the proposed model are increased by 3% and decreased by 4% respectively as compared to maximum membership values in the fuzzy decision with linear member ship function (Linear Programming Model) [43].

7. CONVERGENCE ANALYSIS OF THE SUGGESTED METHOD

To compute the computational time comparison and to monitor the convergence behavior of the proposed algorithm, this analysis is presented. On the first hand, to make comparison in the view of the speed of the convergence, we must define how to measure the execution period. The number of generation (iterations) cannot be applicable as measurement of execution period, and this is due to different algorithms execute different procedure of works in their inner loops and have different parameters and different operators (population size, crossover probability, mutation probability,…etc.), in our comparison, the number of fitness evaluations (objective function) (FEs) was implemented as a measure of the executed computational time for each algorithm, which can better reflect its real running time. On the other hand, the search space was shrunken after phase I concentrating the optimal solution region. So the proposed algorithm converges in accelerated manner to the optimal solution. In addition, the suggested hybrid method enables us to come closely to the optimal solution than the previous techniques based on evolutionary algorithm. In brief, this proposed algorithm in phase I have capability to adjust) its searching region (to shrunk the search space) for the second phase Table 3 gives the percentage saving of time computation; the obtained results declare that using the proposed approach saving around 30% as an average. Tables 3 also declare that the values reported by SCGA gives a valuable decrease in execution period compared to the proposed approach without Sinusoidal chaotic. Finally, the main features of some literature algorithms were described in Table 4. It is concluded that the proposed algorithm exploits the advantages of chaotic mapping and global search capability of GAs for rapidly convergence to the optimal solution.

Table 4. Comparing some optimization algorithm

Features	Classical methods	Local search methods	Evolutionary algorithms	Chaotic methods	Proposed hybrid algorithm
Highly computational time	×	×	✓	×	×
Sensitivity to initial conditions	✓	✓	✓	×	×
Initial parameters requirement	×	×	✓	×	✓
Global search ability	×	×	✓	✓	✓
Local search capability	✓	✓	×	×	✓

The most effective reason of decreasing execution period of executing the suggested method if compared with the other algorithms is that the proposed hybrid algorithm keep away from the formal seeking in the universe of the problem and to concentrate the search in a specified local region.

8. CONCLUSION

This research presents a hybrid genetic algorithm- Sinusoidal chaotic Map to solve general constrained NLP. Regrettably, detecting a universal optimum solution has no known method to the general constrained NLP. The proposed algorithm is consists of a classical genetic algorithm coupled with a Sinusoidal chaotic Map based local search. It has been concluded that coupling with chaotic mapping save executing time, which can better reflect its real running period and accelerate convergence of the algorithm to the global solution. It has been also known that, the sensibility to the initial state is one of the most important characters of chaotic mapping. This feature ensures that there are no two identical new solutions obtained which ensures the population diversity and improve the global searching capability by escaping the local solutions and prevent to get stuck in local optima. Several problems from the literature enabled us to measure our results with respect to other known algorithms. Via the numerical analysis, we can conclude that our proposed combined algorithm is very robust and give very accurate final solution. The basic characteristics of the suggested hybrid technique could be recorded as written in the following points.

1. The proposed algorithm has been robustly applied to solve the constrained NLP with different type of constraint.
2. The proposed algorithm is efficient and robust for solving nonconvex, discontinuous and non-differentiable optimization problems
3. On the basis of the real life application, it was concluded that the proposed hybrid algorithm can provide a true optimal solution by considering conflicting bi-objective functions.

For future work, we suggest to apply the proposed algorithm on highly dimension real-world applications.

ACKNOWLEDGMENT

Prof. Mousa, A. is thankful to the great support of Deanship of Scientific Research, in Taif University for funding the project of “Future Researcher Program” in the project No.1-439-6092.

REFERENCES

[1] Forst W., Hoffmann D., (2010), "Optimization—Theory and Practice", Springer Science & Business Media, New York, first edition.

[2] Craven B. D., (1995), "Control and Optimization", Chapman and Hall, London, first edition.

[3] Rao S.S., (2009), "Engineering Optimization: Theory and Practice", A Wiley-Interscience publication, New Jersey, fourth edition.

[4] Kuma A., (2011), "Network Design Using Genetic Algorithm ", A Thesis submitted to saurshtra university, Rajkot, India.

[5] Biggs M. B., (2008), "Nonlinear Optimization with Engineering Applications ", Springer, Verlag , first edition.

[6] Azzam M. and Mousa A. A., (2010), "Using genetic algorithm and TOPSIS technique for multiobjective reactive power compensation", Electric Power Systems Research, 80(6), 675–681

[7] Mousa A. A., (2010), "Using genetic algorithm and TOPSIS technique for multiobjective transportation problem: a hybrid approach", International Journal of Computer Mathematics, 87(13), 3017-3029

[8] Mousa A.A., El-Shorbagy M.A. and Abd-El-Wahed W.F., (2012) ,"Local search based hybrid particle swarm optimization algorithm for multiobjective optimization", Swarm and Evolutionary Computation, 3, 1–14.

[9] Mousa A. A., Abd El-Wahed Waiel F., Rizk-Allah R. M., (2011), "A hybrid ant colony optimization approach based local search scheme for multiobjective design optimizations", Electric Power Systems Research, 81 , 1014–

[10] Onwubolu G.C., B.V.Babu, (2004), "New optimization Techniques in Engineering ", Springer Science & Business Media, Berlin, first edition.

[11] Brooks S.P., Morgan B.J.T, (1995), "optimization using simulated annealing", Journal of the Royal Statistical Society, 44(2), 241-257.

[12] Alrefaei M. H. and Diabat A.H., (2009), "A simulated annealing technique for multi-objective simulation optimization", Applied Mathematics and Computation, 215 (8), 3029-3035.

[13] M.A. El-Shorbagy, A.A. Mousa and M. A. Farag, (2017) Solving Nonlinear Single-Unit Commitment Problem by Genetic Algorithm Based Clustering Technique , Review of Computer Engineering Research, Vol. 4, No. 1, 11- 29

[14] Osman M.S., Abo-Sinna M.A., and Mousa A.A., (2005) "A Combined Genetic Algorithm-Fuzzy Logic Controller (GA-FLC) In Nonlinear Programming", Journal Of Applied Mathematics & Computation (AMC), 170(2), 821- 840.

[15] Yang X., Yuan J., Yuan J., and Mao H., (2007), "A modified particle swarm optimizer with dynamic adaptation", Applied Mathematics and Computation, 189(2),1205-1213.

[16] M.A. El-Shorbagy, A.A. Mousa, Chaotic Particle Swarm Optimization For Imprecise Combined Economic And Emission Dispatch Problem, Review of Information Engineering and Applications, Vol. 4, No. 1, 20- 35

[17] A.,A. mousa, M elsisy, M s Osman (2013), Ant colony algorithm to support multi-criteria decision making", ISBN 978-3-659-32175-7, Lambert academic publishing GmbH& Co.kG, Berlin.

[18] A.A.Mousa, I.M.El_Desoky (2013), Stability of Pareto optimal allocation of land reclamation by multistage decision-based multipheromone ant colony optimization, Swarm and Evolutionary Computation, Volume 13, Pages 13–21 .

[19] Hopfield J. J. and Tank D. W., (1985) "Neural‖ Computation of Decisions in Optimization Problem", Biological Cybernetics, 52, 141-152

[20] Liu M., Wang R., Wu J. and Kemp R., (30 May- 1 June 2005), "A Genetic-Algorithm-Based Neural Network Approach for Short-Term Traffic Flow Forecasting", Advances in Neural Networks- Second International Symposium on Neural Networks, China.

[21] Donis-Díaz C.A., Muro A.G. and Bello-Pérez R. and Morales E.V., (2014), "A hybrid model of genetic algorithm with local search to discover linguistic data summaries from creep data", Expert Systems with Applications, 41(2), 2035–2042

[22] Yang K., Liu Y. and Yang G., (2013) "Solving fuzzy p-hub center problem by genetic algorithm incorporating local search", Applied Soft Computing, 13(5), 2624– 2632

[23] Hénon M., (1976), "A two-dimensional mapping with a strange attractor", Communications in Mathematical Physics, 50(1), 69- 77

[24] Lorenz E.N., (1993), "The Essence of Chaos", University of Washington Press, USA, first edition

[25] Xiao J., (2009) "Improved Quantum Evolutionary Algorithm Combined with Chaos and Its Application", Springer, Berlin, Heidelberg, First edition.

[26] Alatas B., Akin E., Ozer A. B., (2009). "Chaos embedded particle swarm optimization algorithms", Chaos, Solitons and Fractals, 40(4), 1715- 1734

[27] Zelinka I., Celikovsky S., Richter H. and Chen G., (2010), "Evolutionary Algorithms and Chaotic Systems", Springer, Berlin, Heidelberg, first edition.

[28] Coelho L., Ayal1a H., Mariani V. (2014), "A self-adaptive chaotic differential evolution algorithm using gamma distribution for unconstrained global optimization", Applied Mathematics and Computation – 234,452-459.

[29] Haupt R. L., Haupt S.E., (2004), "practical genetic algorithm", A Wiley-Interscience publication, New Jersey, second edition.

[30] Peitgen H., Jurgens H., Saupe D., (2004), "Chaos and Fractals", Spring Science & Business Media, New York, second edition.

[31] Osman M.S., M.A.Abo-Sinna, and A.A. Mousa "Epsilon-Dominance based Multiobjective Genetic Algorithm for Economic Emission Load Dispatch Optimization Problem, Electric Power Systems Research 79 (2009) 1561–1567.

[32] M.S.Osman , M.A.Abo-Sinna , and A.A. Mousa An Effective Genetic Algorithm Approach to Multiobjective Resource Allocation Problems (MORAPs) " Journal Of Applied Mathematics & Computation 163, pp. 755- 768.

[33] MA El-Shorbagy, AY Ayoub, AA Mousa, IM El-Desoky, An enhanced genetic algorithm with new mutation for cluster analysis , Computational Statistics, 1- 38.

[34] MA El-Shorbagy, MA Farag, AA Mousa, IM El-Desoky, (2019), A Hybridization of Sine Cosine Algorithm with Steady State Genetic Algorithm for Engineering Design Problems, International Conference on Advanced Machine Learning Technologies and Applications,Springer, Cham,143-155.

[35] M.A. EL-Shorbagy, A.A. Mousa,S.M. Nasr (2016), A chaos-based evolutionary algorithm for general nonlinear programming problem, Chaos, Solitons and Fractals , Volume 85, Pages 8-21 .

[36] K. Deb,(1991) "Optimal Design of A Welded Beam Structure Via Genetic Algorithms", ALAA journal 29/11,pp 2013- 2015

[37] FA El-Qorashy, HA Nabwey, AA Mousa , (2015), A Combined Genetic Algorithms-local Search Engine (GAs–LCE) in Constrained Nonlinear Programming , British Journal of Applied Science & Technology 8 (3), 324- 333

[38] C. Floundas and P.Pardalos (1987) , A collection of test problems for constrained global optimization, lecture Notes in computer sciences. Berlin: Sprringer-Verlag.

[39] D Himmelblau. Applied nonlinear programming (1972), NewYork , McGraw-hill , IEEE press.

[40] W. Hock and K. schittkowski,(1981), test examples for nonlinear programming codes , berlin, springer-verlag, lecture notes on Econ. And Math. Syst. Vol. 187

[41] M. S. Osman, M. A. Abo-Sinna, A. A .Mousa,(2004)" A Solution to The Optimal Power Flow Using Genetic Algorithm", Applied mathematics and computation, 155(2), 391-405.

[42] Waiel. F. Abd El-Wahed, A.A.Mousa , M. A. El-Shorbagy,,(2011), Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems, Journal of Computational and Applied Mathematics 235(5)1446– 1453

[43] Kirsch, D. A., (2003), Fuzzy sets in multiobjective optimization. http://members.tripod.com/Kirsch/Fuzzy.

[44] Pandian Vasant , Arijit Bhattachary (2007) , Application of multi-objective fuzzy linear programming in supply production planning problem ,The International Journal of Industrial Engineering: Theory, Applications and Practice ,14(2),151-157.

[45] Linear Programming Calculator, (April 2019), https://comnuan.com/cmnn03/cmnn03004/cmnn03004.php.