A Comparison among Wolf Pack Search and Four other Optimization Algorithms

Shahla Shoghian, Maryam Kouzehgar

Abstract**—**The main objective of this paper is applying a comparison between the Wolf Pack Search (WPS) as a newly introduced intelligent algorithm with several other known algorithms including Particle Swarm Optimization (PSO), Shuffled Frog Leaping (SFL), Binary and Continues Genetic algorithms. All algorithms are applied on two benchmark cost functions. The aim is to identify the best algorithm in terms of more speed and accuracy in finding the solution, where speed is measured in terms of function evaluations. The simulation results show that the SFL algorithm with less function evaluations becomes first if the simulation time is important, while if accuracy is the significant issue, WPS and PSO would have a better performance.

Keywords**—**Wolf Pack Search, Particle Swarm Optimization, Continues Genetic Algorithm, Binary Genetic Algorithm, Shuffled Frog Leaping, Optimization.

I. Introduction

TRADITIONAL mathematical optimization imposes some difficulties on engineering problems which leads to development of alternative solutions such as evolutionary-based algorithms for searching near-optimum solutions to problems.

Evolutionary algorithms are stochastic search methods that are inspired from natural and social behavior of species. For example a recently developed idea is based on the behavior of wolfs that run a social life [1], [2], also the biologic behavior of genes and the interaction of birds or frogs in a group can be key issues while inspiration.

In order to imitate the behavior of these species, which is guided by learning, adaptation, and evolution, various researchers have suggested computational systems to seek for solutions. The first evolutionary-based technique introduced in the literature was the genetic algorithms [3]. the GAs technique has been used in many applications in science and engineering [4], [5] and [6]. However GA suffered from disadvantages such as high processing time and getting stuck in local minima. On the other hand in industrial applications such as robotics and aeronautics [7], [8] two key issues that play the main role are the consumed time and the quality of the answer.

In an attempt to reduce processing time and improve the quality of solutions other evolutionary algorithms are suggested such as: WPS [1], PSO [9] and SFL [10] among which WPS sounds rather new. In this paper, five evolutionary algorithms are reviewed with a special attention to the newly introduced Wolf Pack Search algorithm and Performance comparison among the five algorithms is then presented. The paper is organized as follows. Section II is concerned with a review on binary and continuous genetic algorithms. Section III and IV respectively focus on a brief review on PSO and SFL. Section V deals with elaborating on the wolf behavior to implement the WPS algorithm. Section VI introduce the test objective functions and presents the simulation results and finally section VII concludes the paper

II. Continues/Binary Genetic Algorithm

Genetic Algorithm (GA) was the first evolutionary-based optimization technique developed by Jon Holland [3] and popularized by David Goldberg [11]. GAs were developed on the basis of Darwinian principle and the natural process of evolution through reproduction. According to their proved ability to solve large and nonlinear problems, GA techniques have been used in many applications in science and engineering. GA introduces the solution to a given problem within a string called "chromosome" including a set of values representing for the optimization variables called "genes"[11]. GA starts with a random number of chromosomes. The chromosomes are made up of real numbers in continuous GA while they are strings of zero and ones in Binary GA. Each chromosome's fitness is evaluated by the cost function. In continuous GA the genes must be defined in the specified search domain or must be mapped to the considered interval before evaluation while in binary GA the genes must be decoded to real values before being served to the cost function. On the basis of natural ‘survival of the fittest’, the chromosomes are sorted according to their fitness. The worst ones – specified fraction of the population- are ignored and the better chromosomes exchange information to produce offspring chromosomes replacing the discarded ones. The information flow throughout the population is fulfilled via cross-over and mutation. We must have pairs of parents in order to apply crossover. Selection is applied among the better ones permitted to continue. The probability of selection for parent chromosomes is inversely proportional to their cost, i.e. the less the cost is-the more the fitness is-, the chromosome is more probable to be selected [12].

The offspring resulting from better parents will help breeding the population on next steps.

Shahla Shoghian (MSc. Graduate)was with the faculty of Electrical Engineering , Control Department, University of Tabriz, Tabriz , Iran, (phone:+98914051748, fax: +984114454673 e-mail: s.shoghian@gmail.com

Maryam Kouzehgar (Ph.D. Candidate) is with the faculty of Electrical Engineering, Control Department, University of Tabriz, Tabriz, Iran. (e-mail: mkouzehgar87@ms.tabrizu.ac.ir).

However in order not to hinder the diversification throughout the search space, the less-fitted chromosomes have been given little chances to survive to the next generation. After selecting and pairing the chromosomes, a heuristic reproduction (cross over) is applied on each pair resulting in a pair of offsprings replacing the discarded population.

Cross over in continuous GA includes some concepts of blending methods [12], while in binary GA cross over is carried out by exchanging bits between parents.

After crossover, during the mutation a certain percentage of the population, specified by the mutation rate, are randomly selected and substituted by another random value, usually resulting from adding a normally distributed random number to the original one in real-coded GA, while in binary-coded GA mutation evolves toggling bits [12]. The best chromosome does not take part in mutation due to elitism [12]. Injecting new genetic structures to the evolutionary process, mutation avoids premature convergence and stagnation around local minima [13]. Having produced a new generation, the population is re-evaluated. Finally the algorithm checks the stopping condition and if they are not fulfilled, another iteration of the algorithm is carried out.

III. Particle Swarm Optimization

PSO is an evolutionary computation technique developed by Kennedy and Eberhart in 1995[9]. In this algorithm each solution is regarded as a particle which is defined by its position and the fitness calculated based on the position. Also there is a speed vector which specifies the direction in which the particle is moving. Other parameters which are determined during the run, are as follows:

- P_{best} -the personal best: Each particle remembers the best position that it has visited so far. This best position is known as P_{best} .
- G_{best} - the global best: The best of all positions explored by all particles
- N_{ibest} -the neighborhood best: For each particle N_{ibest} is the best position of the particles in the neighborhood of i^{th} particle.

To apply the algorithm, first the particles are distributed randomly in the search space. Then the cost function is evaluated for each particle, afterwards P_{best} , N_{ibest} , G_{best} are updated. At the end by applying (1) the positions and speeds are updated. Eventually the algorithm checks the stopping criteria and loops until they are satisfied.

$$\begin{aligned} V_i &= wV_i + c_1 rand(N_{ibest} - X_i) + c_2 Rand(P_{best} - X_i) \\ X_i &= V_i + X_i \end{aligned} \tag{1}$$

where V_i is the velocity of i^{th} particle, X_i shows the position of the i^{th} particle, *w* is the inertia weight, utilized to avoid premature convergence and usually is set to 0.5. Separate random numbers are generated to accelerate through P_{best} and N_{ibest} . c_1 and c_2 are acceleration constants both equal to 2; these parameters change the amount of tension in the system, i.e. weighting the stochastic acceleration terms that pull the particle towards P_{best} or N_{ibest} . In some iterations N_{ibest} may be substituted by G_{best} .

Particle velocities are clamped to a maximum value of $V_{\max}$, thus serve a constraint on the global exploration ability. $V_{\max}$ is routinely adjusted at about 10-20% of the dynamic range of the variable on each dimension [14].

IV. Shuffled Frog Leaping

The SFLA is a search scheme benefiting from some concepts of memetic algorithm and particle swarm optimization (PSO). Memetic algorithm is a gene-based optimization algorithm similar to a GA in which chromosomes are represented by elements, called “memes” and differs from GA in applying a local search before cross-over and mutation [15]. The initial population is made up of frogs with random locations that are supposed to be divided into some groups called “memeplexes”. In each group a separate local search is carried out which is called memeplex evolution in which, on the basis of some PSO concept, the frogs move toward the best member in their group. After a definite number of generations spent on memeplex evolution, during the shuffling process, the frogs can share information and use experiences of one another. Afterwards the frogs separate the groups again. Until meeting the satisfied results, both local search and shuffling should be carrying on [10].

The process of the algorithm is as follows.

1. Generating initial population in the search domain.
2. Dividing the p frogs into m memeplexes each containing n frogs. (p=m*n).

In order to realize the division, first the frogs are sorted in terms of their fitness function and the first frog is classified in the first group, the second one in the 2nd group,.., and also the m(th) is classified in the m(th) group. Afterwards circularly the m+1(th) is classified in the 1st group and classification is continued until the last frog.

3. Memeplex evolution

The fitness function is evaluated and the best and worst frog in each memeplex are named as X_{best} and X_{worst} respectively, while the best frog among all is named X_{global} separately.

In each generation, the worst frog is improved during a PSO like scheme as illustrated in (2).

$$\begin{aligned} &D_i = rand() * (X_{best} - X_{worst}); \\ &-D_{max} < D_i < D_{max} \\ &X_{worst}(new) = X_{worst}(old) + D_i \end{aligned} \tag{2}$$

Then calculate fitness of newer X_{worst}, if the result does not became better than the old one, we repeat the above operation by replacing X_{best} by X_{global} , if we could not get the constructive consequence again, we select a new solution randomly instead of X_{worst}. These operations carry on until a pre-defined number of iterations for local search.

4. Shuffling

After memeplex evolution, all frogs of memeplexes are shuffled together.

5. Check the stopping criteria and if they are not satisfied go back to step 2.

The special parameters in this SFL Algorithm are: number of the initial population (p), number of groups (n), number of frogs in each group (m), number iteration of local search.

V. Wolf Pack Search

Most of the time, young wolves (not pups and old ones) instinctively separate from their initial packs in order to reproduction and seek their related pair and territory.

As soon as two alone wolves find each other, they move together and start to seek the territory. This correlation between two wolves will continue until one of them dies.

The emerging theory suggests that the wolves group would work i.e. group concentration is usually more successful in the reproduction than hunting. The packs are managed by two wolves that have higher social position and practically more freedom in comparison to the other wolves of the pack.

These two wolves (two first standing among the population) that are called alpha gain more food and also have exclusive lows for mating.

Most of the time the group chief (the best members) mate with each other, but in case of losing (death or injury) its counterpart mate, alpha wolf can also mate with one of the lower rank wolves. Even losing a sibling mate does not influence the chief and the alone wolf finds another mate for itself quickly.

Usually alpha pair is successful in raising its pups. The other wolves of the pack can mate but when they're in lack of resources such as food and time, the existing resources are devoted to the alpha pair children [1], [2].

The third wolf after the alpha pair is called Beta that more cares the alpha pair's children in comparison to the other wolves.

Also Beta wolf wants to obtain mastership position from the alpha wolf, but some of them, depending on the condition, prefer to hold the same third position.

A. Rank reduction

As illustrated above, rank reduction happens to an alpha wolf if it has passed away or injured. In this case the remaining alpha mate will find another mate for itself among the rest of the pack (preferably except beta). This rank reduction might happen in two ways: suddenly or gradually.

B. Suddenly

The older alpha wolf may give its rank to the fighter wolf peacefully i.e. within a special period of time it goes out of this cycle after a number of generations.

C. Gradually

In this case there is always a battle between alpha and another wolf. This fight may be just a grumbling or a real bloody battle. Defeated wolf usually is sent out of the pack and is sometimes killed by the other wolves.

If wolves leader 1 and leader 2 (two alpha wolfs) were similar to each other (when the similarity with chief (leader 1) is high), leader 2 will mate with the wolves of lower rank, in order to prevent twin offspring.

The high rank and position is mostly based on the character and attitude of leader wolf and not its body or physical power. That means the wolf pack search algorithm does not sort the population to determine the ranking, while the first rank is given to the one who, in comparison to previous generation, has greater changes in its cost function.

Pack size can change based on the amount of found food and characteristics of the leader wolves in the pack. Packs can have 2 to 20 wolves, but 8 wolves is a normal size.

Alone wolves seeking for one another must shelter to territories far away from the neighbor in order to feel safe enough against any probable attack.

The algorithm is as follows.

1. Generate initial population in the search domain randomly.
2. Evaluate each wolf.
3. Divide the search domain into some territories. (here 2 territories)
4. Randomly distribute the wolfs in the territories.
5. Apply a random move on each wolf in its territory.
6. Evaluate each neighborhood (wolfs in each territory) and determine the two ones with best improvement in comparison to the last generation. Name them Leader1 and Leader2.
7. In each territory Leader1 and Leader2 mate to generate two offspring which are exposed to a local search by the beta wolf. Then these offspring replace the two worst ones in each territory.
8. If Leader1 is changed in the last q iterations, then go back to step 4, else continue to step 9.
9. In the corresponding territory Leader1 is substituted by Leader2 and a random wolf is generated in that territory. Go back to 5
10. If by any chance leader1 and Leader2 are the same, Leader2 is replaced by a wolf of lower rank in order to avoid twins.

VI. Simulation Results

All the evolutionary algorithms described above are coded in Matlab® and the simulations is carried out on a 1.8 GHz AMD Laptop. The performance of the five evolutionary algorithms is compared using two benchmark problems whose description is given in the following.

A. F7 Function

This function is non-linear, non-separable, and involves two variables x and y, i.e. as illustrated in (3).

$$F7(x, y) = x\sin(4x) + 1.1y\sin(2y) \qquad (3)$$

For the variable values ranging from 0 to 10, the global optimum solution for this function is known to be -18.5547 when the variables (x,y) equal (0.9039, 0.8668). A sketch of this function is shown in Fig.1.

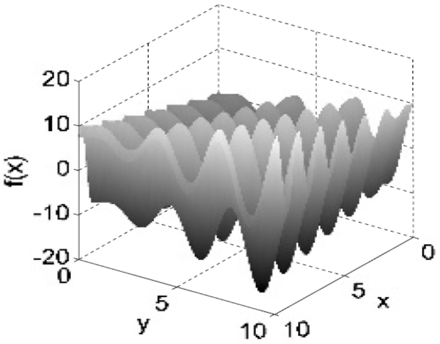

Fig. 1 A sketch of F7 function [11]

B. F15 Fuction

The objective function to be optimized as given by (4) is a nonlinear and non-separable function that involves two variables x and y.

$$F15(x,y) = -\exp(-0.2\sqrt{x^2+y^2} + 3(\cos 2x + \sin 2y)) \quad (4)$$

For the variable values ranging from -5 to 5, the global optimum solution for this function is known to be -16.947 when the variables (x,y) equal (-2.7730,-5). A sketch of this function is shown in Fig.2.

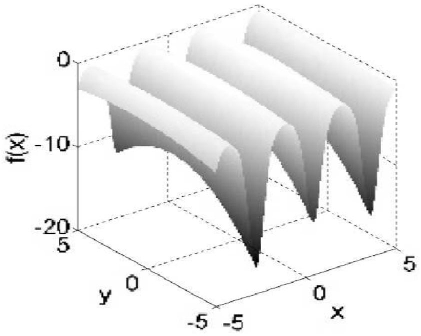

Fig. 2 A sketch of F15 function [11]

C. Results and Discussion

The convergence graph for the algorithms are demonstrated for F7 function in figures 3 to 8 and the Table I includes the corresponding minimum values and the number of function evaluations for the algorithms carried out on F7. Similarly figures 9 to 14 illustrate the convergence graphs for the algorithms when applied to solve F15 and Table 2 contains the quantitative results for F15.

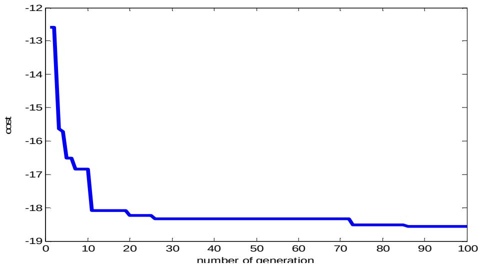

Fig. 3 CGA convergence graph for F7

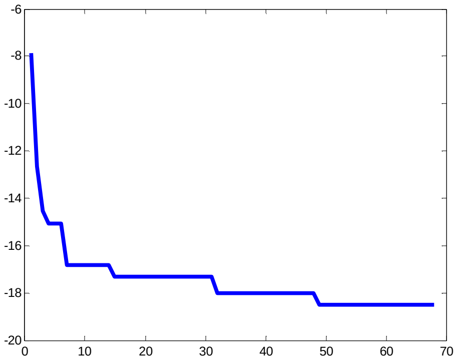

Fig. 4 BGA convergence graph for F7

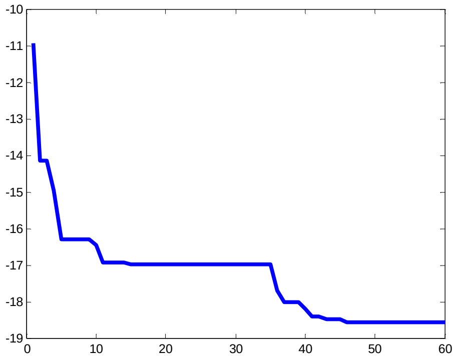

Fig. 5 PSO convergence graph for F7

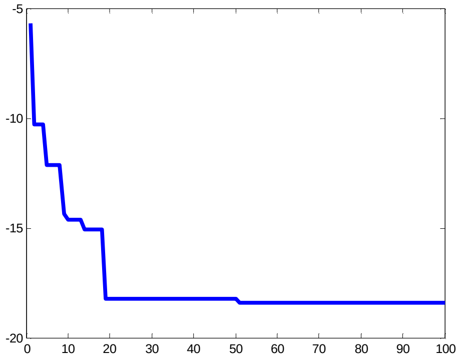

Fig. 6 SFL convergence graph for F7

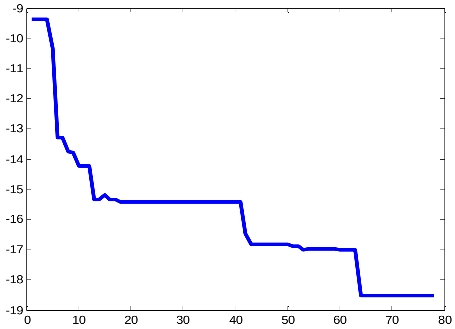

Fig. 7 The convergence graph for WPS with one territory applied on F7

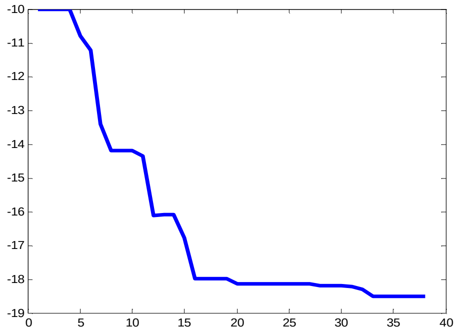

Fig. 8 The convergence graph for WPS with two territories applied on F7

TABLE I
Qualitative Results for F7

	Function Evaluation	Optimal Solution
BGA	1049	-18.4021
CGA	1600	-18.5254
PSO	2440	-18.5547
SFL	6501	-18.442
WPS with one territory	776	-18.5414
WPS with tow territory	1124	-18.5531

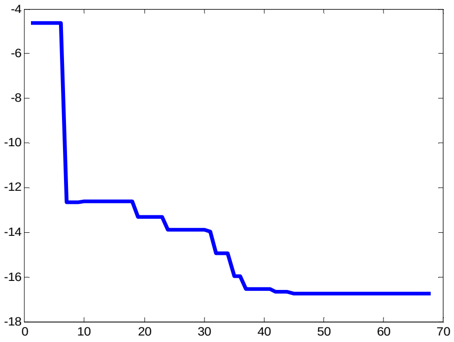

Fig. 9 CGA convergence graph for F15

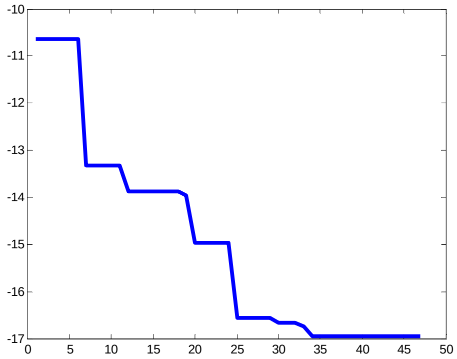

Fig. 10 BGA convergence graph for F15

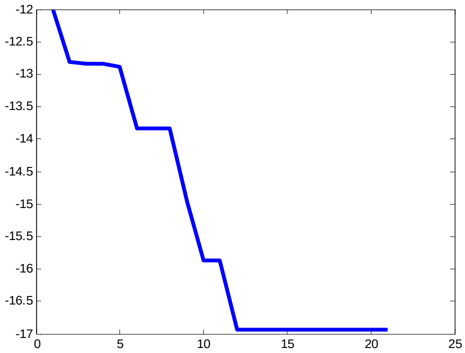

Fig. 11 PSO convergence graph for F15

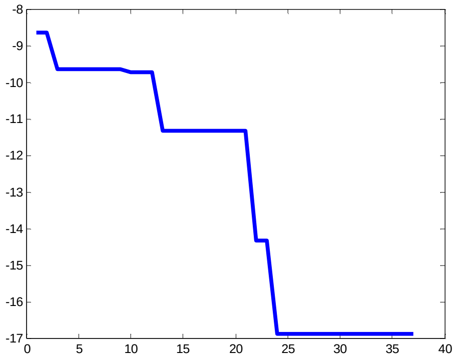

Fig 12 SFL convergence graph for F15

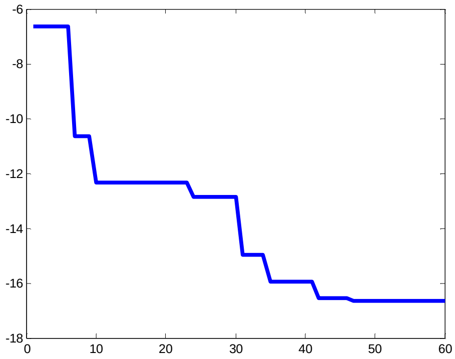

Fig 13 The convergence graph for WPS with one territory applied on F15

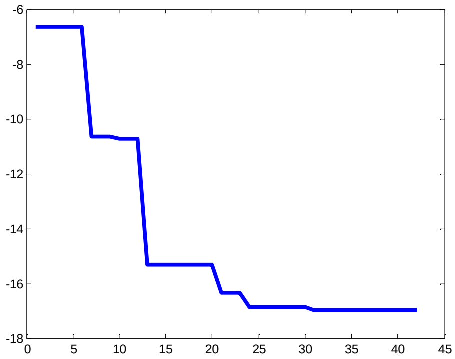

Fig. 14 The convergence graph for WPS with two territories applied on F15

TABLE II
QUALITATIVE RESULTS FOR F15

	Function Evaluation	Gbest1
BGA (Binary Genetic Algorithm)	1011	-16.834
CGA (Continues Genetic Algorithm)	1600	-16.774
PSO (Particle Swarm Optimization)	840	-16.9377
SFL (Shuffled Frog Leaping)	5003	-16.653
WPS (Wolf Pack Search) with one territory	890	-16.954
WPS (Wolf Pack Search) with tow territory	1300	-16.946

VII. CONCLUSION

Achieving the goals basically depend on two key issues, less number of function evaluations (higher speed), or accuracy in approaching to the answer. For instance, in some fields such as missile control, both of those issues are equally important and inseparable from each other.

Accordingly on the basis of figures it is quite obvious that if speed (less function evaluation) is the more important criterion, SFL algorithm is the most recommended one among the compared ones here. However if the accuracy in reaching to the specific point is of greater importance, it would be suggested to choose between WPS or PSO algorithm.

REFERENCES

[1] Gray Wolf - Wikipedia, the free encyclopedia, pp. 1-19.

[2] "Gray Wolf". Discover life in America. Retrieved on 2005-05-05

[3] Holland J. “Adaptation in natural and artificial systems”. Ann Arbor, MI: university of Michigan Press; 1975.

[4] Al-Tabtabai H, Alex PA. Using genetic algorithms to solve optimization problems in construction. Eng Constr Archit Manage 1999;6(2):121–32.

[5] Hegazy T. Optimization of construction time-cost trade-off analysis using genetic algorithms. Can J Civil Eng 1999;26:685–97.

[6] Grierson DE, Khajehpour S. Method for conceptual design applied to office buildings. J Comput Civil Eng 2002;16(2):83–103.

[7] I. Hassanzadeh, S. Mobayen, Optimum Design of PID Controller for 5-bar-linkage Manipulator Using Particle Swarm Optimization, Proceeding of the 4th t International Symposium on Mechatronics and its Applications (ISMA07), Sharjah, U.A.E. March 26-29, 2007

[8] Mraceck, C.P. and Cloutier, J.R., Missile Longitudinal Auto-Pilot Design using the State Dependant Riccati Equation Method, Proceeding of the International Conference on Nonlinear Problems in Aviation and Aerospace, Daytona Beach, F.L., may, 1996.

[9] R.C. Eberhart, and J. Kennedy, “A New Optimizer Using Particle Swarm Theory”, Proceeding of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp.39-43, 1995.

[10] Eusuff MM, Lansey KE. Optimization of water distribution network design using the shuffled frog leaping algorithm. J Water Resour Plan Manage 2003;129(3):210–25.

[11] Goldberg DE. “Genetic algorithms in search, optimization and machine learning”. Reading, MA: Addison-Wesley Publishing Co; 1989.

[12] R.L. Haupt, S.E. Haupt, “Practical Genetic Algorithms”, John Wiley, 2004.

[13] E. Elbeltagi, T. Hegazy, and D. Grierson, “Comparison among Five Evolutionary-Based Optimization Algorithms,” Advanced Engineering Informatics, vol. 19, no. 1, pp. 43–53, 2005.

[14] Russell C. Eberhart and Yuhui Shi, “Particle Swarm Optimization: Developments, Applications and Resources,”. IEEE 2001.

[15] Moscato P. On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Technical Report Caltech Concurrent Computation Program, Report 826, California Institute of Technology, Pasadena, CA; 1989.

Shahla Shoghian was born in Tabriz, Iran in 1983. She received her B.Sc. and M.Sc. degrees in Electrical Engineering-Control from Tabriz University, Tabriz, Iran respectively in 2006 and 2010. Her current interests include intelligent optimization using artificial intelligence algorithms and expert systems and applications of fuzzy theory.

Maryam Kouzehgar was born in Tabriz, Iran in 1986. She became a student member of IEEE (M) in 2009. She received her B.Sc. and M.Sc. degrees in Electrical Engineering-Control from Tabriz University, Tabriz, Iran respectively in 2008 and 2010. She is currently a Ph.D. student in the Department of Control Engineering at Tabriz University. Her current interests include intelligent optimization using artificial intelligence algorithms and expert systems, applications of fuzzy theory, Petri nets, human behavior representation and arial image processing.