Correlation with the fundamental PSO and PSO modifications to be hybrid swarm optimization

Raed A Hasan^{1,*}, Suhel Shahab Najim^{1,2}, Munef Abdullah Ahmed³

¹Electronics Department - Alhawija Technical Institute, Northern Technical University, IRAQ

²Faculty of Computer System and Software Engineering, University Malaysia Pahang, Malaysia

³Faculty of Automatic Control and Computers, University Polytechnic of Bucharest 313 Splaiul Independentei, 060042, Romania

*Corresponding Author: Raed A Hasan

DOI: <https://doi.org/10.52866/ijcsm.2021.02.02.004>

Received April 2021; Accepted Jun 2021; Available online July 2021

ABSTRACT: A swarm is a group of a single species in which the members interact with one another and with the immediate environment without a principle for control or the emergence of a global intriguing behavior. Swarm-based metaheuristics, including nature-inspired populace-based methods, have been developed to aid the creation of quick, robust, and low-cost solutions for complex problems. Swarm intelligence was proposed as a computational modeling of swarms and has been successfully applied to numerous optimization tasks since its introduction. A correlation with the fundamental Particle Swarm Optimization (PSO) and PSO modifications demonstrates that hybrid swarm optimization outperforms existing strategies. The downside of hybrid swarm optimization is that it frequently tends to arrive at suboptimal solutions. As such, efforts are being made into combining HSO and other algorithms to arrive at better quality solutions.

Keywords: Swarm intelligence; PSO; ACO; Hybrid swarm intelligence.

1. INTRODUCTION

A swarm is a group of a single species in which the members interact with one another and their immediate environment without a principle for control or the emergence of a global intriguing behavior. Swarm-based algorithms have risen as nature-inspired and populace-based procedures that are suitable for creating low-cost, quick, and robust solutions for complex problems. Swarm intelligence [1] can be distinguished as a generally new discipline of artificial intelligence (AI) that facilitates the computer modeling of social life pattern of swarms, such as honeybees, flocking of birds, and ants in their colonies [2] [3]. Although these agents are moderately naive and have limited capacities on their own, they tend to interact with each other in a specific pattern to achieve great tasks that support their existence. Social collaboration among swarm communities can be direct or indirect. They interact directly through audio and visual contacts, such as the honeybees' waggle dance. An indirect interaction occurs when one organism deviates from the environment, making the other organisms react to a newly changed environment. An example of indirect interaction is an ant's pheromone trails that it places on its conduit to explore sources of food. An indirect interaction among swarms is referred to as stigmergy, which means a form of communication by means of the environment.

Many researchers have recently focused on the study of social behavior of insects to understand the working principle of nature-inspired frameworks. Computer scientists have projected scientific insights into these natural swarm frameworks to the AI ground since the 1980s. Aimed at controlling a robotic swarm, [4] coined the term "Swarm Intelligence" in the area of global optimization as a set of algorithms in 1989. [5] introduced the Ant Colony Optimization (ACO) algorithm as a new nature-inspired metaheuristic that aims at providing solution to combinatorial optimization issues. [6] introduced the PSO, which mimicks the social flocking behavior of birds. These two famous SI algorithms have been widely used in

different fields of study since their introduction; various modifications have been made on the algorithms to improve their performance in other fields since the 1990s. Later in 2005, D. Karabago presented Artificial Bee Colony Algorithm as a new SI algorithm [7].

The computational modeling of the natural behavior of swarms has encouraged the adoption of SI algorithms in various optimization tasks because they have been effectively applied to several domains, such as optimal route identification, function optimization issues, structural optimization, scheduling, and data and image analysis. Furthermore, computational swarm optimization has applications in other fields of study, such as bioinformatics, medicinal informatics, machine learning, operations research, dynamical frameworks, and even business and finance. SI depicts another branch of optimization strategy roused by the natural behavior of a swarm, for instance, ants, bees, and birds. Basic principles are needed to exploit such behavior, coupled with a metapartial solution search approach. Metaheuristic procedures are mainly focused on the iterative improvement of the quality of candidate solutions. A combination of basic standards inside metaheuristics enables them to handle combinatorial issues. Some of these are non-deterministic NP-hard problems [8]. Reported research and practice on these procedures supported researchers for the continuous extension and intensification of SI and metaheuristics [9]. The historical backdrops of metaheuristics and SI are particularly interwoven. Among the first and highly acknowledged metaheuristics are Tabu Search, Genetic Algorithms, and Simulated Annealing (SA). At present, SI, ACO, and PSO are viewed as the most acute SI strategies.

2. META-HEURISTIC SEARCH

2.1 ACO & PSO Algorithms

ACO and PSO are the two most important metaheuristics.

2.1.1. Ant Colony Optimization (ACO)

ACO is inspired by the activities of ants in their natural colonies. In ACO, artificial “ants” (simulation agents) find optimal solutions by traveling through a parameter space, signifying every single conceivable solution. The guideline of an ant colony algorithm is that a unique chemical trail (pheromone) is deposited on the soil amid trips and directs other ants toward an objective solution [9]. The path taken by ants has a high degree of pheromone deposition, and a high pheromone deposition rate encourages more ants taking the same path. Paths with few ants tend to have a low degree of pheromone deposition as the chemical diminishes over time. The pheromone is deposited by genuine ants to guide other ants to resources. Similar to real ants, artificial ants in the ACO determine their positions and the quality of their solutions in search spaces. The purpose was to enable more ants to better solutions after iterations [10]. Another algorithm similar to the ACO is the bee’s algorithm, which mimics the natural foraging lifestyle of honeybees.

The decision-making process of natural ants when taking their paths is shown in Figure 1. At point A, which is the decision-making point, some ants take one direction, and others may follow another direction haphazardly. When these ants have the same velocity, ants that take a shorter path will get to point B faster than those that take an alternative route. Hence, the deposition rate of pheromone on the shorter path is higher (because more ants are taking the route) than that on the longer path [11]. Subsequently, the extent of pheromone left on the short path is greater than that on the long path given that more ants pick the short side than the long side. The ACO is based on the principle used by natural ants in resolving optimization problems.

FIGURE 1. Direct ratio to the number of ants (Majida, 2013)

Activities of genuine and artificial ants are similar in some respects.

- i. The deposition of pheromone by atrails fortify the most encouraging solution segments of the construction graph.

- ii. Artificial ants devise solutions by traveling through the construction graph and selecting their route according to the pheromone trails left behind.
- iii. The diminishing rate of an artificial pheromone is faster at each iteration, mimicking the gradual disappearance of the pheromone deposited by natural ants .
- iv. The development of the ACO hinges mainly on the selection of the fitness function in consideration of the parts of the construction graph of an issue that will receive a high degree of pheromone deposition, as well as determination of the misuse of the propitious components by ants while building new solutions.

The purpose of the third stage of ACO is building ant solutions (i.e., finding the position of quality paths on the construction graph of a problem). This purpose is achieved by stochastically traversing the adjoining nodes of the graph. The ants are driven to consecutively pick the components of the solution (based on a probability rule) that will influence the extent of pheromone deposition. For every ant, the solution is constructed after all the components of the solution have been selected by the ant, that is, when the ant has completely moved through the construction graph. Once a solution has been reached by an ant, the ant can assess the full solution to be used in the pheromone updating step of the ACO in order to estimate the extent of pheromone deposition needed. The flow chat of the ACO is presented in Figure 2.

FIGURE 2. Flowchart of ACO (Christian et al., 2016)

The pheromone model is mainly used to create solutions for the considered issues (based on probabilistic rules) by gathering them from the solution component arrangement. The ACO methodology generally strives to address the optimization problems by implementing the following steps:

- i. Candidate solutions are developed by using a pheromone model and by considering a probabilistic distribution of parameters above the solution space.
- ii. The potential solutions are used for the alteration of the pheromone values in a way that is deliberated toward first-rate solutions
- iii. The pheromone guides the ants and enables them to explore the search space with the optimal solutions. An important element of the ACO algorithm is the strengthening of the solution paths, and this element is reliant on the quality of the solution. Hence, good solutions are expected to include good solution components as the learning of the inputs of the elements to the good solutions can facilitate their assembly into better solutions.

2.1.2. Pseudocode of the ACO

The pseudocode of the ACO is presented in Figure 2. For the ACO, the fitness function is normally seen as the minimization of the cost of solution components. Artificial ants specifically aim at strolling on the construction graph of the solution to select the nodes that will minimize the overall cost of a selected solution path.

```

Set the base attractiveness,  $\tau$ , and visibility,  $\eta$ , for
each edge;
for i < IterationMax do:
  for each ant do:
    choose randomly (based on equation) the next state
    to move
    into;
    add that move to the memory(tabu list) for each ant;
    repeat until each ant completed a path;
    end;
  for each ant that completed a path do:
    update attractiveness  $\tau$  for each path(edge) that the
    ant traversed;
    end;
  if (local best path better than global path)
    save local best path as global path;
  end;
end;

```

FIGURE 3. Pseudocode of the ACO

2.2 Particle Swarm Optimization (PSO)

In 1995, Eberhart and Kennedy developed a heuristic optimization procedure based on the PSO algorithm. It was inspired by the flocking pattern of birds, which was initially modeled by Craig Reynolds and then by Frank Heppner. It is a worldwide optimization algorithm intended for managing issues, and the finest solution can be indicated as a point or surface in an n-dimensional space. As a population-based method, the PSO searches the best solution by using an array of particles moving at different velocities. The positions of these particles in the solution space is adjusted progressively on the basis of their historical performance and that of their neighboring particles in the search space [12]. The PSO algorithm provides solutions for problems that can be considered a map of points in an n-dimensional space. The expression “particles” alludes to individuals basically depicted as the swarm positions within the search space.

Every particle moves within the search space at a velocity that signifies the speed of each particle in each dimension. Each particle stores the memory of its previously best-found solution, that is, its best position at any point accomplished in the search space, which is called its. The mystery of the PSO success is based on the fact that the experience of each particle is persistently communicated to a part or the entire swarm, driving the general swarm movement toward the most potential areas of the search space.

Particles move within the search space and are evaluated using some fitness benchmark at each iteration. The particles assess their current positions at each iteration on the basis of the fitness function of the problem to be optimized. Furthermore, the particles update their positions in the swarm either locally within their immediate surrounding (as in the original PSO) or within the entire swarm (as in the PSO variants) depending on their current fitness and their best-found positions.

2.2.1. Steps of the PSO algorithm

Step 1: Initialize the input parameters populace or particles with arbitrary positions and velocities on d dimensions in the search space. Every parameter is considered a particle.

Step 2: Gauge the anticipated optimization fitness task of each parameter in d factors.

Step 3: Compare the particles' fitness and particles p_{best} . If the existing value is better than p_{best} , then set p_{best} value to the existing value and p_{best} location to the current location in the d-dimensional space.

Step 4: Determine the value of the current principal fitness in the whole inputs. If the current fitness value is better than the existing global best $g(best)$, assign the current principal fitness value to the global best and relegate the recent coordinates to $g(best)$ coordinates.

Step 5: Update the speed and locus of the molecule.

$$v_{ia} = v_{ia} + c_1 * \text{rand}() * (p_{ia} - x_{ia}) + c_2 * \text{rand}() * (p_{ga} - x_{ia}) \quad (1)$$

$$x_{ia+1} = x_{ia} + v_{ia} \quad (2)$$

If the maximum iteration is achieved, the procedure is terminated, else, repeat Stage 2–5. The optimal gain parameter is calculated according to their fitness function and updated for every iteration. The flowchart of the PSO algorithm is presented in Figure 4.

FIGURE 4. Flowchart of PSO (Roohollah Kalatehjari et al; 2014)

The position of each particle is updated when the current position is better than the historically best position. The velocity of each particle is updated based on that of the global best particle (or the local best neighbour, i.e., the neighbouring particle with the best position than the particles' present one) by drawing nearer toward it. Prior to the end of every iteration of PSO, the manifestation of the swarms' global best element (or the native best element in the neighbourhood) is updated if the current location update of any particle in the entire swarm is better than that of the current global best. Figures 5 and 6. show the pseudocode of PSO in detail.

However, this methodology is preferable to the other global minimization strategies because of its ability to address the issue of local minima as a result of the large number of particles that form the particle swarm [13].

2.2.2. Real ants vs. artificial ants

Comprehending a natural sensation and outlining a nature-inspired framework are closely related but are different tasks. Natural processes are better understood by trials and observations, but building a nature-inspired framework is restricted to one's imagination and accessible technology. Even though the fundamental standards of ACO metaheuristic are motivated by the societal conduct of ant colonies, a few attributes of artificial ants do not require to be indistinguishably similar to genuine ants.

2.3 PSO vs. ACO

For the ACO, the basic parts of the algorithm are the pheromone update and the solution edifice of the ants. The three basic steps of the ACO are captured in three vital steps. The artificial ants in ACO create solutions from few sets of partial solutions. The initial stage is aimed at avoiding partial solution and reach a feasible solution in each iteration. The discovery of the solution segment depends on the search path taken by the ants. This optimal solution is considered a regular set of potential solutions that must meet a set of specific limitations.

PSO is an easy and adequately complex approach to the management of different optimization problems [13]. It is computationally cheap and reasonably fast to implement. The basic steps of the PSO are the computation of the position

```

for each particle  $i$  in  $S$  do
  for each dimension  $d$  in  $D$  do
    //initialize all particles' position and velocity
     $x_{id} = Rand(x_{min}, x_{max})$ 
     $v_{id} = Rand(-v_{max}/3, v_{max}/3)$ 
  end for
  //initialize particle's best position
   $pb_i = x_i$ 
  //update the global best position
  if  $f(p_{best}) < f(g_{best})$  then
     $g_{best} = p_{best}$ 
  end if
end for

```

FIGURE 5. PSO Initialization

```

Algorithm 2
//initialize all particles
Initialize
repeat
  for each particle  $i$  in  $S$  do
    //update the particle's best position
    if  $f(x_{id}) < f(p_{best})$  then
       $P_{best} = x_{id}$ 
    end if
    //update the global best position
    if  $f(p_{best}) < f(g_{best})$  then
       $g_{best} = p_{best}$ 
    end if
  end for
  //update particle's velocity and position
  for each particle  $i$  in  $S$  do
    for each dimension  $d$  in  $D$  do
       $v_{ia} = v_{ia} + c_1 * rand() * (p_{ia} - x_{ia}) + c_2 * rand() * (p_{ga} - x_{ia})$ 
       $x_{ia+1} = x_{ia} + v_{ia}$ 
    end for
  end for
  //advance iteration
   $it = it + 1$ 
until  $It > MAX\_ITERATIONS$ 

```

FIGURE 6. PSO (global best)

and velocity of particles. An additional velocity factor is introduced for the calculation of the positions of particles. A factor is explicitly actuated on the notion that the average velocity of the swarm is greater than the extreme velocity. Hybrid swarm optimization performs better than the original PSO and its variants. It addresses the issue of premature convergence, and the solution quality is not affected by the relocation approach. However, HSO has the problem of frequently tending to arrive at suboptimal solutions. Hence, better quality solutions can be achieved by combining HSO with other improved algorithms.

Metaheuristic can be classified into two sets: single-solution metaheuristic algorithms, where a solitary solution (and search trajectory) is considered at a time and populace metaheuristic algorithms, where multiple solutions develop simultaneously. Inside every category, constructive metaheuristics can be recognized, where a key is built by introducing new solution components at each iteration.

Metaheuristics are developed to address complex optimization problems, as such problems, cannot be handled using basic optimization techniques [14]. Metaheuristics are currently considered the most effective approaches for various real-world problems. Using metaheuristics for optimization tasks has certain advantages, such as:

Broad relevance: Implementable on any problem that can be considered a function optimization problem.

Hybridization: Can be hybridized with the traditional optimization techniques to improve performance.

Easy implementation and simple to understand.

Ability to solve bigger problems in an efficient manner. They can be easily designed and implemented and are highly adaptable.

Metaheuristics are more attractive because of (i) the intricacy of the internal issues that drive the need for better techniques, and (ii) the huge number of potential solutions that restrict the utilization of complicated algorithms.

Cooperation of some metaheuristic algorithms

FIGURE 7. Difference between different metaheuristics

In Table 1, the average optimal solution for all the proposed algorithms is shown. The columns are the tasks to be performed, and the lines are about the algorithms. Each cell is the average. The highest values of all ABC algorithms are presented in the table. Algorithm J and the proposed algorithm performed best in most cases except when the number of tasks is 100 and 500. The reason is that the algorithm is set up to work well within the space of large solutions. The highest value for the modified PSO algorithm is shown in the table.

3. CONCLUSION

A swarm is a group of a single species in which the members interact with one another and their immediate environment without a principle that controls or allows the emergence of a global intriguing behavior. As the computational modeling of swarms was projected, a correlation with the original PSO and PSO variants demonstrated that HSO outperformed existing strategies.

Table 1. Differences between different metaheuristics

Algorithm	Number of task					
	100	500	2000	4000	8000	20000
TLBO	0.046E+01	0.36E+01	3.32E+01	1.1E+01	4.1E+01	8.9E+01
DOLTLBO	0.045E+01	0.37E+01	3.36E+01	1.1E+01	3.9E+01	8.1E+01
PSOTLBO	0.043E+01	0.37E+01	3.28E+01	1.1E+01	4.1E+01	8.9E+01
JAYA	0.048E+01	0.38E+01	3.38E+01	1.1E+01	3.9E+01	8.5E+01
SGA	0.068E+01	0.46E+01	3.95E+01	1.2E+01	4.3E+01	8.5E+01
ABC	0.021E+01	0.24E+01	2.64E+01	8.0E+00	3.6E+01	7.9E+01
PSO	0.143E+01	0.67E+01	5.05E+01	1.5E+01	5.1E+01	1.01E+02
Modified PSO	0.045E+01	0.37E+01	3.27E+01	1.0E+01	3.8E+01	8.0E+01

REFERENCES

- [1] R. A. Hasan, M. A. Mohammed, Z. H. Salih, M. A. B. Ameen, N. Tăpuș, and M. N. Mohammed, "HSO: A Hybrid Swarm Optimization Algorithm for Reducing Energy Consumption in the Cloudlets," *Telkomnika*, vol. 16, no. 5, pp. 2144–2154, 2018.
- [2] R. A. Hasan and M. N. Mohammed, "A krill herd behaviour inspired load balancing of tasks in cloud computing," *Studies in Informatics and Control*, vol. 26, pp. 413–424, 2017.
- [3] M. A. Mohammed, Z. H. Salih, N. Tăpuș, and R. A. K. Hasan, "Security and accountability for sharing the data stored in the cloud," *2016 15th RoEduNet Conference: Networking in Education and Research*, pp. 1–5, 2016.
- [4] M. A. Mohammed and R. A. Hasan, "Particle swarm optimization for facility layout problems FLP-A comprehensive study," *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing*, pp. 93–99, 2017.
- [5] R. A. Hasan, M. A. Mohammed, N. Tăpuș, and O. A. Hammood, "A comprehensive study: Ant Colony Optimization (ACO) for facility layout problem," *2017 16th RoEduNet Conference: Networking in Education and Research*, pp. 1–8, 2017.
- [6] M. A. Mohammed, R. A. Hasan, M. A. Ahmed, N. Tapus, M. A. Shanan, and M. K. Khaleel, "A Focal load balancer based algorithm for task assignment in cloud environment," *2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–4, 2018.
- [7] M. A. Ahmed, R. A. Hasan, A. H. Ali, and M. A. Mohammed, "The classification of the modern arabic poetry using machine learning," *Telkomnika*, vol. 17, pp. 2667–2674, 2019.
- [8] R. A. Hasan, I. Alhayali, A. Royida, N. D. Zaki, and A. H. Ali, "An adaptive clustering and classification algorithm for Twitter data streaming in Apache Spark," *Telkomnika*, vol. 17, 2019.
- [9] R. A. Hasan, M. N. Mohammed, M. A. B. Ameen, and E. T. Khalaf, "Dynamic Load Balancing Model Based on Server Status (DLBS) for Green Computing," *Advanced Science Letters*, vol. 24, pp. 7777–7782, 2018.
- [10] E. T. Khalaf, M. N. Mohammad, K. Moorthy, and R. A. Hasan, "Biometric Template Protection based on Hill Cipher Algorithm with Two Invertible keys," *Advanced Science Letters*, vol. 24, pp. 7643–7649, 2018.
- [11] M. A. Mohammed, A. A. Kamil, R. A. Hasan, and N. Tapus, "An Effective Context Sensitive Offloading System for Mobile Cloud Environments using Support Value-based Classification," *Scalable Computing: Practice and Experience*, vol. 20, pp. 687–698, 2019.
- [12] M. A. Mohammed, I. A. Mohammed, R. A. Hasan, N. Tăpuș, A. H. Ali, and O. A. Hammood, "Green Energy Sources: Issues and Challenges," *2019 18th RoEduNet Conference: Networking in Education and Research*, pp. 1–8, 2019.
- [13] O. A. Hammood, M. N. M. Kahar, W. A. Hammood, R. A. Hasan, M. A. Mohammed, and A. A. Yoob, "An effective transmit packet coding with trust-based relay nodes in VANETs," *Bulletin of Electrical Engineering and Informatics*, vol. 9, pp. 685–697, 2020.
- [14] H. D. K. Al-Janabi, H. D. K., and R. A. H. Al-Bukamrh, "Impact of Light Pulses Generator in Communication System Application by Utilizing Gaussian Optical Pulse," *2019 22nd International Conference on C.*