A Modified Multi-Objective Particle Swarm Technique with Chaos for Structural Optimization

Mohamed F. El-Santawy* and Ramadan A. Zein Eldin

Department of Operation Research, Institute of Statistical Studies and Research (ISSR)
Cairo University, Egypt
*Corresponding author: lost_zola@yahoo.com

Abstract: The main goal of this paper is to assess the incorporation of Chaos search to Multi-Objective Particle swarm optimization. The proposed algorithm combined chaotic maps to produce random numbers needed by the algorithm during search. The new technique so-called Multi-Objective Chaotic Particle Swarm Optimization (MOCPSO) uses an external archive for keeping the solutions found over iterations. Fitness Sharing method is employed to maintain diversity of solutions found in the external archive. For validity, the proposed technique is applied to a well-known structural optimization problem called two-bar truss problem, and the results show the efficiency of adding chaos.

[Mohamed F. El-Santawy and Ramadan A. Zein Eldin. **A Modified Multi-Objective Particle Swarm Technique with Chaos for Structural Optimization.** *Life Sci J* 2013;10(4):3420-3422] (ISSN:1097-8135). http://www.lifesciencesite.com. 511

Keywords: Chaos search; External Archive; Fitness Sharing; Multi-Objective Optimization; Particle Swarm Optimization.

1. Introduction

Decision situations often involve multiple criteria or objectives. In many cases, objectives are *incommensurable*, meaning they are not comparable with respect to magnitude and value, and *conflicting*, meaning that the different objectives cannot be arbitrarily improved without decreasing the value of another. This result in trade-offs between the objectives [3]. Multi-Objective programming is a part of mathematical programming dealing with decision problems characterized by multiple and conflicting objective functions that are to be optimized over a feasible set of decisions. Such problems, referred to as Multi-Objective Programs (MOPs), are commonly encountered in many areas of human activity including engineering, management, and others [4].

Particle swarm optimization (PSO) is a population-based stochastic optimization technique modeled on the social behaviors observed in animals or insects, e.g., bird flocking, fish schooling, and animal herding. It was originally proposed by James Kennedy and Russell Eberhart in 1995 [7]. Since its inception, PSO has gained increasing popularity among researchers and practitioners as a robust and efficient technique for solving difficult optimization problems [3].

A structure in mechanics is defined by J.E. Gordon [6] as "any assemblage of materials which is intended to sustain loads." Chaos is a kind of common nonlinear phenomenon, which has diverse, complex and sophisticated native under apparent disorder.

In this paper, a new method is developed to solve the two-bar truss problem. The new method combined Particle Swarm Optimization (PSO) algorithm to chaos search in order to enhance exploration during search. The proposed method called Multi-Objective Chaotic Particle Swarm Optimization (MOCPSO) uses an external archive for keeping the *nondominated* solutions gained during search. This paper is structured as following: Section 2 is made for Multi-Objective Optimization basic concepts, section 3 is devoted to the Particle Swarm Optimization technique, the proposed technique is illustrated in section 4, the two-bar problem is presented in section 5, and finally in section 6 conclusion is introduced.

2. Multi-Objective Optimization

A general multi-objective optimization problem consists of a number of objectives to be optimized simultaneously. In general, a k-objective minimization problem can be written as

$$min\ \{(f_1(x),\dots, f_k(x)) : x \in X\} \qquad (1)$$

we usually assume that the set X is given implicitly in the form of constraints resulted in the feasible region in the decision space [3], i.e., $X := \{x \in R^n : g_j(x) \le 0, j = 1,\dots,s;\ h_j(x) = 0, j = 1,\dots,m\}$.

Definition 1 (Pareto Dominance): Without loss of generality in a minimization problem, a decision vector $x_1 \in X$ is said to *dominate* a decision vector $x_2 \in X$ *iff* the following two conditions are satisfied:

1. The decision vector x_1 is not worse than x_2 in all objectives, or $\forall\ i \in \{1,2,\dots,k\} : f_i(x_1) \le f_i(x_2)$.
2. The decision vector x_1 is strictly better than x_2 in at least one objective, or $\exists\ i \in \{1,2,\dots,k\} : f_i(x_1) < f_i(x_2)$.

If any of the above conditions is violated, then x_1 does not dominate x_2. A decision vector $x_1 \in X$ is called *Pareto-optimal* if there is no another $x_2 \in X$ that *dominates* it and in this case x_1 is called *nondominated* with respect to X

Definition 2 (Pareto Optimal Set): The Pareto Optimal Set P^* is defined by [2]:

$$P^* = \{x \in X \mid x \text{ is } \textit{pareto-optimal}\} \quad (2)$$

3. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an evolutionary computation technique for optimization. It is inspired by the social behaviour of individuals in groups in nature [7]. The behaviour of an individual is influenced by its own experience and that of its neighbours. PSO became so popular for its simplicity since its original version; it only adopts one operator for finding creating new solutions not like other evolutionary algorithms (EAs), also it has been found effective in a wide variety of applications, in PSO the members of the entire population are maintained through the search procedure so that information is socially shared among individuals to direct the search towards the best position in the search space [5].

PSO algorithm reaches optimal solutions through two mechanisms as shown in equations 3 and 4, equation 3 for updating positions (solutions), while 4 is for velocity updating for achieving better positions (solutions). The swarm contains a population of N particles, in the iteration (t) the position of particle (i) in the search space is presented by X_i (t), and V_i (t) is the velocity updating which will be added to the old position to obtain the new position of particle (i). X_{pbest} is the best position attained by the particle during the search, while X_{leader} is the best global solution among the swarm.

$$X_i(t) = X_i(t-1) + V_i(t) \quad (3)$$

$$V_i(t) = w\,V_i(t-1) + C_1 r_1 (X_{pbest} - X_i(t)) + C_2 r_2 (X_{leader} - X_i(t)) \quad (4)$$

where w is the inertia weight factor; C_1 and C_2 are acceleration constants and r_1, r_2 are random values $\in$ [0,1].

Some authors added some techniques to the general algorithm of PSO in order to deal with Multi-Objective Optimization problems [10], they proposed many approaches in this direction, in the case of multi-objective optimization problems, each particle might have a set of different leaders from which just one can be selected in order to update its position. Such set of leaders is usually stored in a different place from the swarm that is called external archive. It is important to indicate that the majority of the currently proposed MOPSO approaches redefine the concept of leader. A quality measure that indicates how good a leader is very important. Obviously, such feature can be defined in several different ways. Thus other strategies have to be found to limit the archive size while preserving its diversity and spread [3].

4. Multi-Objective Chaotic Particle Swarm Optimization

Mathematically, chaos is randomness of a simple deterministic dynamical system and chaotic system may be considered as sources of randomness [1]. A chaotic map is a discrete-time dynamical system

$$z_{k+1} = f(z_k), \quad 0 < z_k < 1, \; k = 0,1,2,\ldots \quad (5)$$

running in the chaotic state. The chaotic sequence $\{z_k : k = 0,1,2,\ldots\}$ can be used as spread-spectrum sequence and as a random number sequence. Thus, in each iteration PSO parameters can be updated by chaotic maps. Also, the two random numbers (r_1 and r_2) are produced by moving the chaotic more steps. One famous function is employed in the proposed algorithm called the Logistic map**.** In 1976, Robert May pointed out that the logistic map led to chaotic dynamics. A logistic map is a polynomial map. It is often cited as an example of how complex behaviour can arise from a very simple nonlinear dynamical equation [9]. This map is defined by

$$z_{k+1} = \mu\, z_k (1 - z_k) \quad (6)$$

Obviously, $z_k \in [0,1]$ under the conditions that the initial $z_0 \in [0,1]$, where k is the iteration number and $\mu = 4$.

In the proposed algorithm, the solutions are kept diversified in the archive by using the fitness sharing method. The main idea of fitness sharing is to distribute a population of individuals along a set of resources [8]. When an individual i is sharing resources with other individuals, its fitness f_i is degraded in proportion to the number and closeness to individuals that surround it, and in this way promoting and maintaining diversity. In general Fitness sharing for an individual i is defined as:

$$fshare_i = \frac{f_i}{\sum_{j=0}^{n} sharing_i^j} \quad (7)$$

where n is the number of individuals in the population.

5. Two-Bar Truss Structural Problem

As shown in Fig. 1, the truss has to carry a certain load without elastic failure. Thus, in addition to the objective of designing the truss for minimum volume (which is equivalent to designing for minimum cost of fabrication), there are additional objectives of minimizing stresses in each of AC and BC.

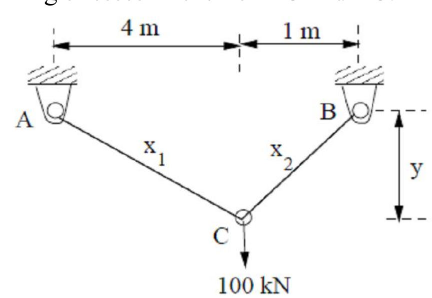

Fig. 1: The Two-Bar Truss Problem

The two-objective optimization problem for three variables y (vertical distance between B and C in m), x_1(length of AC in m) and x_2(length of BC in m) is constructed as follows:

$$\text{minimize} f_1(x) = x_1\sqrt{16+y^2} + x_2\sqrt{1+y^2}$$
$$\text{minimize} f_2(x) = \max\ (\sigma_{AC}, \sigma_{BC})$$
$$ST \quad \max\ (\sigma_{AC}, \sigma_{BC}) \leq 1(10^5) \qquad (8)$$
$$\sigma_{AC} = \frac{20\sqrt{16+y^2}}{yx_1}, \quad \sigma_{BC} = \frac{80\sqrt{1+y^2}}{yx_2}$$
$$1 \leq y \leq 3 \text{ and } x \geq 0$$

In this proposal, the parameters are set as following: N = 50; C_1 and C_2 = 2. The values of w, r_1 and r_2 are obtained by moving the logistic map one step, size of external archive is 100 solution, and we had 100 iterations. Fig. 2 shows the pareto front produced using the proposed method. An additional constraint of maximum stress being smaller than $1(10^5)$ is added to the original problem. The solutions are spread in the following range: (0.00392 m^3, 91607 kPa) and (0.05315 m^3, 8103 kPa).

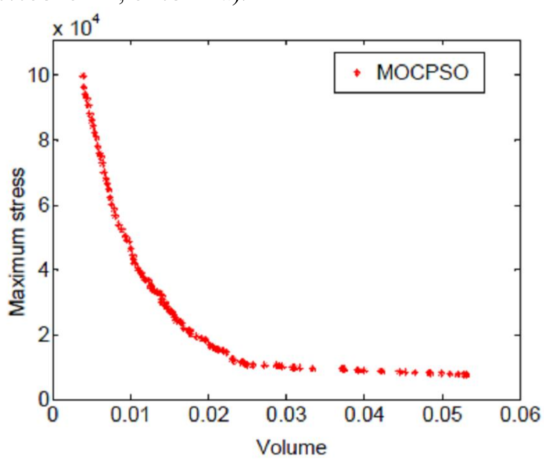

Fig. 2: Pareto Front produced using MOCPSO

6. Conclusion

In the proposed algorithm, Chaos search is combined to Multi-Objective Particle Swarm Optimizer to enhance exploration during search. The two Truss problem is solved in an efficient way. Also, an external archive is incorporated to the algorithm in order to keep solutions found and maintain diversity during search.

***Corresponding Author:**
Mohamed Fathi El-Santawy
E-mail: lost_zola@yahoo.com
12/11/2013

References

1. Alatas, B. (2010),"*Chaotic bee colony algorithms for global numerical optimization*", Expert System with Applications, vol. 37, pp. 5682–5687.
2. El-Santawy, M. F. and Ahmed, A. N. (2011), "*The chaotic ϵ-constraint approach*", Journal of American Science, vol. 7, No. 7, pp. 470-472.
3. El-Santawy, M. F. (2009), An Intelligent Search Technique for Solving Multi-Objective Optimization Problems, M.Sc. Thesis, Department of Operations Research, Institute of Statistical Studies and Research (ISSR), Cairo University, Egypt.
4. Ehrgott, M. And Ruzika, S. (2008), "*Improved ε-Constraint Method for Multiobjective Programming*", Journal of Optimization Theory and Applications, vol. 138, pp. 375–396.
5. Engelbrecht, A. P. (2005), Fundamentals of Computational Swarm Intelligence, John Wiley & Sons.
6. Gordon, J. G. (1978), Structures or Why Things Don't Fall Down, Penguin, Baltimore.
7. Kennedy, J. And Eberhart, R. C. (1995), "*Particle swarm optimization*", In Proceedings of the 1995 IEEE International Conference on Neural Networks, volume 4, pages1942–1948. IEEE Press, Piscataway, NJ.
8. Lechuga, M. S. And Rowe, J. (2005), "*Particle swarm optimization and fitness sharing to solve multiobjective optimization problems*", In Congress on Evolutionary Computation (CEC'2005), pp. 1204–1211, Edinburgh, Scotland, UK.
9. Ott, E. (2002), Chaos in Dynamical Systems, Cambridge University Press, Cambridge, UK.
10. Reyes-Sierra, M. and Coello, C. A. (2006),"*Multi-Objective Particle Swarm Optimizers:A Survey of the State-of-the-Art* ", International Journal of Computational Intelligence Research (IJCIR)Vol.2, No.3, pp. 287–308.