Optimization
Iranian Journal of Optimization
Volume 11, Issue 1, 2019, 9-16
Research Paper
Online version is available on: www.ijo.iaurasht.ac.ir
Islamic Azad University
Rasht Branch
ISSN: 2588-5723
E-ISSN:2008-5427

Comparative Study of Particle Swarm Optimization and Genetic Algorithm Applied for Noisy Non-Linear Optimization Problems

Hossein Towsyfyan[1], Amin Kolahdooz[2*], Hazem Esmaeel[3] and Shahed Mohammadi[4]

[1] Department of Mechanical Engineering, University of Huddersfield, Huddersfield, UK
[2] Department of Mechanical Engineering, Khomeinishahr Branch, Islamic Azad University, Isfahan, Iran
[3] Department of Mechanical Engineering, University of Thi-Qar, Nasiriyah, Iraq
[4] Department of Computer Science and Systems Engineering, Ayandegan University, Tonekabon, Iran

Received: 17 November 2017
Accepted: 25 March 2018

Abstract

Optimization of noisy non-linear problems plays a key role in engineering and design problems. These optimization problems can't be solved effectively by using conventional optimization methods. However, metaheuristic algorithms such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) seem very efficient to approach in these problems and became very popular. The efficiency of these methods against many new metaheuristic optimization algorithms has been proved in previous works, however a robust comparison between GA and PSO to solve noisy nonlinear problems has not been reported yet. Therefore, in this paper GA and PSO are adapted to find optimal solutions of some noisy mathematical models. Based on the obtained results, GA shows a promising potential in terms of number of iteration to converge and solutions found so far for either for optimization of low or elevated levels of noise.

Keywords:
noisy non-linear problems
GA
PSO

***Correspondence E_mail**: Aminkolahdooz@gmail.com

INTRODUCTION

The optimisation of systems and processes is of high importance in manufacturing engineering, computational mechanics and control field. Optimisation problems are solved by using rigorous or approximate mathematical search techniques. Rigorous methods have employed linear programming, integer programming, dynamic programming or branch-and-bound techniques to approach the optimal solution for moderate-size problems (Yazdi et al., 2016; Chai-ead et al., 2011). These algorithms are guaranteed to find for every finite size instance of a combinatorial problem an optimal solution that exists in bounded time. Still, for combinational problems that are NP-hard, no polynomial time algorithm exists, assuming that P!=NP. Thus, complete methods might need exponential computation time in the worst-case scenario. This often leads to computation times that are usually too high for being useful in practical purposes (Pal et al., 2017). To overcome these problems, researchers have proposed approximate meta-heuristics algorithms to search for near optimal solutions (Elbeltagi et al., 2007).

Metaheuristic optimization techniques have become very popular and applied successfully to solve optimization problems arise in different industrial applications. These techniques already proved their simplicity, flexibility and efficiency in finding out optimal solutions of many complex optimization problems (Sai et al., 2016). Therefore recently, many new metaheuristic optimization algorithms such as Grey Wolf Optimizer (Mirjalili et al., 2014), Firefly Algorithm (Chai-ead et al., 2011 and Yang & He, 2013), Brain Storm Optimization algorithm (Zhang et al., 2012) and Imperialist Competitive Algorithm (Towsyfyan et al., 2013) were proposed in finding optimal solutions of noisy non-linear optimization problems. Based on the obtained results of different comparative studies, the efficiency of these new methods in solving noisy non-linear optimization problems still needs more of verifications and improvements (Sai et al., 2016; Azimi et al., 2017).

This paper aims to compare the Particle Swarm Optimisation (PSO) with Genetic Algorithm (GA) for solving noisy non-linear problems. Rest of the paper is organized as follows. The PSO and GA algorithms are briefly explained, this is next followed by a brief explanation of modeFRONTIER Software which is used for optimization in this work. Noisy non-linear mathematical models used for experimentation are presented in section 4. Experimental settings and results are then presented in next section. Last section finally concludes the paper.

EVOLUTIONARY ALGORITHMS & OPTIMIZATION

Particle swarm optimization (PSO)

PSO is a type of optimisation method that takes its basic concept from the behaviour of large groups of social animals. This may be a swarm of bees looking for a hive location or a flock of birds looking for food or a place to roost. It is a stochastic and population-based method. Eberhart and Kennedy were the first to check the validity of this method in optimisation (Eberhart & Kennedy, 1995). It is found that many problems of optimisation, as in Genetic Algorithms, can be worked out through the Particle Swan Optimisation technique.

The PSO system depends on the creation of a number of particles regarded as a swarm that aim at checking and flying over the hyper-dimensional solution space simultaneously (Kennedy, 2010). The mission of each single particle is to record their personal best position and read both local best (lbest) and swarm's best position (gbest). The velocity vector in such types of search is a driving factor that directs the particles so that they can be improved. For such an aim, velocity vector is always connected with personal best position (gbest), local best (lbest) and. An inertia factor, W, determines the influence that the previous velocities have on the current one. Additional factors such as cognitive and social factors are brought to have control over a particle and its confidence in itself or in the swarm. The main duty of the cognitive factor, C1, is to determine the level of confidence in success for each particle. The social factor C2 is responsible for detecting that confidence level. Table 1 shows the standard PSO nomenclature.

Genetic algorithm

Genetic Algorithm (GA) is an optimisation method that is non-deterministic and population-

Table 1: Nomenclature of Particle Swarm Optimisation

Symbol	Meaning
p_i	Particle i
$x_i(t)$	Particle position at time t
$v_i(t)$	Particle velocity at time t
lbest	Best position found by particle
gbest	Best position found by swarm
W	Inertia factor
C1	Cognitive factor
C2	Social factor

based. It was Holand who brought this technique to light (Holand, 1975). What marks this method is its dependence on imitating natural evolution: only the fittest will survive. In other words, the genetic properties of the parents are changed so that new generation of individuals will be fitter than the previous ones. For this change, mutation and crossover are used among other genetic processes to achieve the desired effect. Of course, global optimum is the utmost objective of these genetic operations, so they are modified and set for this purpose. Fig.1. shows the optimisation process of a GA. Currently, there are four major genetic operations that create the cornerstone of genetic algorithm technique: Tournament Selection, Crossover (Single-point crossover and Multiple-point crossover), mutation and elitism (Lin et al., 2003). In order to reduce the burdens arising from much computation while improving and increasing the search process to a greater extent, a number of various genetic operations and selection criteria for Genetic Algorithm have been examined for many years to find the most appropriate schemes to follow. The following schemes are the most popular ones used in the selection process and genetic modification.

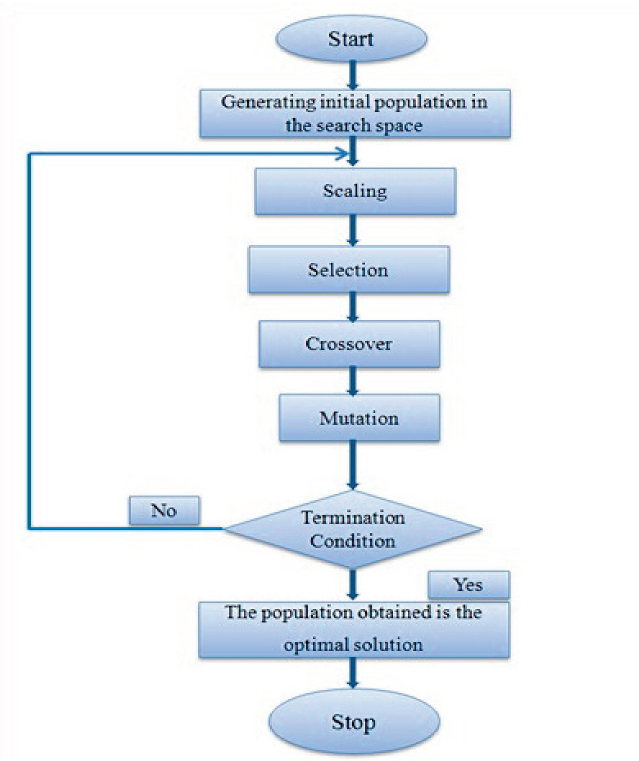


Fig. 1. Genetic algorithm flow chart

MODEFRONTIER SOFTWARE

The tool used to undertake the comparison of optimisation algorithms described above was modeFRONTIER. This package is a multidisciplinary, multi-objective design optimisation code, written to allow easy coupling to different commercial computer-aided engineering (CAE) tools. modeFRONTIER provides an environment in which product engineers and designers can integrate their various CAE tools, such as Finite Element Analysis software. The user manual of modeFRONTIER illustrates how a given problem can be handled (modeFRONTIER, 2008). Incorporating the analysis tool within the modeFRONTIER framework is reasonably straight forward with direct interfaces for Matlab and Simulink and other engineering softwares. In general, to understand the modeFRONTIER, Fig. 2. may be inspected, which indicates a simple example of both process design and optimisation. With modeFRONTIER three main steps are essential for achieving the goal: parameterise the problem, set objectives and choose the strategy for optimisation.

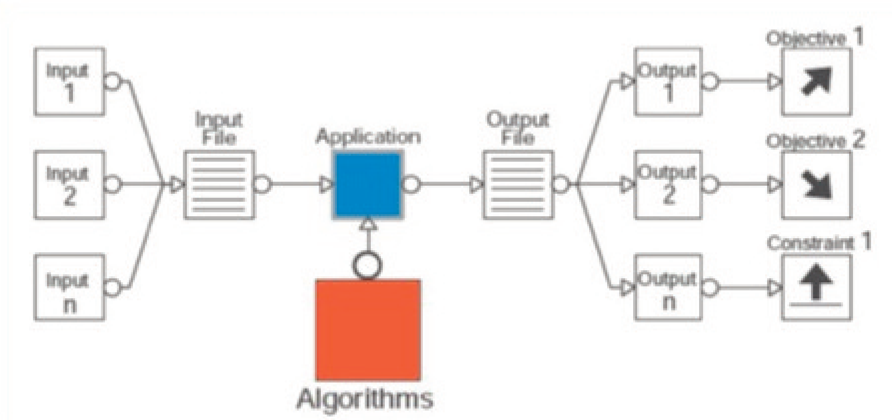


Fig.2. General modeFRONTIER process of integration and optimisation

NOISY NON LINEAR MATHEMATICAL FUNCTIONS

In this paper, PSO and GA were applied to analyze various types of non-linear mathematical models taken from literature. The typical natures of selected surfaces to be used in this study are the Branin Function, Sphere Function, Six-hump camel Function and Shubert Function. Considering the solution space in a certain region of 3D response surfaces, some models contain global optimum and multiple local optimums as described below.

Branin function

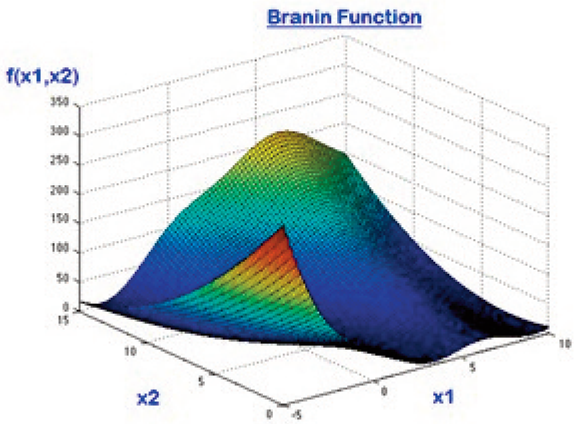
The Branin, or Branin-Hoo (see Fig.3.), has been investigated in several researches (modeFRONTIER, 2008, Molga et al., 2005 and Picheny et al., 2012). The recommended values of a, b, c, r, s and t are: $a = 1$, $b = 5.1/(4\pi^2)$, $c = 5/\pi$, $r = 6$, $s = 10$ respectively and $t = 1/(8\pi)$. This function is usually evaluated on the square $x1 \in [-5, 10]$, $x2 \in [0, 15]$ and has three global minima as follow:

$$f(x^*)=0.397887, at\ x^*=(-\pi,12.275),(\pi,2.275)\ and\ (9.42478,2.475) \quad (1)$$

Sphere functions

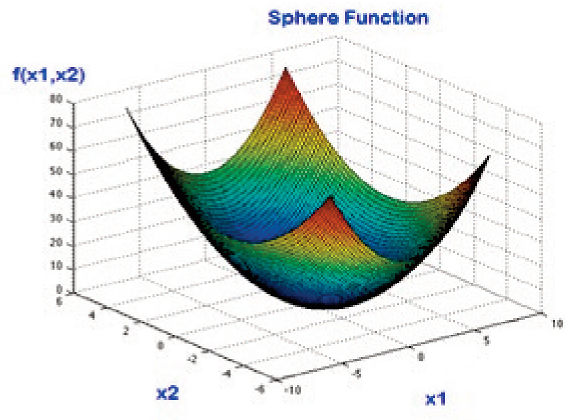
The Sphere function is continuous, convex and unimodal function. The plot shows its two-dimensional form in Fig. 4. The function is usually evaluated on the hypercube $xi \in [-5.12, 5.12]$, for all $i = 1, \ldots, d$ and has d local minima except for the global one. This function has been considered in several optimization works (Molga & Smutnicki, 2005; Picheny et al., 2012).

$$f(x^*)=0, at\ x^*=(0,\ldots,0)$$



$$f(\mathbf{x}) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1-t)\cos(x_1) + s$$

Fig.3. Matlab plot for Branin function



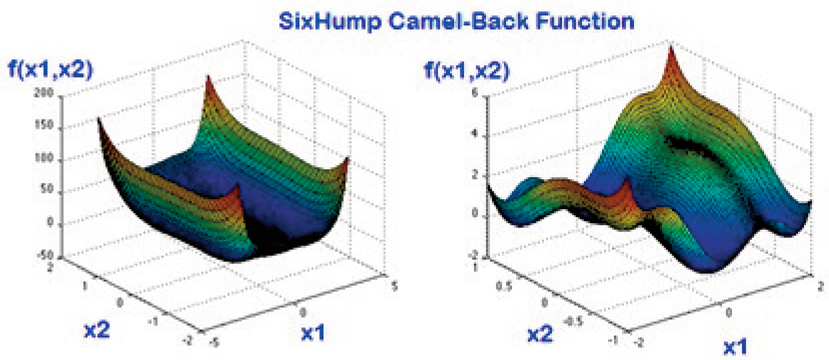
$$f(\mathbf{x}) = \sum_{i=1}^{d} x_i^2$$

Fig. 4. Matlab plot for sphere Function

Six-hump camel function

As shown in Fig. 5. the plot on the left shows the six-hump Camel function on its recommended input domain, and the plot on the right show only a portion of this domain, to allow for easier viewing of the function's key characteristics. The function is usually evaluated on the rectangle $x_1 \in [-3, 3]$, $x_2 \in [-2, 2]$ and has six local minima, two of which are global as follows (Molga & Smutnicki, 2005):

$$f(x^*)=-1.0316, at\ x^*= (0.0898,-0.7126)\ and\ (-0.0898,0.7126)$$



$$f(\mathbf{x}) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4+4x_2^2)x_2^2$$

Fig. 5. Matlab plot for Camelback Function

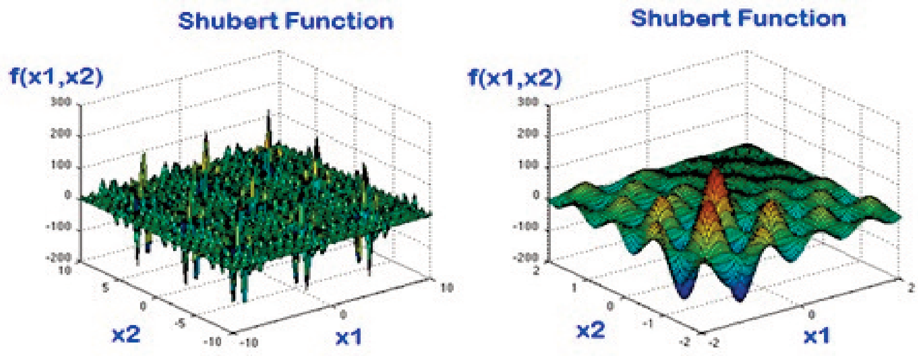
Shubert function

The Shubert function has several local minima and many global minima. As can be seen in Fig. 6. the second plot shows the function on a smaller input domain, to allow for easier viewing. The function is usually evaluated on the square $x_i \in [-10, 10]$, for all i = 1, 2,... although this may be restricted to the square $x_i \in [-5.12, 5.12]$, for all i = 1, 2,... Shubert function has a global minimum as follows (modeFRONTIER, 2008):

$$f(x^*)=-186.7309$$

RESULTS AND DISCUSSION

A careful investigation was carried out to compare the design efficiency of the GA and PSO algorithms. The algorithm parameters were determined to yield the best results as shown in Table 2.



$$f(\mathbf{x}) = \left(\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right)$$

Fig. 6. Matlab plot for Shubert Function

Table 2: Parameteres used in PSO and GA

	PSO Parameters	**GA Parameters**
Population Size	20, Sobol	20, Sobol
Max Generation	15, 20, 25, 30, 35	15, 20, 25, 30, 35
Crossover Percent	0.5	0.5
Crossover Percent	0.5	0.5

Due to the stochastic nature of the algorithms, each execution of the algorithm results in a different result, therefore in the entire study the best solution out of five different executions is presented as the optimization result. Table3 demonstrates the iteration process of GA and PSO method for optimization of four noisy non-linear functions. GA seems to be absolutely better in terms of speed of convergence for all target functions. The exact solutions are compared with results obtained by GA and PSO in Table 4. A significant accuracy is seen in the performance of the GA for all target functions.

Table 3: Number of iteration and best solution for both algorithms applied on four noisy non-linear mathematical models

Method		**GA**				**PSO**			
Function	**Number of Iteration**	**X1**	**X2**	**X3**	**Global Min**	**X1**	**X2**	**X3**	**Global Min**
Branin	15	-3.14	12.275	-	0.397887	-3.121	12.171	-	0.402903
	20	-3.14	12.275	-	0.397887	-3.121	12.171	-	0.402903
	25	-3.14	12.275	-	0.397887	-3.121	12.171	-	0.402903
	30	-3.14	12.275	-	0.397887	-3.139	12.314	-	0.399921
	35	-3.14	12.275	-	0.397887	-3.14	12.301	-	0.398215
Sphere	15	2 E-5	2 E-5	1 E-5	9 E-10	-0.048	-0.039	-0.038	0.005359
	20	0	0	1 E-5	1 E-10	-0.029	-0.022	-0.024	0.001982
	25	0	0	0	0	-0.033	0.0025	-0.027	0.001882
	30	0	0	0	0	-0.033	0.0025	-0.027	0.001882
	35	0	0	0	0	-0.033	0.0025	-0.027	0.001882
Six-Hump Camel	15	-0.0898	0.7126	-	-1.0316	-0.0886	0.7316	-	-1.0286
	20	-0.0898	0.7126	-	-1.0316	-0.0886	0.7316	-	-1.0286
	25	-0.0898	0.7126	-	-1.0316	-0.0886	0.7316	-	-1.0286
	30	-0.0898	0.7126	-	-1.0316	-0.0875	0.7180	-	-1.0314
	35	-0.0898	0.7126	-	-1.0316	-0.0875	0.7180	-	-1.0314
Shubert	15	-7.708	5.482	-	-186.7309	-7.680	5.476	-	-184.8377
	20	-7.708	5.482	-	-186.7309	-7.680	5.476	-	-184.8377
	25	-7.708	5.482	-	-186.7309	-7.680	5.476	-	-184.8377
	30	-7.708	5.482	-	-186.7309	-7.680	5.476	-	-184.8377
	35	-7.708	5.482	-	-186.7309	-7.712	5.480	-	-186.6680

Table 4: Comparison between exact solution and both algorithms applied on four noisy non-linear mathematical models

Function	Number of Iteration to get best solution		GA global minimum	PSO global minimum	Exact global minimum
	GA	PSO			
Branin	15	35	0.397887	0.398215	0.397887
Sphere	25	25	0	0.001882	0
Six Hump	15	30	-1.0316	-1.0314	-1.0316
Shubert	15	35	-186.7309	-186.6680	-186.7309

CONCLUSION

In this study, GA and PSO were applied in order to find the optimal solutions of noisy non-linear continuous mathematical functions. The results showed GA has a promising potential to be used as an effective tool in a variety of noisy nonlinear problems. GA seems to be absolutely better in terms of speed of convergence for all target functions. This might be due to the effect from generating the completely different random numbers to be used in the iterative procedures of the algorithm. When there was no noise on the process yields, the performance of both algorithms seems to be not so different to approach to the optimum. GA tends to be better, especially on the functions having multi-peaks. Complexity or difficulty level of the functions had no effect to the GA as expected.

REFERENCE

Azimi, M., Kolahdooz, A., & Eftekhari, S. A. (2017). An Optimization on the DIN1. 2080 Alloy in the Electrical Discharge Machining Process Using ANN and GA. *Journal of Modern Processes in Manufacturing and Production, 6*(1), 33-47.

Chai-ead, N., Aungkulanon, P., Luangpaiboon, P. (2011). Bees and Firefly Algorithms for Noisy Non-Linear Optimisation Problems. *Proceeding of the International Multiconference of Engineering and Computer Scientists,* Vol II, IMECS 2011, March 16-18, Hong Kong.

Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In Micro Machine and Human Science, 1995. MHS'95., *Proceedings of the Sixth International Symposium on* (pp. 39-43). IEEE.

Elbeltagi, E., Hegazy, T., & Grierson, D. (2007). A modified shuffled frog-leaping optimization algorithm: applications to project management. *Structure and Infrastructure Engineering, 3*(1), 53-60.

Holland, J. H. (1975). Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. Ann Arbor, MI: *University of Michigan Press*, 439-444.

Kennedy, J. (2010). Particle swarm optimization Encyclopedia of Machine Learning, *Springer*, 760-766.

Lin, W. Y., Lee, W. Y., & Hong, T. P. (2003). Adapting crossover and mutation rates in genetic algorithms. *J. Inf. Sci. Eng., 19*(5), 889-903.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software, 69*, 46-61.

modeFRONTIER, (2008). modeFRONTIER v4.0 User manual. Trierste, Italy: ESTECO srl.

Molga, M., & Smutnicki, C. (2005). Test functions for optimization needs. *Test functions for optimization needs*, 101.

Pal, S.K., Rai, C.S., Singh A.P. (2017). Comparative Study of Firefly Algorithm and Particle Swarm Optimization for Noisy Non-Linear Optimization Problems. *International Journal of Intelligent Systems and Applications, 4*(10), 50-57.

Picheny, V., Wagner, T., & Ginsbourger, D. (2013). A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization, 48*(3), 60.

Sai, V. O., Shieh, C. S., Lin, Y. C., Horng, M. F., Nguyen, T. T., Le, Q. D., & Jiang, J. Y. (2016, May). Comparative Study on Recent Development of Heuristic Optimization Methods. In *Computing Measurement Control and Sensor Network (CMCSN), 2016 Third International Conference on* (pp. 68-71). IEEE.

Towsyfyan, H., Salehi, S.A.A., Davoudi, GH., Bahmanpour M. (2013). A new approach to solve differential equations arising in fluid mechanics, *International Journal of Mathematical Modelling & Computation, 3*(2), 115-124.

Yang, X. S., & He, X. (2013). Firefly algorithm: recent advances and applications. *International Journal of Swarm Intelligence, 1*(1), 36-50.

Yazdi, M.S., Latifi Rostami, S.A., Kolahdooz, A. (2016). Optimization of geometrical parameters in a specific composite lattice structure using neural networks and ABC algorithm. *Journal of Mechanical Science & Technology, 30*(4), 1763-1771.

Zhan, Z. H., Zhang, J., Shi, Y. H., & Liu, H. L. (2012). A modified brain storm optimization. In E*volutionary Computation (CEC), 2012 IEEE Congress on* (pp. 1-8). IEEE.