Journal of Advanced Computer Science & Technology, 8 (2) (2019) 40-45

Journal of Advanced Computer Science & Technology

Website: www.sciencepubco.com/index.php/JACST

Research paper

A comparative study of particle swarm optimization and genetic algorithm

Saman M. Almufti [1] *, Amar Yahya Zebari [2], Herman Khalid Omer[1]

[1] Department of Computer Science and Information Technology, College of Computer Science & Information Technology, Nawroz University, Duhok, Iraq

[2] Department of Statistics, College of Science, Van Yuzuncu Yil University, Van, Turkey

**Corresponding author E-mail: Saman.Almofty@gmail.com*

Abstract

This paper provides an introduction and a comparison of two widely used evolutionary computation algorithms: Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) based on the previous studies and researches. It describes Genetic Algorithm basic functionalities including various steps such as selection, crossover, and mutation.

Keywords: *Particle Swarm Optimization (PSO); Genetic Algorithms (GAS); Swarm Intelligence; PSO and GA Comparison.*

1. Introduction

Particle Swarm Optimization (PSO) and Genetic algorithms (GAs) belongs to evolutionary computation optimization methods, which are widely used in the computer science fields to solve real life problems (Kennedy and Eberhart 1995).

Particle Swarm Optimization (PSO) algorithm belongs to swarm intelligence fields (Almufti, 2017) (Almufti, 2019); it's concerned with the designing of intelligent interactive multi-agent systems that cooperate to gather to achieve a specific goal (Almufti, Marqas, & Asaad, 2019) (Almufti & Shaban, 2018) (Almufti, 2019). PSO is designed as an inspiration from the social behaviors of real living organisms such as bird flocking, fishing schooling, it was first introduced by Eberhart Kennedy and its widely used to solve computational problems (Kennedy and Eberhart 1995), (Hochbaum, S. 1997), (Almufti, 2017).

Genetic algorithms (GAs) are powerful search techniques used to solve problems in many different disciplines (Alba, E. and Troya, J. 1999), based on the mechanisms of natural genetics and natural selection it simulates a survival of good fitting gene among the species which is created by random changes in the gene-structure of the chromosomes in the evolutionary biology (D.E. Goldberg. 1989). New generation of solutions is created from solutions in previous generation. Genetic algorithms basic strategy to find the best solutions is to crossover the parent genes (Lim, S., Sultan, A., Sulaiman, M., Mustapha, A. and Leong, K. 2017). Various crossover techniques are founded to obtain the optimum solution as early as possible in minimum generations.

2. Particle swarm optimization (PSO)

The Particle Swarm Optimization (PSO) algorithm is a computational optimization method, which is based on the social behavior of living organisms such as fishing schooling, bird flocking (Almufti, 2017) (Almufti & Shaban, 2018). The Particle Swarm Optimization algorithm was first designed in 1995 by the scientists Kennedy and Eberhart base on the social learning and social psychological model of birds seeking food, each bird is an agent in the search space, during the food search time, swarm agents are cooperates with other agents around it to find food (Kennedy, J., and Mendes, R. 2002).

Particle swarm optimization (PSO) is a computational method that is used in computer science field to optimize problem solution by iteratively improvement of a candidate solution among a set of solutions with regard to a given measurement of fitness quality (Asaad, R., Abdulnabi, N. 2018) (Almufti & Shaban, 2018). Such methods are commonly known as metaheuristics methods (Almufti, 2017). PSO algorithm fallows many steps to generate a optimal or near-optimal solution for any given problem as shown in figure 1.

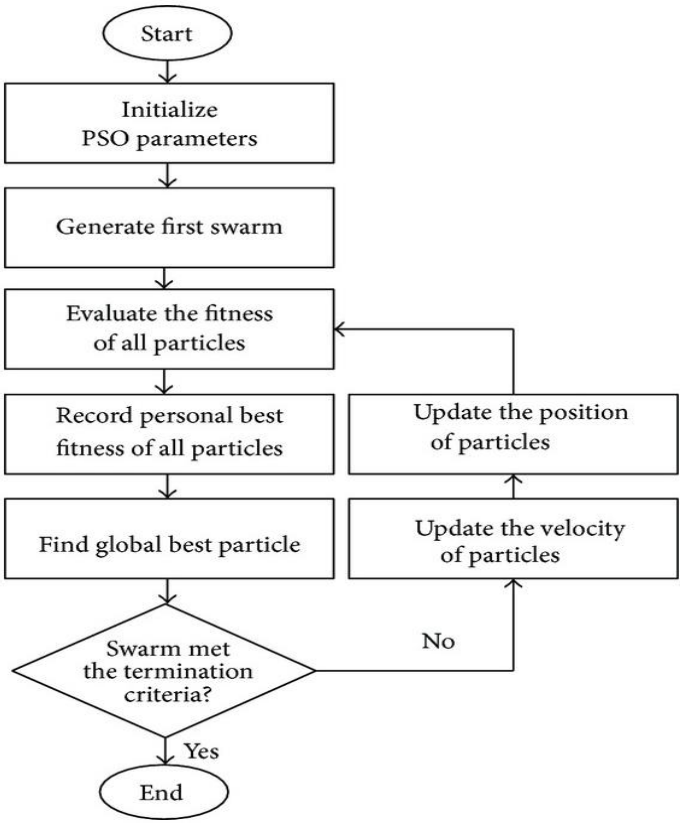

Fig. 1: PSO Flowchart.

3. PSO Significant differences

- Derivative-free: unlike many other conventional techniques, PSO is a derivative-free algorithm.
- Flexibility: PSO has the flexibility of integration with different optimization techniques to build a hybrid algorithm.
- Parameter: unlike many other evolutionary techniques, PSO has less parameter to adjust.
- Escape local minima: PSO is able to escape from local minima.
- Easy for implementation: PSO can easily implement and program to work with basic logic and mathematical operations.
- Handle objective functions: PSO is able to handle objective functions that have a stochastic nature, such as representing one of the optimization variables as random.
- Initial solution: In PSO a good initial solution is not required to start its iteration process.

4. Applications of particle swarm optimization algorithm

Particle swarm optimization has been used across a wide range of applications. Areas where PSOs have shown particular promise include multimodal problems and problems for which there is no specialized method available or all specialized methods give unsatisfactory results (Poli, R., Kennedy, J. and Blackwell, T. 2007).
The main PSO application categories are as follows:

- Communication networks design
- Systems of fuzzy and neuro-fuzzy
- Signal processing
- Neural networks
- Data mining, classification and clustering.
- Sensors and sensor networks
- Metallurgy applications.
- Medical, biological, and pharmaceutical applications
- Robotics applications.
- Design of antennas
- Restructuring and Designing electricity networks and load dispatching
- Computer graphics and visualization
- Image and video analyzing and optimizing applications
- Control applications
- Electronics and electromagnetics Applications
- Design applications
- Modeling systems.
- Power systems and power generation applications
- Scheduling applications

- Security and military applications
- Economics and Finance systems.
- Combinatorial optimization problems
- Detecting or diagnosing of faults and the recovery from them.

5. Genetic algorithm (GA)

The genetic algorithm (GA) is an evolutionary computation algorithms (Almufti, Marqas & Ashqi, 2019), introduced and investigated by the scientist John Holland and one of his students at the University of Michigan at the beginning of the 60s. In 1975 they published the first achievement of GA entitled "Adaptation in Natural and Artificial System" (Holland, J. H. 1975, DeJong, K. 1975), this algorithm is successfully adapted for solving complex problems such as NP-Hard problems, where the optimal solution is hard to find (Almufti, Marqas & Ashqi, 2019).
Genetic algorithm is inspired by the principles of genetics and evolution, and mimics the reproduction behavior observed in biological populations. It belongs to stochastically search algorithm bases on principles of natural selection and recombination.
GA evolves a population of initial individuals called chromosomes, where each chromosome represents a solution to the problem to be solved. Each chromosome composed of predetermined number of genes. Then applying recombination to these structures, such as crossover and mutation (Thengade, A. and Dondal, R. 2012), they attempt to reach to the optimal solution of the given problem by manipulating a population of candidate solutions. Then the manipulated population is evaluates and the survived best solutions are passed to reproduce and mate to form the next generation. Gradually after a number of generations, good traits dominate the population, resulting a good solution for the problem as shown in (figure 2).
Darwinian evolution is the basic mechanism in GAs in which the bad traits that appear in individuals, which do not survive the process of selection are iteratively eliminated from the population. The good traits, which survives are mating to form better individuals.

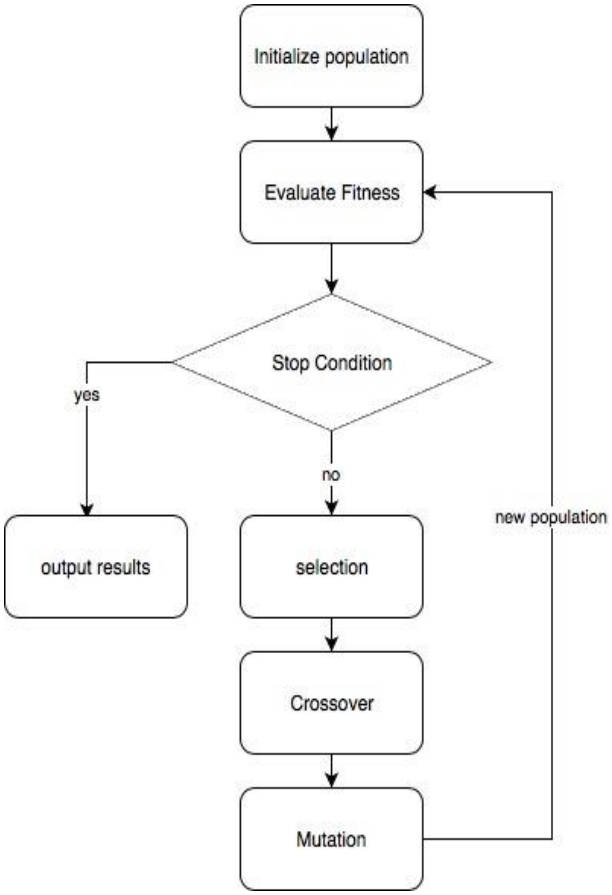

Fig. 2: GA Flowchart.

6. Basic recombination operations of genetic algorithm

6.1. Selection operation

Genetic Algorithm can use many different techniques to select the individuals to be copied over into the next generation (Mazza, C., Piau, D. 2001). In this study we compared the following crossover operators:

- Roulette-wheel selection is a fitness-proportionate selection in which the chance of selecting an individual's is proportional to the deferent between its fitness and its competitors' fitness.
- Elitist selection: It certainly selects the fittest members of each generation.
- Fitness-proportionate selection: fittest individuals are more likely, but not always, to be selected.
- Tournament selection: The population is divided into Subgroups of individuals, and members of each subgroup compete against each other. From each subgroup only one individual is chosen to reproduce for the next generation.

6.2. Crossover operators

Once the individuals have been selected the next step is to produce the new generation (offspring). Crossover is the operation of merging genetic information of two parent individuals to form a new generation (offspring). Genetic Algorithm can use many different crossover techniques (Kora, P. and Yadlapalli, P. 2017). In this study we compared the following two simple and widely used crossover operators:

- Single-point crossover: Single-point crossover composes of two steps. First, members of the chromosomes are mated randomly. Second, for each pair of chromosomes an integer position k is selected randomly between 1 and the chromosome length l. then two new chromosomes are generated by swapping all the genes between k+1 and l (Goldberg D. E., 1989) as shown in figure 3.

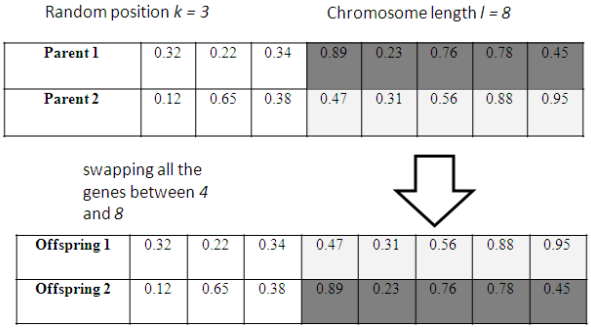

Fig. 3: Single-Point Crossover.

- Two-point crossover: This type of crossover is similar to Single-point crossover; the only difference occurs is that for each pair of chromosome it requires selecting two positions k and s between 1 and chromosome length l instead of one. Then the all genes between the two selected positions are swapped (Sywerda G., 1989) (A.J., U. and P.D., S. 2015) as shown in figure 4.

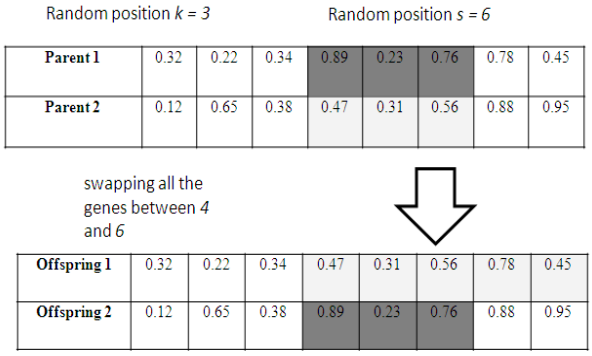

Fig. 4: Two-Point Crossover.

6.3. Mutation operation

After selection and crossover, mutation operator is applied. Selection and crossover generate individuals from the parents, some are directly copied, and others are produced by crossover. In order to ensure that the individuals are not similar to each other, GA allows for a small chance of mutation (Coello, Carlos., 2000). A mutation operator takes a solution as an input and changes it to generate another solution it alters one or more gene values in a chromosome from its initial state to generate an individual that is differed for its parents (Li, Z., Liu, X., Duan, X. and Huang, F. 2010). In this study we compared the following mutation operators:

- Inversion Mutation: pick two genes at random and then reverse the genes between them (Sivanandam S. N. , Deepa S. N., 2007).

For the following chromosome: 0 1 2 3 4 5 6 7
Two genes (1 and 4) are randomly selected 0 1 2 3 4 5 6 7
The genes between 1 and 4 are inverted 0 4 3 2 1 5 6 7

- Scramble Mutation: In this mutation, randomly select a subset of genes at and then rearrange the genes in those positions randomly. Subset does not have to be contiguous.

For the following chromosome: 0 1 2 3 4 5 6 7
A subset of gene (3456) is randomly selected 0 1 2 3 4 5 6 7
The genes of the selected subset are randomly scrabbled 0 1 2 5 6 3 4 7

- Swap Mutation: Simple in a chromosome it selects two genes at random and swap their positions. (Sivanandam S. N. , Deepa S. N., 2007).

For the following chromosome: 5 3 2 1 7 4 0 6
Randomly choose two genes (i.e.; 3 and 4) and swap them: 5 4 2 1 7 3 0 6

7. GA significant differences

- Genetic algorithm operates on a whole population (strings), this improves the chance to reach a global optimum and reduces the risk of becoming trapped in a local stationary point.
- Genetic algorithm manipulates coded versions of the problem parameters instead of the parameters themselves.
- Normal genetic algorithms can be applied to any kind of discrete or continuous optimization problem because GA does not use any auxiliary information about the objective function value such as derivatives.
- Genetic algorithm uses probability in transition operators while conventional methods apply deterministic transition operators.

8. Applications of genetic algorithm

Genetic algorithms have been used for many real life problems. Some applications of GA are shown bellow:

- Combinatorial Optimization: traveling salesman (TSP), Sequence scheduling, routing, bin packing, graph coloring and partitioning.
- Nonlinear dynamical systems predicting, data analysis
- Functions for creating images
- Strategy planning
- Scheduling–manufacturing, resource allocation and facility scheduling
- Robot trajectory planning
- Evolving LISP programs (genetic programming)
- Finding shape of protein molecules
- TSP and sequence scheduling
- Machine Learning–Designing neural networks.
- Signal Processing filter design

9. Advantage of PSO over GA

The following are some advantages of PSO over GA

Genetic Algorithm	Particle swarm optimization
Genetic algorithm requires some genetic operator like as crossover, mutation, selection	PSO only requires a few parameters to adjust, easy to implement.
GA computational cost is very high	PSO minimize this function.
GA checks only present fitness function.	PSO checks local and global functions
GA doesn't have memory to store previous fitness value,	PSO have memory to store previous fitness value
Difficult in implementation	Easy to implement

10. Summary

This paper presented a comparison study between two famous metaheuristics algorithms: Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). PSO is a heuristic search method that is inspired from the collaborative behavior of real living swarms.
Both PSO and GA are similar in that they are population-based search approaches and both depend on information's sharing among their population members to enhance their search processes using a combination of deterministic and probabilistic rules. During the time both PSO and GA are used to solve many problems and they are well-established algorithm with many versions and many applications.
This paper shows some advantages of PSO over GA, as it is easer to use dose not requires a good initial state to start its iterations to word the goals.

References

[1] Alba, E. and Troya, J. (1999). A survey of parallel distributed genetic algorithms. Complexity, 4(4), pp.31-52. https://doi.org/10.1002/(SICI)1099-0526(199903/04)4:4<31::AID-CPLX5>3.0.CO;2-4.

[2] A.J., U. and P.D., S. (2015). CROSSOVER OPERATORS IN GENETIC ALGORITHMS: A REVIEW. ICTACT Journal on Soft Computing, 06(01), pp.1083-1092. https://doi.org/10.21917/ijsc.2015.0150.

[3] Almufti, S. (2017). Using Swarm Intelligence for solving NPHard Problems. Academic Journal of Nawroz University, 6(3), pp. 46-50. https://doi.org/10.25007/ajnu.v6n3a78.

[4] Almufti, S. (2015). U-Turning Ant Colony Algorithm powered by Great Deluge Algorithm for the solution of TSP Problem. [online] Hdl.handle.net. Available at: http://hdl.handle.net/11129/1734 [Accessed 5 Aug. 2018].

[5] Almufti S., & Shaban A., (2018), U-Turning Ant Colony Algorithm for Solving Symmetric Traveling Salesman Problem, Academic Journal of Nawroz University, vol. 7, no. 4, pp. 45-49, Available: 10.25007/ajnu. v6n4a270. https://doi.org/10.25007/ajnu.v6n4a270.

[6] Almufti, S., R. Asaad, R., & B. Salim, (2019). Review on Elephant Herding Optimization Algorithm Performance in Solving Optimization Problems. International Journal of Engineering & Technology, 7(4), 6109-6114.

[7] Almufti, S., Marqas, R., & Ashqi V., (2019). Taxonomy of bio-inspired optimization algorithms. Journal Of Advanced Computer Science & Technology, 8(2), 23. https://doi.org/10.14419/jacst.v8i2.29402.

[8] Almufti, S., Marqas, R., & Asaad, R. (2019). Comparative study between elephant herding optimization (EHO) and U-turning ant colony optimization (U-TACO) in solving symmetric traveling salesman problem (STSP). Journal Of Advanced Computer Science & Technology, 8(2), 32. https://doi.org/10.14419/jacst.v8i2.29403.

[9] Asaad, R., Abdulnabi, N. (2018). Using Local Searches Algorithms with Ant Colony Optimization for the Solution of TSP Problems. Academic Journal of Nawroz University, 7(3), 1-6. https://doi.org/10.25007/ajnu.v7n3a193.

[10] Coello, Carlos. "An updated survey of GA-based multiobjective optimization techniques." ACM Computing Surveys, vol.32, no.2, p.109-143 (June 2000). https://doi.org/10.1145/358923.358929.

[11] D. E. Goldberg. Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley, 1989.

[12] D.E. Goldberg. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, New York,

[13] DeJong, K. 1975. An Analysis of the Behavior of a Class of Genetic Adaptive Systems, PhD Dissertation, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor.

[14] G. Sywerda, Uniform crossover in genetic algorithms, in Proceedings of the 3rd International Conference on Genetic Algorithms, 1989, pp. 2-9.

[15] Hochbaum, S. (1997), Approximation Algorithms for NP-Hard Problems. PWS Publishing Company, Boston. https://doi.org/10.1145/261342.571216.

[16] Holland, J. H. 1975. Adaptation in Natural and Artificial Systems, University of Michigan Press. Ann Arber.

[17] Kennedy J., Eberhart R. (1995), Particle swarm optimization, in: IEEE Inter-national Conference on Neural Networks Proceedings, vols. 1– 6, pp.1942–1948.

[18] Kennedy, J., and Mendes, R. (2002). Population Structure and Particle Swarm Performance. Proceedings of the 2002 World Congress on Computational Intelligence. https://doi.org/10.1109/CEC.2002.1004493.

[19] Kora, P. and Yadlapalli, P. (2017). Crossover Operators in Genetic Algorithms: A Review. International Journal of Computer Applications, 162(10), pp.34-36. https://doi.org/10.5120/ijca2017913370.
[20] Li, Z., Liu, X., Duan, X. and Huang, F. (2010). Comparative Research on Particle Swarm Optimization and Genetic Algorithm. Computer and Information Science, 3(1). https://doi.org/10.5539/cis.v3n1p120.
[21] Lim, S., Sultan, A., Sulaiman, M., Mustapha, A. and Leong, K. (2017). Crossover and Mutation Operators of Genetic Algorithms. International Journal of Machine Learning and Computing, 7(1), pp.9-12. https://doi.org/10.18178/ijmlc.2017.7.1.611.
[22] Mazza, C. and Piau, D. (2001). On the effect of selection in genetic algorithms. Random Structures and Algorithms, 18(2), pp.185-200. https://doi.org/10.1002/1098-2418(200103)18:2<185::AID-RSA1005>3.0.CO;2-7.
[23] Poli, R., Kennedy, J. and Blackwell, T. (2007). Particle swarm optimization. Swarm Intelligence, 1(1), pp.33-57. https://doi.org/10.1007/s11721-007-0002-0.
[24] Sivanandam S.N. and Deepa S. N.2007, Introduction to Genetic Algorithms, Springer, ISBN 9783540731894.
[25] Thengade, A. and Dondal, R. (2012). Genetic Algorithm – Survey Paper. International Journal of Computer Applications (IJCA), (0975 - 8887), pp.25-29.