A dynamic adaptive PSO based on chaotic search for constrained multiobjectuve optimization Problems`

Abdallah A. Mousa[1,2], B. N. Al-Matrafi[1] ,Abdallah A. Galal[1,3]
[1]Department of Mathematics and Statistics, Faculty of Sciences, Taif University, Taif, KSA
[2]Department of Basic Engineering Science, Faculty of Engineering, Menoufia University, Shibin Al Kom, Egypt
[3]Physics and Engineering Mathematics Department, Faculty of Engineering, Tanta University, Tanta, Egypt

Abstract—. In this paper, a new enhanced particle swarm optimization method with dynamic adaptation is presented. To enrich the searching behaviour and to avoid being trapped into local optimum, dynamic adaptive of PSO parameters are incorporated into PSO. In this algorithm, inertia weight of each particle is dynamically updated, also personal influence and social influence parameters are dynamically adapted during the process. Moreover, a modified velocity updating formula of the particle is presented, where, a new constriction factor which control the feasibility of the particles is presented. The proposed approach can be viewed as the global optimization algorithm while the chaotic search CS is employed for the local search. Thus, the possibility of exploring a global minimum in problems with many local optima is increased. The proposed method can obtain the global optimal results quickly, due to fast globally converging characteristics of PSO and the effective local search ability of CS. The results, provided by the proposed algorithm for benchmark engineering problems, are promising when compared with exiting well-known algorithms. Also, our results suggest that our algorithm is better applicable for solving real-world application problems.

Index Terms—multiobjective optimization, particle swarm, local search; chaos

1 Introduction

The In multiple objectives optimization, several conflicting objectives have to be minimized simultaneously. Generally, no unique solution exists but a set of mathematically equally good solutions can be identified, by using the concept of Pareto optimality. Many applications of constrained optimization can be found in engineering [1–2], economics and finance [3], medicine [4-5], management and planning [6], etc. There exist many solution strategies to solve the constrained optimization problems. One of the basic approaches is the weighting method [7], where one single-objective optimization problem is formed by weighting several objective functions. Similar problem has the ε-constraint method, introduced in [7].

Recently, there has been a boom in applying evolutionary algorithms to solve multiobjectuve optimization problems [8–11]. Evolutionary algorithms (EAs) are stochastic search methods that mimic the metaphor of natural biological evolution and/or the social behavior of species. The development of metaheuristic optimization theory has been flourishing. Many metaheuristic paradigms such as genetic algorithm [8,12,13], simulated annealing [14], tabu search [14,15], and ant colony algorithm[16] has become an interesting approach to solve many hard problems. Recently particle swarm optimization PSO [11,17] have shown their efficacy in solving computationally intensive problems. Animals, especially birds, fishes etc. always travel in a group without colliding, each member follows its group, adjust its position and velocity using the group information, because it reduces individual's effort for search of food, shelter etc. Particle swarm optimization is evolutionary technique similar to genetic algorithm because both are population based and are equally affective. Particle swarm optimization has better computational efficiency, i.e. it requires less memory space and lesser speed of CPU, it has less number of parameters to adjust. Genetic algorithm and other similar techniques (e.g. simulated annealing), work for discrete design variables, whereas particle Swarm optimization work for discrete as well as analogue systems, because it is inherently continuous, does not need D/A or A/D conversion. Although for handling discrete design variables Particle swarm optimization needs some modification to be done in particle swarm optimization methods. PSO is an evolutionary computation technique, developed for optimization of continuous nonlinear, constrained and unconstrained, non differentiable multimodal functions [18]. PSO is inspired firstly by general artificial life, the same as bird flocking, fish schooling and social interaction behaviour of human and secondly by random search methods of evolutionary algorithm [19]. In [20], a dynamic multi-swarm particle swarm optimizer (DMS-PSO) was proposed whose neighborhood topology is dynamic and randomized. DMS-PSO gives a better performance on multimodal problems than some other PSO variants, but the local search performance is not satisfactory. Dynamic pattern are challenging for PSO, a self-adapting multi-swarm has been derived [21]The multi-swarm with exclusion has been favorably compared, on the moving peaks problem, to the hierarchical swarm, PSO re-initialization and a state-of-the-art dynamic-optimization evolutionary algorithm known as self-organizing scouts. To deal with discrete events, an algorithm based on discrete developed in [22]. This approach solves the overlapping coalition formation problem in multiple virtual organizations. Recently, in [23] presented a PSO method demonstrating a significant performance improvement over the SPSO, QIPSO, UPSO, FIPS, DMSPSO, and CLPSO algorithms. Because the proposed method utilizes fuzzy set theory for the adaptation of parameters, it is referred to as the adaptive fuzzy PSO (AFPSO). To enrich the searching behaviour and to avoid being trapped into local optimum, dynamic adaptive of PSO parameters are incorporated into PSO. PSO has very fast converging characteristics at the beginning; however it has slow local search ability, so chaotic local search are incorporated to explore a large space and the best value of all the particles will be taken as the initial starting point for the CS. Thus, the possibility of exploring a global minimum in problems with more local optima is increased. The search will continue until a termination criterion is satisfied.

In this paper, a modified adaptive particle swarm optimizer with dynamic adaptation is proposed. To enrich the searching behaviour and to avoid being trapped into local optimum, dynamic adaptive of PSO parameters are incorporated into PSO. Implementation of dynamic adaption of PSO parameters aim to force the algorithm to converge rapidly. Moreover, a modified velocity updating formula of the particle is presented. The results, provided by the proposed algorithm for benchmark problems, are promising when compared with exiting well-known algorithms. Based on the numerical simulation, some conclusions are drawn and some points for future work is established.

2. Multiobjective optimization (MO)

A general optimization (minimization) problem [7] of M objectives can be mathematically stated as:

$$\left.\begin{array}{ll}\text{Minimize} & \overline{f}\left(\overline{x}\right)=\left[f_i\left(\overline{x}\right),\ i=1,2,\dots,M\right]\\ \text{subject to} & g_j\left(\overline{x}\right)\le 0,\ j=1,2,..,J.\end{array}\right\} \quad (1)$$

given $\overline{x}=\left[x_1,x_2,\dots,x_n\right]$, where n represents the dimension of the decision variable space, $f_i\left(\overline{x}\right)$ is the i-th objective function, $g_j\left(\overline{x}\right)$ is the j-th inequality constraint. The MO problem then reduces to finding an $\overline{x}$ such that $f_i\left(\overline{x}\right)$ is optimized. Since the notion of an optimum solution in MO is different compared to the single objective optimization (SO), the concept of Pareto dominance is used for the evaluation of the solutions.

Definition 1: (Pareto dominance). A vector $\overline{u}=\left(u_1,u_2,\dots,u_M\right)$ is said to dominate a vector $\overline{v}=\left(v_1,v_2,\dots,v_M\right)$ (u dominate v denoted by $\overline{u}$ f $\overline{v}$), for a MO minimization problem, if and only if

$$\forall i\in\{i,\dots,M\},\ u_i\le v_i\wedge\exists i\in\{i,\dots,M\}:u_i<v_i$$

where M is the dimension of the objective space.

Definition 2: (Pareto optimality). A solution $\overline{u}\in U$, where U is the universe, is said to be Pareto optimal if and only if there exists no other solution $\overline{v}\in U$, such that u is dominated by $\overline{v}$. Such solutions u are called nondominted solutions. The set of all such nondominted solutions constitutes the Pareto-Optimal Set.

Definition 2: (Ideal objective vector) : An objective vector minimizing each of the objective functions is called an ideal (perfect) objective vector. The component z_i of the ideal objective vector $z^I\in R^M$ are defined by minimizing each of the objective functions individually subject to the constraints, that is, by solving

$$\begin{array}{l}Min\ f_i(x)\\ s.t.\ \ x\in S\qquad for\ \ i=1,\dots,M\end{array}$$

From the ideal objective vector we obtain the lower bounds of the Pareto optimal set for each objective function.

3. The particle swarm optimization

Particle swarm optimization (PSO) is an evolutionary computation algorithm motivated by the simulation of social behavior [24-26]. Namely, each agent (individual) utilizes two important kinds of information. The first one is their own experience (Cognitive); that is, they have tried the choices and know which state has been better so far, and they know how good it was. The second one is other agent's experiences (social); that is, they have knowledge of how the other agents around them have performed. Namely, they know which choices their neighbors have found are most positive so far and how positive the best pattern of choices was. In the PSO system, each individual makes his decision according to his own experiences (Cognitive) and other agent's experiences (social). The system initially has a population of random solutions. Each potential solution, called a particle (agent, indivodual), is given a random velocity and is flown through the problem space. The agents have memory and each agent keeps track of its previous best position (called the Pbest) and its corresponding fitness. There exist a number of Pbest for the respective agents in the swarm and the agent with greatest fitness is called the global best (Gbest) of the swarm. Each particle is treated as a point in a n-dimensional space. The i-th particle is represented as Xi = (xi1, xi2, . . . , xin). The best previous position of the i-th particle (Pbesti) that gives the best fitness value is represented as Pi = (pi1,pi2, . . . , pin). The best particle among all the particles in the population is represented by Pg = (pg1,pg2, . . . , pgn). The velocity, i.e., the rate of the position change for particle i is represented as Vi = (vi1, vi2, . . . , vin).

The particles are manipulated according to the following equations (the superscripts denote the iteration):

$$v_i^{k+1}=w\times v_i^k+c_1\times r_1\times\left(p_i-x_i^k\right)+c_2\times r_2\times\left(p_g-x_i^k\right) \quad (3)$$

$$x_i^{k+1}=x_i^k+v_i^{k+1}(2) \quad (4)$$

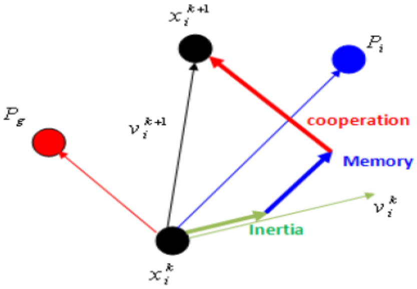

Fig. 1: Description of velocity and position updates in particle swarm optimization for a two dimensional parameter space.

Where $i = 1, 2, ..., N$, and N is the size of the population; w is the inertia weight; c1 and c2 are two positive constants, called the cognitive and social parameter respectively; r1 and r2 are random numbers uniformly distributed with in the range [0,1]. Equation (3) is used to determine the i-th particle's new velocity v_i^{k+1}, at each iteration, while equation (4) provides the new position of the i-th particle x_i^{k+1}, adding its new velocity v_i^{k+1}, to its current position x_i^k . Figure 1 shows the Description of velocity and position updates of a particle for a two-dimensional parameter space, also the pseudo code of the general PSO algorithm is shown in figure 2 .

```
Randomly initialize positions and velocities of all particles .
While termination criteria has not satisfied Do{
    Set Pbest and Gbest.
    Calculate particle velocity according to equation (3).
     Update particle position according to equation (4).
     Evaluate the objective function value (fitness value).
}
satisfactory solution has been found.
```

Fig. 2. The pseudo code of the general PSO algorithm .

4. Parameters Adaptation

The swarm population size is often between 10 to 40. The reason for a lower population size is that it significantly lowers the computing time. This is because during initialization, all the particles must be in the feasible space. Randomly initialized particles are not always in the feasible space. So initialization may take a longer time if the population is too large. However, for complex cases, a larger population size is preferred. In PSO, there are not many parameters that need to be tuned. Only the following several parameter need to be care of: maximum velocity $V_{\max}$, inertia weight w, acceleration coefficient C_1 and C_2 . Previously, Shi and Eberhart [27] introduced constant inertia weight and linear inertia weight varying usually between 0.8 and 0.4 where it in the first iteration 0.8 and decreasing during the process of run to be 0.4 in the last iteration. On the other hand, Kennedy [28] asserted that the sum of the cognitive and social values c1 and c2 should approximately equal 4.0. For constriction, Carlisle and Dozier [29] have shown that it is advantageous to adjust the cognitive/social ratio to favor cognitive learning (an individualistic swarm). They report that values of 2.8 and 1.3 respectively for the cognitive and social components yield the best performance for the test set they consider.

4.1. Inertia Weight Parameter

In this paper we present an improved PSO algorithm, which uses the dynamic inertia weight that changes according to iterative generation account. We introduce a new modified inertia weight parameter such that:

$$w = 0.5\left(1 - \sin\left(\frac{\pi t}{4}\right)\right), \quad \text{t=1,2,....N}$$

Where t is the generation number, the modified inertia parameter and be visualized in figure 3.

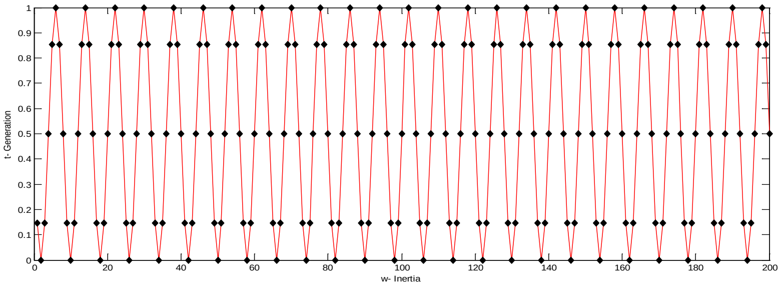

Fig.3. Modified inertia weight parameter

4.2. Cognitive and Social Parameter

In this section, we present a new procedure to generate cognitive and social parameter. The idea of this technique is as follows

$$C_1 = 1.3\gamma + 1.8(1-\gamma) \quad ,$$

$$C_2 = 1.3(1-\gamma) + 1.8\gamma$$

Where $\gamma = 1.4\delta - 0.2$ and $\delta \in [0,1]$ is a random generated number. Figure 4 gives schematic view of possible sampling region.

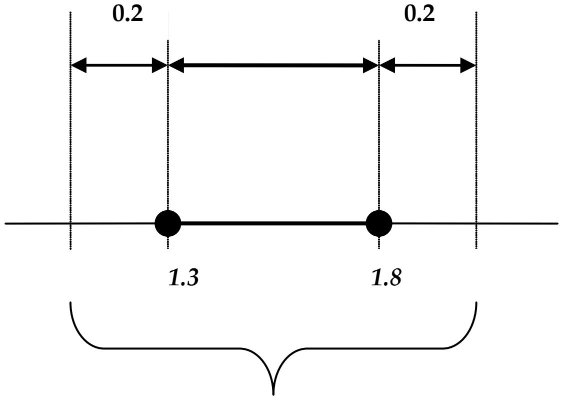

Fig. 4: Possible sampling region for Cognitive and social parameters

This procedure produces a chaos pattern as shown in figure 5. The generating pattern for 50 iteration is shown in figure 5.

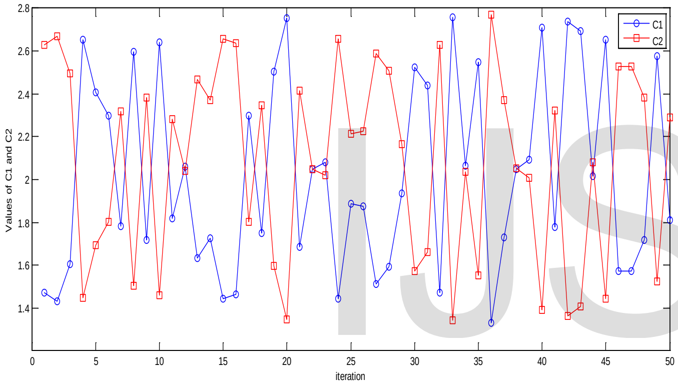

Fig. 5 : generating pattern of cognitive and social parameter for 50 iteration

5. The Proposed Optimization System

In this section, we describe a proposed approach for solving multiobjective optimization problems. A population of particles was generated randomly independent of each other and distributed uniformly, which navigates through the search space. PSO is enriched through chaotic constriction factor, this enrichment, accelerate the convergence property of the proposed optimization tool and retain the feasibility of the particles, where it controls the movement velocity of each particle so as to improve search engine visibility. Then chaotic local search is employed as a neighbourhood search engine to explore the less crowded area on the Pareto front. The description diagram of the proposed algorithm is shown in figure 6 and it is described as follows:

Phase I: PSO

Step 1. Initialization: A population of particles with random positions and velocities on n-dimensions is initialized in the problem space.

Step 2. Evaluation: The desired optimization fitness function $\left[f_i\left(\bar{x}\right),\ i=1,2,\dots,M \right]$ in n variables is evaluated for each particle.

Step 3. Setting Pbest and Gbest: Set Pbest of each particle equal to its current position, and set Gbest equal to the position of best initial particle.

Step 4. Updating the velocity and position:

- Update the velocity of each particle according to equation (3).

  To enrich the searching behavior and to avoid being trapped into infeasible region, chaotic dynamics (Chaotic constriction factor χ) is incorporated into the PSO:

- Update the position of each particle according to

$$x_i^{k+1} = x_i^k + \chi\, v_i^{k+1}$$

To restrict velocity and control it during Evolution of particles, some authors [30-31] use a constant/dynamic constriction factor χ, which has a constant value to improve the performance of PSO. A well-known logistic equation is employed, where it exhibits chaotic dynamics.

$$\chi_{n+1} = \mu \cdot \chi_n (1 - x_n),\ \chi_0 = 10^{-6}, \mu = 4,\ n = 0,1,2..$$

Where, n is the age of the infeasible particle (How long it's still unfeasible?)
the new position x_i^{k+1} depends on velocity v_i^{k+1}.

$$x_i^{k+1} = x_i^k + v_i^{k+1}$$

Then, v_i^{k+1} makes the particle to lose its feasibility, so we introduce a new modified factor χ such that new modified position of the particle is computed as::

$$x_i^{k+1} = x_i^k + \chi\, v_i^{k+1}$$

Interested readers could refer to Liu [31] for more details. Pseudo code of the proposed chaotic constriction factor is shown in figure 6.

```
Procedure make
    ( $POPULATION = \{ p_i = \{x_i, v_i\} : (i = 1,2,\dots pop_{size}) \}$ )
  Begin
  $i \leftarrow 1$
    While $(i < \text{pop-size})$ do
      $\chi_0 = 10^{-6}$
      While $p_i = \{x_i, v_i\}$ unfeasible
        $x_i^{k+1} = x_i^k + \chi\, v_i^{k+1}$
          Check feasibility
          $\chi_{n+1} = \mu \cdot \chi_n (1 - x_n),$
        End
      End
```

Fig 6.Pseudo code of Chaotic constriction factor

Step 5. Evaluation: Evaluate the desired optimization fitness function in n variables for each particle.

Step 6. Updating Pbest and Gbest : For each particle, compare its current objective value with its Pbest value. If the current value is better, then update Pbest with the current position and objective value. Determine the best particle of the entire current population with the best objective value. If the objective value is better than that of Gbest, then update Gbest with the current best particle.

Step 8. Ranking: Ranks individuals (particles) according to their objective value, and returns a column vector containing the corresponding individual fitness value.

Step 9. Archive $A^{(t)}$ Update : In order to ensure convergence to the true Pareto solutions, we concentrated on how elitism could be presented in the algorithm. So, we propose an archiving-selection strategy (figure 7) that guarantees at the same time progress towards the Pareto-optimal set and a covering of the whole range of the non-dominated solutions. This can be done using update function where, it gets the new population $P^{(t)}$ and the old archive set $A^{(t-1)}$ and determines the updated one, namely $A^{(t)}$.

```
Input: Current archive $A$, new solution $x$
  If $\{\exists\, x' \in A \mid x' \succ x\}$ Do
    $A' \leftarrow A$
      Else
      $D = \{x' \in A \mid x \succ x'\}$
      $A' = A \cup \{x\} \setminus D$
    End
Output: $A'$
```

Fig.7: Pseudo code of the Archive updating

Phase 2: Chaotic Local search

In this section, Chaotic Local search is described, depending on chaotic equation a new chaotic local search has been driven as follows:

A well-known logistic equation Id employed for generating neighborhood solution.

$$\chi_{n+1} = \mu \cdot \chi_n (1 - \chi_n),\ \chi_0 = 10^{-6}, \mu = 4,\ n = 0,1,2....;$$

Although the above equation is deterministic, it exhibits chaotic dynamics. Interested readers could refer to Liu [31] for more details.

The general procedure can be described by the following steps:

Step 1: Start with each population point $\overline{x}_i == \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, called the starting point, and the prescribed step lengths Δx_i in each of the coordinate directions $\mathrm{u}_i, i = 1, 2, ..., n.$ Set k = 1.

Step 2: Compute $f_k = f\ (\mathrm{X}_k)$. Set i = 1, $\mathrm{Y}_{k0} = \mathrm{X}_k$, and start the exploratory move as stated in step 3.

Step 3: The variable x_i is perturbed about the current temporary starting point $\mathrm{Y}_{k,i-1}$ to obtain the new temporary base point as

$$Y_{k,i} = Y_{k,i-1} + \chi_i \Delta x_i,$$

$$\chi_i = \mu \cdot \chi_{i-1}(1 - \chi_i),\ \chi_{i-1} = 10^{-6}, \mu = 4,\ n = 0,1,2....$$

This process of finding the new temporary point is continued for $i = 1, 2, ...\text{until}\ n$ is perturbed to find $Y_{k,n}$.

The algorithm maintains a finite-sized archive of non-dominated solutions which gets iteratively updated in the presence of new solutions based on the concept of dominance, such that new solutions are only accepted in the archive if they are not dominated by any other element in the current archive

```
Initialization: $POPULATION = \{ p_i = \{x_i^t, v_i^t\} : (i = 1, 2, ...pop_{size}) \}$
While (navigation not completed) do

Phase 1:
Evaluation
Set $P_{best}, G_{best}$
Update $v_i^{t+1} = \psi(x_i^t, v_i^t, P_{best}, G_{best}),\ x_i^{t+1} = \xi(x_i^t, v_i^{t+1})$ for all particles
Update archive
Phase 2:
For i=1 : size of archive Do {
Start with each archive point $\overline{x}_i == \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$,
    For j=1 : n Do {
    $F = [f_1(\overline{x}_i), f_2(\overline{x}_i), .... f_M(\overline{x}_i)]$, $Y_{i0} = \overline{x}_i$
    $\overline{x}_i$ is perturbed $Y_{k,i} = Y_{k,i} + \chi_i \Delta x_i$        End
          End
```

Fig. 8. The pseudo code of the proposed algorithm

6. Computational Experiment

The performance of the proposed algorithm for global optimization continuous function is tested on several constrained well-known engineering benchmark problems [32-38]. The algorithm is coded in MATLAB 8.0 and the simulations are run on CPU with a 2.7GHz Core 2 Duo Intel Processor. Table 1 list the parameter setting used in this simulation

Population size	100
Cognitive parameter	Dynamic Cognitive parameter
Social parameter	Dynamic Social parameter
Maximum velocity of particles	Vmax=Xmax-Xmin
Inertia weight	Dynamic inertia weght
Intial constriction factor χ_0	10e-6

Table 1. The algorithm parameters

6.1. Applications

The proposed approach is applied to some engineering problem was chosen from the engineering application.

- A welded beam design

A welded beam design is used by Deb [33], where a beam needs to be welded on another beam and must carry a certain load F (Fig. 9).

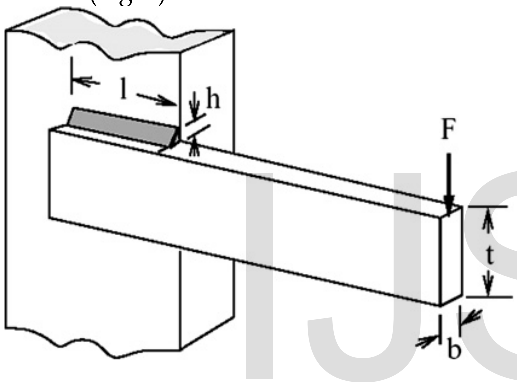

Fig. 9: The welded beam design problem

It is desired to find four design parameters (thickness b, width t, length of weld l, and weld thickness h) for which the cost function of the beam and the deflection function at the open end are minimum. The overhang portion of the beam has a length of “14 inch” and “F=6000 Ib “ force is applied at the end of the beam. A little thought will reveal that a design for minimum deflection at the end (or maximum rigidity of the above beam) will make all four design dimensions to take large dimensions. Thus, the design solutions for minimum cost and maximum rigidity (or minimum-end-deflection) are conflicting to each other. In the following, the mathematical formulation of the two-objective optimization problem is presented as follows:

$$Min\ f_1(x) = 1.10471h^2l + 0.04811tb(14+l)$$

$$Min\ f_2(x) = 2.1952/t^2b$$

$$s.t.$$

$$g_1(x) = 13600 - r(x) \geq 0, \qquad g_2(x) = 30000 - \sigma(x) \geq 0$$

$$g_3(x) = b - h \geq 0, \qquad g_4(x) = P_c(x) - 6000 \geq 0$$

$$h, b \in [0.125, 5] \quad l, t \in [0.1, 10]$$

where

$$\tau = \sqrt{(\tau')^2 + (\tau'')^2 + l\tau'\tau'' / \sqrt{0.25(l^2 + (h+l)^2)}} \quad \tau' = 6000/\sqrt{2hl}$$

$$\tau'' = \frac{6000(14+0.5l)\sqrt{0.25(l^2+(h+l)^2)}}{2\sqrt{2hl}\,(l^2/12 + 0.25(h+t)^2)} \quad \sigma = 504000/t^2b$$

$$P_c = 64746.022(1 - .0282346t)tb^3$$

In the Welded Beam design problem, the non-linear constraints can cause difficulties in finding the Pareto front. As shown in Fig. 10.

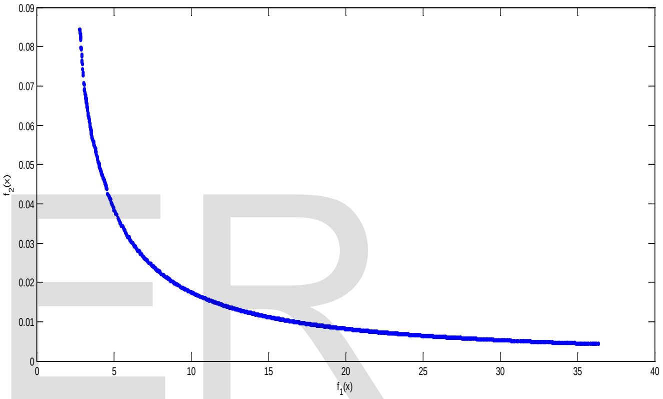

Fig. 10: Pareto optimal front of welded beam using our approach

- Two-Bar Truss

Figure 11 illustrates the two-bar truss that is to be optimized [34]. This problem was adapted from Kirsch [35]. It is comprised of two stationary pinned joints, A and B, where each one is connected to one of the two bars in the truss. The two bars are pinned where the join one another at joint C, and a 100 kN force acts directly downward at that point. The cross-sectional areas of the two bars are represented as x_1 and x_2, the cross-sectional areas of trusses AC and BC respectively. Finally, y represents the perpendicular distance from the line AB that contains the two-pinned base joints to the connection of the bars where the force acts (joint C). The two-bar truss is shown below.

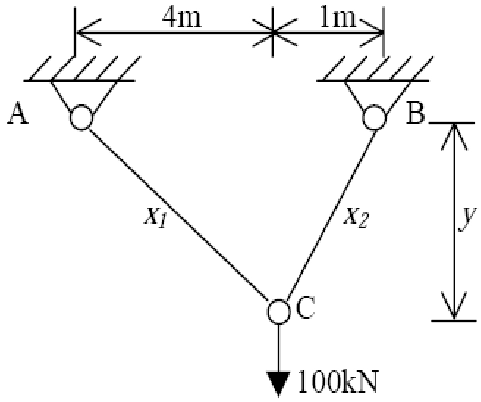

Fig.11: Two-Bar Truss

The problem has been modified into a two-objective problem in order to show the non-inferior Pareto set clearly in two dimensions. The stresses in AC and BC should not exceed "100,000 kPa" and the total volume of material should not exceed 0.1 m3. The reason the objective constraints have been imposed is that the Pareto set is asymptotic and extends from -∞ to ∞. As x_1 and x_2 go to zero, fvolume goes to zero and fstress,AC and fstress,BC go to infinity. As x_1 and x_2 go to infinity, fvolume goes to infinity and fstress,AC and fstress,BC go to zero. Hence, in order to generate Pareto optimal solutions in a reasonable range, objective constraints are imposed. The problem formulation is shown below.

$$\text{Minimize } f_{volume} = x_1(16+y^2)^{0.5} + x_2(1+y^2)^{0.5}$$

$$\text{Minimize } f_{stress,AC} = \frac{20(16+y^2)^{0.5}}{yx_1}$$

$$\text{subject to}$$

$$f_{volume} \le 0.1$$

$$f_{stress,AC} \le 100000$$

$$f_{stress,BC} = \frac{80(1+y^2)^{0.5}}{yx_2} \le 100000$$

$$1 \le y \le 3$$

$$x_1, x_2 > 0$$

Figure 12 declares the Pareto optimal solution of the Two-Bar Truss. Obviously from the results, the proposed algorithm is able to maintain an almost uniform set of non-dominated solution points along the true Pareto-optimal front.

- Speed Reducer Design

The well-known Speed Reducer test Problem represents the design of a simple gear box such as might be used in a light airplane between the engine and propeller to allow each to rotate at its most efficient speed (Fig.13).

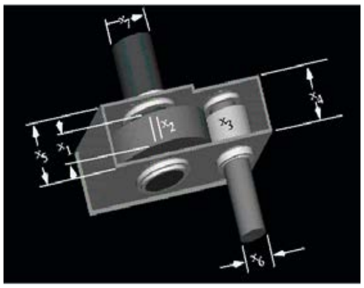

Fig. 13. A speed reducer

The objective is to minimize the speed reducer weight while satisfying a number of constraints imposed by gear and shaft design practices. This problem was modeled by Golinski[36] as a single-level optimization, and since then many others have used it to test a variety of methods. Here, the problem has been converted into a two objective optimization problem. The mathematical formulation, of the problem is now described. There are seven design variables, $(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$, which represent as depicted in table 2.

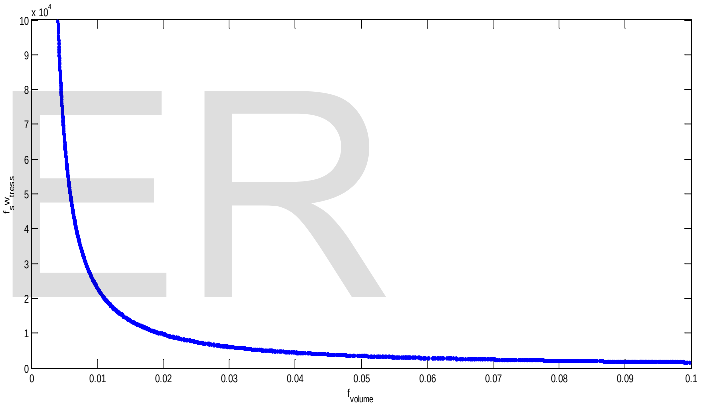

Fig. 12: Pareto optimal front of two- truss using our approach

x_1	width of the gear face, cm
x_2	teeth module, cm
x_3	number of pinion teeth (Integer)
x_4	shaft 1 length between bearings, cm
x_5	shaft 2 length between bearings, cm
x_6	diameter of shaft 1, cm
x_7	diameter of shaft 2, cm

Table 2. Design variables

The first objective $f_1(\cdot)$ is to find the minimum of a gear box volume, (and, hence, its minimum weight). The second objective, $f_2(\cdot)$, is to minimize the stress in one f the two gear shafts. The design is subject to constraints imposed by gear and shaft design practices. An upper and lower limit is imposed on each of the seven design variables. There are 11 other inequality constraints as depicted in table 3.

g_1	Upper bound on the bending stress of the gear tooth
g_2	Upper bound on the contact stress of the gear tooth
g_3, g_4	Upper bounds on the transverse deflection of shafts 1, 2
$g_5 - g_7$	dimensional restrictions based on space and experience
g_8, g_9	design requirements on the shafts based on experience
g_{10}, g_{11}	Constraints on stress in the gear shafts

Table 3: The problem constraints

The optimization formulation is

$$Min\ \mathrm{f}_1 = \mathrm{f}_{\mathrm{volume}} =$$

$$0.7854\mathrm{x}_1 x_2^2 (10 x_3^2 / 3 + 14.933 x_3 - 43.0934) - 1.508 x_1 (x_6^2 + x_7^2)$$

$$+ 7.477 (x_6^3 + x_7^3) + 0.7854 (x_4 x_6^2 + x_5 x_7^2),$$

$$Min\ \mathrm{f}_2 = f_{stress} = \frac{\sqrt{(745 x_4 / x_2 x_3)^2 + 1.69 \times 10^7}}{0.1 x_6^3}$$

As shown in Figure 14, our proposed approach works well in both distribution and spread. Also, it keep track of all the feasible solutions found by iteratively update the archive content during the optimization.

$$\mathrm{g}_1 : \frac{1}{x_1 x_2^2 x_3} - \frac{1}{27} \le 0, \qquad \mathrm{g}_2 : \frac{1}{x_1 x_2^2 x_3^2} - \frac{1}{397.5} \le 0,$$

$$\mathrm{g}_3 : \frac{x_4^3}{x_2 x_3 x_6^4} - \frac{1}{1.93} \le 0, \qquad \mathrm{g}_4 : \frac{x_5^3}{x_2 x_3 x_7^4} - \frac{1}{1.93} \le 0$$

$$\mathrm{g}_5 : x_2 x_3 - 40 \le 0, \qquad \mathrm{g}_6 : \frac{x_1}{x_2} - 12 \le 0,$$

$$\mathrm{g}_7 : 5 - \frac{x_1}{x_2} \le 0, \qquad \mathrm{g}_8 : 1.9 - x_4 + 1.5 x_6 \le 0,$$

$$\mathrm{g}_9 : 1.9 - x_5 + 1.1 x_7 \le 0, \qquad \mathrm{g}_{10} : f_2(\overline{x}) \le 1100,$$

$$\mathrm{g}_{11} : \frac{\sqrt{(745 x_5 / x_2 x_3)^2 + 1.575 \times 10^8}}{0.1 x_7^3} \le 850.$$

$$2.6 \le x_1 \le 3.6, \quad 0.7 \le \mathrm{x}_2 \le 0.8, \quad 17 \le x_3 \le 18, \quad 7.3 \le \mathrm{x}_4 \le 8.3,$$

$$7.3 \le x_5 \le 8.3, \quad 2.9 \le \mathrm{x}_6 \le 3.9, \quad 5.0 \le x_7 \le 5.5,$$

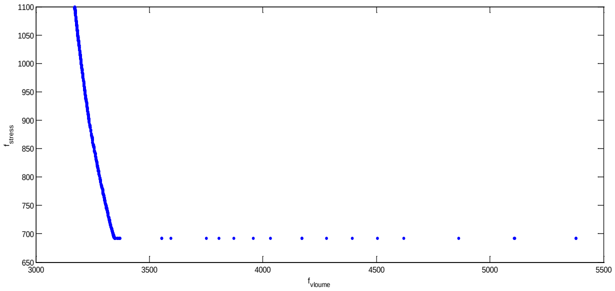

Fig. 14. Result for the speed reducer design

6.2. Performance Assessment

Our proposed algorithm was tested on set of engineering applications in engineering. The test suite is a collection of different characteristics of Pareto front. Each of the test problems was run twenty times independently, with different seeds, ID metric is used to measure the performance of the algorithm.

Performance Metric (ID): Let P^* be an ideal solution (in the objective space). Let A be an approximate set to the PF, the average distance from P* to A is defined as:

$$\mathrm{ID}\left(A, P^*\right) = \frac{\sum d\left(P^*, A\right)}{|A|}$$

Where $d\left(P^*, A\right)$) is the minimum Euclidean distance between P^* and the points in A.

Table 4 shows the ideal solution for each problem (Calculated from the Pareto front for each application).

Problem	Ideal solution
welded beam design	(2.78001,0.004407)
Two-Bar Truss	(0.0040169,1.72146e003)
Speed Reducer Design	(3.17017e003,0.6930265e003)

Table 4. Ideal solution

Table 5 shows the comparison between the proposed approach and five of the most recent evolutionary algorithms [39-42] using ID performance metric. In this paper, multiobjective PSO is enhanced with chaotic local search scheme. The performance of our approach was evaluated on four test benchmark functions from engineering domain. In Tables 5, the best and worst obtained IGD values for all the test problems are presented with their mean and standard deviation. Also, for each problem, we can rank the different methods according to the ID values and get table 6 and table 7. It is obvious that the algorithm performances well on most of the test problems. As is evident from table 5, table 6, table 7, in all problems, global convergence is obtained and the complete Pareto optimal frontier is discovered. The primary cause of this behavior is local search strategy which enables the algorithm to search less crowded area in the search space. Hence an effective integration of chaotic local search and PSO algorithm is the reason for a better performance of the hybrid algorithm. Overall, the proposed hybrid algorithm performs well on the test problems used for this study. The inclusion of chaotic local search speeds-up the search process and also helps in obtaining a fine-grained value for the objective functions.

Problem	The proposed algorithm			hybrid particle swarm [41]	Ant Optimization System [42]	quantum genetic algorithm [43]	Rough Sets Based Approach [44]
	Mean(ID)	Smallest (ID)	Largest(ID)	Mean(ID)	Mean(ID)	Mean(ID)	Mean(ID)
welded beam design	13.7812	13.6325	14.1023	13.8823	13.9321	13.8102	13.7923
Two-Bar Truss	1.9102e4	1.8233e4	1.9821e4	1.9786e4	1.9524e4	1.9203e4	1.9352e4
Speed Reducer Design	2.7156e2	2.5123e2	2.8130e2	2.7801e2	2.7351e2	2.7542e2	2.7831e2

Table 5. The mean, standard deviation, the smallest and the largest values of the IGD used for each test problem.

Problem	The proposed algorithm	hybrid particle swarm [41]	Ant Optimization System [42]	quantum genetic algorithm [43]	Rough Sets Based Approach [44]
welded beam design	Rank 1	Rank 4	Rank 5	Rank 3	Rank 2
Two-Bar Truss	Rank 1	Rank 5	Rank 4	Rank 2	Rank 3
Speed Reducer Design	Rank 1	Rank 4	Rank 2	Rank 3	Rank 5

Table 6. Ranking of the IGD values.

Method	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
The proposed algorithm	3	0	0	0	0
hybrid particle swarm [41]	0	0	0	2	1
Ant Optimization System [43]	0	1	0	1	1
quantum genetic algorithm [42]	0	1	2	0	0
Rough Sets Based Approach [44]	0	1	1	0	1

Table 7. Statistics of Ranking.

6. Conclusion

In this paper, a hybrid multiobjective algorithm combining adaptive PSO and chaotic local search is proposed. To enrich the searching behaviour and to avoid being trapped into local optimum, dynamic adaptive of PSO parameters are incorporated into PSO. In the proposed algorithm, inertia weight of each particle is dynamically updated, also personal influence and social influence parameters are dynamically adapted during the process. Moreover, a modified velocity updating formula of the particle is presented, where, a new constriction factor which control the feasibility of the particles is presented. Chaotic search CS is employed for the local search to explore the less crowded area in the Pareto front. The results, provided by the proposed algorithm for benchmark engineering problems, are promising when compared with exiting well-known algorithms.

References

[1] S. Jaganathan, S. Palaniswami, G. Maharaja Vignesh, R. Mithunraj, Applications of multi objective optimization to reactive power planning problem using ant colony algorithm, Eur. J. Sci. Res. 51 (2011) 241–253.

[2] F. Petterson, N. Chakraborti, S.B. Singh, Neural networks analysis of steel plate processing augmented by multi-objective genetic algorithms, Steel. Res. Int. 78 (2007) 890–898.

[3] R.E. Steuer, Y. Qi, M. Hirschberger, Portfolio selection in the presence of multiple criteria, in: C. Zopounidis, M. Doumpos, P.M. Pardalos (Eds.), Handbook of Financial Engineering, in: Springer Optimization and Its Applications, vol. 18, Springer, US, 2008, pp. 3–24.

[4] K.L. Kiran, S. Lakshminarayanan, Treatment planning of cancer dendritic cell therapy using multi-objective optimization, in: Proceedings of ADCHEM 2009, Istanbul, Turkey.

[5] A. Petrovski, J. McCall, B. Sudha, Multi-objective optimization of cancer chemotherapy using swarm intelligence, in: AISB 2009 Symposium on Adaptive and Emergent Behaviour and Complex Systems.

[6] R. Janssen, M. van Herwijnen, T.J. Stewart, J.C.J.H. Aerts, Multiobjective decision support for land-use planning, Environ. Plann. B 35 (2008) 740–756.

[7] Miettinen K. "Non-linear multiobjective optimization" Dordrecht: Kluwer Academic Publisher (2002).

[8] A.A.Mousa, 'Using genetic algorithm and TOPSIS technique for multiobjective transportation problem :ahybrid approach', International Journal of Computer Mathematics 87(13)(2010)3017–3029.

[9] Waiel. F. Abd El-Wahed, A.A.Mousa , M. A. El-Shorbagy, Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems, Journal of Computational and Ap-

plied Mathematics 235 (2011) 1446–1453.(Top 25 Hottest Articles, Oct. to Dec. 2010, Jan. to Mar. 2011, April to June 2011)

[10] A.A.Mousa, M.A.El-Shorbagy, WaielF.AbdEl-Wahed, Local search based hybrid particle swarm optimization for multiobjectuve optimization ,International Journal of Swarmand Evolutionary Computation3(2012)1–14.

[11] A.A.Mousa,I.M.El_Desoky,Stability of Pareto optimal allocation of land reclamation by multistage decision- based multipheromone ant colony optimization, Swarm and Evolutionary Computation(2013), http://dx.doi.org/10.1016/j.swevo.2013.06.003i

[12] Renata E. Onety, Roberto Tadei, Oriane M. Neto, Ricardo H.C. Takahashi , Multiobjective optimization of MPLS-IP networks with a variable neighborhood genetic algorithm , Applied Soft Computing, Volume 13, Issue 11, November 2013, Pages 4403-4412

[13] James N. Richardson, Guy Nordenson, Rebecca Laberenne, Rajan Filomeno Coelho, Sigrid Adriaenssens , Flexible optimum design of a bracing system for façade design using multiobjective Genetic Algorithms , Automation in Construction, Volume 32, July 2013, Pages 80-87

[14] H. Ahonen, A.G. de Alvarenga, A.R.S. Amaral , Simulated annealing and tabu search approaches for the Corridor Allocation Problem ,European Journal of Operational Research, Volume 232, Issue 1, 1 January 2014, Pages 221-233

[15] Chen-Fu Chen, Muh-Cherng Wu, Keng-Han Lin , Effect of solution representations on Tabu search in scheduling applications ,Computers & Operations Research, Volume 40, Issue 12, December 2013, Pages 2817-2825.

[16] Celal Özkale, Alpaslan Fığlalı , Evaluation of the multiobjective ant colony algorithm performances on biobjective quadratic assignment problems ,
Applied Mathematical Modelling, Volume 37, Issues 14–15, 1 August 2013, Pages 7822-7838

[17] Kun Fan, Weijia You, Yuanyuan Li , An effective modified binary particle swarm optimization (mBPSO) algorithm for multi-objective resource allocation problem (MORAP) ,Applied Mathematics and Computation, Volume 221, 15 September 2013, Pages 257-267

[18] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the 1995 IEEE International Conference on Neural Networks, vol. 4, 1995, pp.1942–1948.

[19] X. Jie, X. Deyun, New metropolis coefficients of particle swarm optimization,in:IEEE, 2008.

[20] J. J. Liang, and P. N. Suganthan, "Dynamic Multi-Swarm Particle Swarm Optimizer," Proc. of IEEE Int. Swarm Intelligence Symposium, pp. 124-129, 2005.

[21] Blackwell, T.M., Swarm in dynamic environments, proc genetic and evolutionary computation conference(2003)1:12.

[22] G. Zhang, J. Jiang, Z. Su, M. Qi, H. Fang, “Searching for overlapping coalitions in multiple virtual organizations,” Information Sciences 180 (17), 2010,pp.3140–3156.

[23] Y.T. Juang et al., “Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions,” Inform. Sci, 2010, DOI 10.1016/j.ins.2010.11.025.

[24] J. Kennedy and R. Eberhart. Particle Swarm Optimization, Proc. of IEEE International Conference on Neural Networks, vol. IV, Perth, Australia, (1995) 1942–1948.

[25] J. kennedy, R.C. Ebrhart, A new optimizer using particle swarm theory, in: Proc. of the sixth Int. Symp. on Micro Machine and Human science (MHS'95), Nagoya, Japan (1995) 39-43.

[26] J. Kennedy and R.C. Eberhart. Swarm Intelligence, Morgan Kaufmann, San Francisco, (2001).

[27] Shi Y., and Eberhart R. C., "A modified particle swarm optimizer”, In Proceedings of the IEEE International Conference on Evolutionary computation, pp.69–73. IEEE Press, Piscataway, USA, 1998.

[28] Kennedy J., Eberhart R. C., and Shi Y., Swarm Intelligence, Morgan Kaufmann, 2001.

[29] Carlisle A. and Dozier G., ”Adapting particle swarm optimization to dynamic environments”, In International Conference on Artificial Intelligence, Vol.I, pp.429–434, Las Vegas, 2000.

[30] Syamasree Biswas, Kamal Krishna Mandal, Niladri Chakraborty ,Constriction factor based particle swarm optimization for analyzing tuned reactive power dispatch, Frontiers in Energy June 2013, Volume 7, Issue 2, pp 174-181.

[31]] B. Liu, L. Wang ands Y.-H. Jin, “Improved particle swarm optimization combined with chaos,” Chaos Solutions & Fractals, vol. 25, no. 5, 2005, pp. 1261-1271.

[32] Osman M.S., M.A.Abo-Sinna, and A.A. Mousa " IT-CEMOP: An Iterative Co-evolutionary Algorithm for Multiobjective Optimization Problem with Nonlinear Constraints" Journal of Applied Mathematics & Computation (AMC) 183, pp373-389, (2006)

[33] . K. Deb, A. Pratap, and S. Moitra, (2000) " Mechanical Component Design for Multiple Objectives Using Elitist Non-Dominated Sorting GA", Kangal Report No. 200002.

[34] Ruhul Sarker, Hussein A. Abbass, and Samin Karim , An Evolutionary Algorithm for Constrained Multiobjective Optimization Problems , at the 5th Australasia-Japan JointWorkshop University of Otago, Dunedin, New Zealand November 19 th -21st 2001.

[35] Kirsch, U., 1981, Optimal Structural Design, McGraw-Hill Co., New York.

[36] J. Golinski, Optimal Synthesis Problems Solved by Means of Nonlinear Programming and Random Methods, Journal of Mechanisms, Vol. 5, pp. 287-309, (1970).

[37] Messac, A., 1996, “Physical Programming: Effective Optimization for Computational Design,” AIAA Journal 34, (1), 149-158.

[38] Shapour Azar * Brian J. Reynolds Sanjay Narayanan, comparison of two multiobjective optimization techniques with and withingenetic algorithms, Proceedings of the 1999 ASME Design Engineering Technical Conferences September 12-15, 1999, Las Vegas, Nevada

[39] A.A.Mousa , M. A. El-Shorbagy , Waiel. F. Abd El-Wahed, Local search based hybrid particle swarm optimization for multiobjective optimization, International journal of Swarm and evolutionary computation, 3(2012),1:14.

[40] A. A. Mousa, Hybrid Ant Optimization System for Multiobjective Economic Emission Load Dispatch Problem Under Fuzziness, Swarm and Evolutionary Computation, Volume 18, October 2014, Pages 11–21.

[41] A. A. Mousa and E. E. Elattar, “Best compromise alternative to EELD problem using hybrid multiobjective quantum genetic algorithm,” International Journal of Applied Mathematics & Information Sciences, 8, No. 6, 2889-2902 (2014).

[42] Mohamed A. Hussein, Ahmed A. EL-Sawy, EL-Sayed M. Zaki, A. A. Mousa, Genetic Algorithm and Rough Sets Based Hybrid Approach for Economic Environmental Dispatch of Power Systems, British Journal of Mathematics & Computer Science, Vol.: 4, Issue.: 20 (16-31 October).