Науковий вісник НЛТУ України
Scientific Bulletin of UNFU
https://doi.org/10.36930/40340513
Article received 06.05.2024
Article accepted 23.05.2024
UDC 004.[021+8]
Correspondence author: I. O. Rabiichuk rabiychuk.igor@gmail.com
ISSN 1994-7836 (print)
ISSN 2519-2477 (online)

I. O. Rabiichuk, A. V. Fechan
Lviv Polytechnic National University, Lviv, Ukraine

THE MAIN CHALLENGES OF ADAPTABILITY OF SWARM INTELLIGENCE ALGORITHMS

Analyzed three swarm intelligence algorithms, namely Ant Colony Optimization (ACO), Bee Colony Optimization (BCO), Particle Swarm Optimization (PSO) and the adaptability of these algorithms to a dynamic environment. Firstly, the ACO algorithm was analyzed, the behavior of ants in nature, the purpose of the algorithm, and its shortcomings in a dynamic environment. Then the existing modifications of this algorithm to changing environments were investigated, namely ACO with dynamic pheromone updating (AACO), ACO with adaptive memory (ACO-AP), ACO with multi-agent system (MAS-ACO), ACO with machine learning algorithms (MLACO). The advantages and disadvantages of these modifications are also discussed in detail. The software tools that implement the functionality of this algorithm, such as AntTweakBar, AntOpt, EasyAnt have been mentioned. These software tools provide an opportunity to develop new modifications of the ACO algorithms and to study existing ones. Furthermore, the capabilities of the BCO algorithm were clarified and the behavior and parameters of this algorithm were described, its pros and cons in a dynamic environment were investigated. The following BCO modifications were considered: Group Bee Algorithm (GBA), Artificial Bee Colony (ABC), and open source software: PySwarms, PyABC. The third part of the article investigates the work of the PSO algorithm, its advantages and disadvantages of adaptation to dynamic environments. Dynamic Particle Swarm Optimization with Permutation (DPSO-P), Dynamic Multi-swarm Particle Swarm Optimization Based on Elite Learning (DMS-P50-EL) are considered as modifications of PSO to adapt to dynamic environments. The libraries for work such as SciPy, DEAP, PyGAD, Particleswarm, JSwarm (has a wide API and well-written documentation), Dlib have been mentioned. Finally, a comparative table with the most important properties (resistance to environmental changes, complexity of implementation, the possibility of using for a UAV swarm, etc.) for all three algorithms was created, a brief description of similar articles comparing algorithms of swarm intelligence was also made, and the conclusions of the study were drawn.

Keywords: Ant Colony Optimization (ACO); Bee Colony Optimization (BCO); Particle Swarm Optimization (PSO); Adaptation to dynamic environments.

Introduction / Вступ

Swarm theory algorithms are an integral part of the functioning of society; they are used in modern technologies (distributed computing and network optimization), the military sphere (cyber security and unmanned aerial vehicles), forecasting economic situations (modeling market trends and detecting changes in economic indicators), and predicting climate change. Classical swarm theory algorithms have problems with adaptability in dynamic environments, as they rather designed for use in static environments. This problem is studied by many scientists such as Yuhui Shi who researches PSO and its adaptive variations, as well as Russell Eberhart who is a co-author of PSO and works on the next versions of the algorithm to improve the interaction with the dynamic environment, Milorad Jelić is engaged in improving the adaptability of the BCO algorithm, and Carlos Blum works on the algorithm ACO.

In this subject area, there are many studies on improving the adaptability of swarm algorithms, but these improvements usually require a large number of hardware resources, which makes it impossible and expensive to use these algorithms in a swarm of small unmanned aerial vehicles, such as a swarm of FPV drones. The relevance of the research is to adapt and hybridize current modifications of swarm intelligence algorithms to improve adaptability in dynamic environments using a small number of hardware resources.

Object of research – the processes and mechanisms of adaptation of swarm intelligence algorithms in dynamic environments and their application in automated devices, in particular, mechanisms of self-organization, interaction, and optimization of swarm intelligence.

Subject of research – the methods and means of adaptation of swarm algorithms ACO, BCO, and PSO, including modification strategies, for their effective use in automated devices under dynamically changing conditions.

The purpose of the work – analyze three swarm intelligence algorithms such as ACO, BCO, PSO, and their adaptations and modifications to dynamic environments to apply them to control a swarm of unmanned vehicles.

To achieve this *purpose*, the following main *research*

Інформація про авторів:
Рабійчук Ігор Олександрович, магістр, аспірант, асистент, кафедра програмного забезпечення.
Email: ihor.o.rabiichuk@lpnu.ua; https://orcid.org/0009-0002-0881-0406
Фечан Андрій Васильович, д-р техн. наук, професор, кафедра програмного забезпечення.
Email: andrii.v.fechan@lpnu.ua; https://orcid.org/0000-0001-9970-5497

Цитування за ДСТУ: Рабійчук І. О., Фечан А. В. Основні виклики адаптаційності алгоритмів ройового інтелекту. Науковий вісник НЛТУ України. 2024, т. 34, № 5. С. 97–103.
Citation APA: Rabiichuk, I. O., & Fechan, A. V. (2024). The main challenges of adaptability of swarm intelligence algorithms. *Scientific Bulletin of UNFU*, 34(5), 97–103. https://doi.org/10.36930/40340513

objectives are identified:

1. Study the basic algorithms of ACO, BCO, and PSO, which will provide an understanding of the basic capabilities of current algorithms and their purpose.
2. Investigate the possibility of adaptation to dynamic environments, which will allow investigating what current modifications of these algorithms exist at the moment.
3. Investigate existing modifications of these algorithms for interaction with a dynamic environment, which will provide the ability to consider the existing advantages and disadvantages and unexplored problems in the existing modifications.
4. Research software tools that implement these algorithms, which will help to practically investigate the problems of adaptability of swarm algorithms in dynamic environments.

Analysis of recent research and publications. In recent years, swarm intelligence algorithms have gained enormous popularity in the world, as they are used in forecasting economic factors, stock exchanges, the value of cryptocurrency, forecasting weather conditions, tasks of a traveling salesman, and tourist forecasting [7, 10, 11]. They became most relevant in Ukraine during the war. Many researchers are trying to apply swarm algorithms in unmanned aerial vehicles [11, 15].

The article *Modeling, Guidance, and Robust Cooperative Control of Two Quadrotors Carrying a "Y"-Shaped-Cable-Suspended Payload* examines the interaction of a swarm of drones for cargo delivery. It also develops a mathematical model for controlling the movement of drones carrying a common load, so that yaw angles, speed and pitch are synchronized to prevent loss of cargo. Another development is a new "Y"-shaped connection for dropping the load. This connection enables unhooking the load synchronously due to the controller and the cut-off device on the leg of this connection. This is better than disconnecting devices on every drone [17].

The work *Air-Ground Collaborative Multi-Target Detection Task Assignment and Path Planning Optimization* describes the interaction of several different types of drones, such as unmanned aerial vehicles and ground drones. This interaction helps in various fields. For instance in agriculture, where an unmanned aerial vehicle controls the operation of ground-based automated agricultural machinery, or in intelligence, where a UAV performs the work of a scout, and ground drones perform various tasks such as mining/demining, and shooting at the enemy. The work also describes the rules of interaction of different types of drones and describes a practical experiment [9].

The scientific work of Ukrainian authors from the Kharkiv Aviation Institute is devoted to the use of swarm intelligence algorithms for the design of control of a group of UAVs. The advantages of using UAVs for such tasks as monitoring, video recording, patrolling, search and rescue operations are considered. The article analyzes the algorithms, such as ACO (ant algorithm), BA (bee swarm algorithm), and FA (firefly algorithm). The authors also propose a hybrid of algorithms for effective management [6].

ACO, BCO, and PSO algorithms are also used to find the optimal path and can be used in the construction of an Internet network, cargo delivery [7, 8, 10], in the construction of flight corridors.

The main problem with these algorithms is that they are designed for static environments and require development and modification to adapt to dynamic environments so that the results of these algorithms are effective, so developers create new solutions [5, 13, 16, 18].

For example, the GBA algorithm (group bee algorithm) is an improved version of the BA algorithm (bee algorithm). Its task is to find the optimal solution in a multidimensional space. The BA algorithm is inefficient due to a large number of parameters and computational complexity, while GBA uses groups of bees to explore different-sized decision regions, which results in a reduction in the number of parameters. The efficiency was proven through 12 test functions [11].

Research results and their discussion / Результати дослідження та їх обговорення

***Ant Colony Optimization* (*ACO*).** Firstly, Ant Colony Optimization (ACO) will be analyzed. This algorithm is designed for optimization tasks, in particular, the tasks of finding the shortest paths [10], making schedules, and planning routing. This algorithm was developed by Marco Dorigo and his colleagues in the 1990 s and was written in Marco Dorigo and Thomas Stützle's book "*Ant Colony Optimization*", which was published in 2004. This algorithm was based on the behavior of a colony of ants looking for food. One ant (agent) in search of food lays down pheromones, which attract other ants. The more food and the shorter the path, the more pheromones the ant lays down and this gives a great chance that other ants will also follow this path. In a situation when ants have two paths to food the following is observed: a shorter and a longer one, in the beginning, the ants will explore both paths and follow both, but in the process, more pheromones will be laid on the shorter path because traveling on this path is faster. As a result, over more passes, more pheromones will be laid and this will attract other ants. Eventually, they will all shift to a shorter path with a large quality of pheromones, and thus the optimal solution will be found. The problem with this algorithm is that if another shortest path is added to the previous two, the ants will not use it even if there are more pheromones on the existing path.

In practice, this algorithm is used in courier services to find shorter and better delivery routes, the so-called solution to the traveling salesman problem [10]. This algorithm is used by such delivery services [8] as DHL, UPS, and FedEx. Another application is the use of the ant algorithm in the optimization of traffic lights in Singapore, this makes it possible to optimize the movement of vehicles in traffic jams. Similar solutions are found in London, Paris, Barcelona, Milan, and Los Angeles. Furthermore, this algorithm is practically used by telecommunications companies to optimize network routes. It allows them to build a more optimal and high-quality network for data transmission. Modifications of the ACO algorithm for adaptation in dynamic environments:

***ACO with a dynamic update of pheromones* (*AACO*)**: this modification of the algorithm is based on the dynamic evaporation of pheromones, which prompts artificial ants (agents) to search for new paths, i.e. pheromones evaporate at each iteration. It is also possible to use a global route table where ants themselves can decide which route to choose depending on prospects and regardless of whether there is a maximum number of pheromones on any route.

Pros: Improved response to changes; Adaptation to changes is possible; Ability to find better solutions in dynamic environments.

Cons: Requires more computing resources; Remaining problem of getting stuck in local optima; Difficulty of implementation.

ACO with adaptive memory (***ACO-AP***): This is a modification of ACO that allows the use of an adaptive memory model. There are two types of memory: short-term and long-term. The short-term is used to store information about recently visited vertices in the graph, and the long-term is about the best paths found. This modification allows for updating the behavior of ants during the search, which allows finding the best solutions in a dynamic environment.

Pros: More advanced solutions. Fast achievement of results. Improved ability to adapt to complex tasks.

Cons: A large number of parameters. Complex interpretation of results.

ACO with multi-agent system (***MAS-ACO***)**:** This algorithm can use agents/ants that communicate with each other based on pheromones or based on knowledge about the task, or hybrid variants of algorithms where both types of agents are used. Agents with pheromones provide a higher speed of solving problems, with knowledge about tasks – higher quality, as they can evaluate the solutions found. Hybrid algorithms are used where better quality and speed are needed. This modification differs from the usual ACO due to the presence of a clear structure, communication with each other, and the use of both local and global information [18].

Pros: Quality. Better coordination and cooperation. Flexibility and adaptability.

Cons: A large amount of data is required for training. More time is needed to find a solution.

ACO with machine learning algorithms (***MLACO***)**:** this is a version that combines ACO and machine learning, where the ACO algorithm is used to find solutions, and ML is used to set the direction of the search. The peculiarity of this algorithm is that each path/vertex of the graph is initialized with the same number of pheromones and ants visiting these vertices take away a certain quantity of pheromones, after each iteration, at the end of the iteration, the pheromones are updated and those vertices, which are in the best solutions receive more pheromone. The ML model is trained on the data over time and this makes it possible to improve its predictions, which affects the speed and quality of solutions found by agents.

Pros: Increased productivity. Ability to solve problems with many parameters.

Cons: Sophisticated hardware. Time to train the model.

Software tools that use the ACO algorithm:

- AntOpt – software developed by OptumInsight, which uses a modified ACO algorithm called ACS (use of pheromone update algorithm, use of local search, use of so-called *elite* agent/ant) to find optimal solutions in routes and is used in logistics and telecommunications.
- EasyAnt – a platform developed by Swarm Intelligence Lab for optimization tasks. The product is open-source and intended for scientific research purposes to develop new and improved versions of ACO.
- AntTweakBar – an open-source library developed by NVIDIA, also for research purposes to find new modifications of the ACO algorithm.

Bee Colony Optimization (***BCO***)**.** This algorithm was first proposed in 2001 by Dr. Dusan Teodorović. The algorithm simulates the behavior of a swarm of honeybees searching for food sources. That is, there is a hive with a certain number of bees, where they are divided based on their roles (for example, scouts, workers, and observers) [2] and each bee has its own role. Scouts fly over the fields in search of the place with the largest number of flowers. Having found a target, they return to the hive where, using a dance (a form of communication), they inform the swarm about the found target. This dance is also an attempt to agitate the swarm to attract more worker bees to the found area.

After a worker bee has flown to a given area, the bee has the choice of returning to the same location or following another scout bee.

The BCO algorithm uses the modulation of artificial bee agents that try to find the maximum value of the function on the solution area. That is when initializing the algorithm, a certain number of agents (bees) and their motion vectors are randomly assigned so that they can explore more possible decision areas. Each iteration consists of two phases: going forward for search and returning for information exchange. The output from the algorithm can be specified by such parameters as the maximum number of iterations, the maximum processing time, and the value of the global solution is greater than or equal to the given one.

To improve adaptability, the best global solution is updated and remembered after each iteration, one of the methods to improve this algorithm is to run the algorithm with the same number of parameters on different processors where they exchange information about the best global solution [2] after each iteration, this leads to an improvement in the quality of decisions in the final result.

BCO is used to solve combinatorial problems of any complexity, as well as in resource allocation, logistics, energy, and robotics.

This algorithm is used by companies such as Google (to predict the route in their self-driving cars), Deutsche Telekom (to place mobile towers), and IBM (to balance loads in their cloud environments).

The problems of standard BCO in dynamic environments arise from initially set static parameters such as the number of bees and the exit condition (finite number of operations, limited processor time) since the quality of the solution depends on the setting of these parameters. If the environment changes dynamically, this will lead to the fact that the solution will be not found or the solution will be of poor quality.

In comparison, the ACO algorithm and the BCO algorithm are very similar in their structure. However, while the ant algorithm is looking for the shortest path, the bee algorithm is aiming for the best area of solutions and is more adapted to environmental changes [6].

In addition to the basic BCO algorithm, there are many other different modifications inspired by bee swarm behavior, such as GBA (group bee algorithm), and ABC (artificial bee algorithm) [15].

GBA (***Group Bee Algorithm***) is a group bee algorithm, based on the group/corporate work of bees, its specificities are that the algorithm can be executed several times since scouts work in groups to search for goals, not individually, and also use joint intelligence to make decisions [11].

Pros: Adaptation to changes. Interaction among groups. Improved distribution of tasks.

Cons: Setting a large number of parameters. Difficulty in implementing communication mechanisms.

ABC (***Artificial Bee Colony***) is an artificial bee colony, which uses such agents as workers, observers, and scouts. Scouts are responsible for researching a new solution area.

Ordinary workers become scouts after it is impossible to improve a local solution after a certain number of attempts. Observers watch workers and choose the most promising areas. Then they explore their suburbs, while workers explore the solution area assigned to them [5].

The algorithm works until stopping criteria similar to those used in BCO are reached. The difference between ABC and BCO is that ABC focuses on individual bees working, observing, and searching for new food sources, while BCO uses the collective behavior of bees to find solutions.

Pros: Ease of implementation. Flexibility to change.

Cons: Slow approach to the optimal solution. Insufficient search depth.

To improve BCO in dynamic environments, crossing with other algorithms of swarm theory, artificial intelligence for better decision-making, dynamic updating of the number of individuals in a bee swarm, reduction of bee movement inertia, use of adaptive strategies, forecasting, use of parallel groups of agents, and fast response to a dynamically changing events can be used.

Software tools that use the BCO algorithm:

- PySwarms is an open-source Python library for visualizing the optimization process and solving single/multidimensional problems based on bee swarms. This library contains various implementations of BCO modifications. The library is most often used for resource allocation and planning tasks, etc.
- PyABC is another open-source Python library that implements ABC (Artificial Bee Colony), which is a modified BCO. It is convenient to configure parameters such as the intensity factor, the size of the swarm, and the number of iterations.

Particle Swarm Optimization (PSO). The particle swarm algorithm is based on the behavior of flocks of birds and schools of fish and was first developed by James Kennedy and Russell C. Eberhart in 1995.

This algorithm began as a simulation of a simplified social environment to simulate a collision-resistant flock of birds but evolved into an abstract algorithm [7] that is used in various fields of society. It is used for planning train timetables, university timetables, the location of goods in the warehouse, planning the expected number of tourists in different places, hotel occupancy, predicting energy network loads, planning the location of taxis in the city to reduce the time of getting to the client, location of police cars to monitor event; and in many other areas.

The idea of the algorithm is to form a certain number of agents (particles) in the field of research, where each particle is looking for the best solution. The best values are stored in the best global solution and the particles move towards the best global solution, but in addition, to avoid local minima, the particles move towards the nearest neighbor that has a better position. Briefly, PSO can be described as follows:

- Initialization of a certain number of agents in the solution area
- The positions of the agents are randomly assigned, but each agent has its position.
- Agents move towards the best solutions they find.
- Agents also move towards the nearest agents with better solutions.
- The last two points are repeated until the best global solution is found.

The exit condition can be the maximum number of iterations, the maximum convergence where the particles either do not move at all or the changes are so insignificant that it does not give any result, other exit conditions are also possible, such as the maximum operating time of the algorithm or when the algorithm reaches the desired level of optimization.

The parameters of the particle swarm algorithm are:

- The number of agents in the solution area, this parameter can be different depending on the needs, a large number of agents leads to the use of a large number of resources, as well as a longer optimization time, but the result is finding better solutions.
- Weights of the global solution – this parameter is denoted as C1 and it affects how much the particle will focus on the best global solution.
- Weights of the local solution – this parameter is denoted as C2 and affects how much the particle will focus on the best local solution.
- The speed of particle movement – it is set through a function whose parameters are the position of the particle in the best local and global solution, and the parameters are the coefficients C1 and C2, the speed is limited to ensure the stability of the algorithm.

The consideration of the classical PSO in dynamic environments will not be effective. For example, if the best solution appeared in a random position during the operation of the algorithm, there is a small probability that it will be found since the agents are grouped and head to the current best global solution. There are many modifications of this algorithm and one of them is DPSO-P [13].

DPSO-P (Dynamic Particle Swarm Optimization with Permutation) is a modification of PSO that provides the ability to track new best solutions if they appear when the dynamic environment changes. This is achieved by using three methods A, B, C, and their hybridization.

- A – when the environment changes, it is a scattering of a certain percentage (a changing parameter) of agents with the worst decisions on random positions in the solution area. It is usually optimal to use 10 % of agents.
- B – zeroing of the best memory of all particles for forced search of new best solutions.
- C – random perturbation, i.e. changing the position of the particles by exchanging the coordinates of the particles. The perturbation occurs randomly during each iteration.

The authors of this algorithm tested and compared the effectiveness of this DPSO-P algorithm with the basic PSO. They found that DPSO-P is more effective in most cases [13], except for the case when PSO uses a large number of agents, which requires a lot of computing power and time but leads to better solutions.

Pros: Ability to explore dynamic changes in the environment. Permutations of agents.

Cons: Lower quality, with incorrectly selected parameters. Complex implementation.

(DMS-PSO-EL) Dynamic Multi-swarm Particle Swarm Optimization Based on Elite Learning – this version of the algorithm is based on dividing the population into several sub-swarms, there are two types of swarms to balance between exploitation and research opportunities. Swarms work in parallel to explore the solution domain and share information about the best solutions found. The algorithm can change the number of agents in the swarm and the number of swarms depending on the optimization task, as well as train so-called elite agents, which [16].

Pros: Acceleration of solution area examination. High-quality solutions.

Cons: Extremely resource-intensive. Algorithm complexity.

Software tools that use the PSO algorithm:

- SciPy, DEAP, and PyGAD are open-source Python libraries that contain various PSO implementations for finding the best solutions in the solution area.
- Particleswarm is a MATLAB function used to optimize solutions, the advantage of this function is the solution of problems for multidimensional spaces. The parameters of this function are objectiveFunction (this parameter accepts the function to be optimized), nvars (specifies the number of parameters to be optimized), lb and ub (upper and lower bounds of the variables). The results are placed in the next variable: x (values of the optimized parameters), fval (best value of the function), exitflag (exit condition), output (detailed information about the optimization process). Additional parameters in this function are the inertia weight and the number of iterations.
- JSwarm is an open-source Java library that has a powerful API and well-written documentation. This library contains many examples of work, which makes it easy to learn or get started. The parameters that can be customized are the weights for local and global solutions, the number of agents in the population, and the maximum and minimum values of the best solution.
- Dlib is a C++ library that has a PSO implementation.

The properties of the ant algorithm, the bee algorithm, and particle swarm optimization have been demonstrated in a table (Table 1) for comparison and better understanding.

Table. Comparison of ACO, BCO, PSO algorithms / Порівняння алгоритмів ACO, BCO, PSO

	ACO	BCO	PSO
Adaptable to dynamic environments	No	No	No
The existence of algorithm modifications to dynamic environments	Yes (AACO, ACO-AP, MAS-ACO, MLACO, …)	Yes (GBA, ABC, …)	Yes (DPSO-P, DMS-PSO-EL, …)
Resistance to environmental changes	Low	Average (using different types of agents makes it possible to track changes)	Low
Use of resources	Average efficiency	High efficiency	Average efficiency
Possibility of parallel execution to find solutions	No (the basic algorithm does not support the global best solution)	Yes	Yes
Purpose	Finding optimal ways or solutions	Finding the optimal solution area	Finding optimal solutions
The possibility to be used for a UAV swarm	Finding the optimal route	Exploration and research of territories	Swarm formation, movement coordination
Software solutions	AntTweakBar, AntOpt, EasyAnt	PySwarms, PyABC	SciPy, DEAP, PyGAD, Particleswarm, JSwarm, Dlib
Complexity of implementation	High	High	Average

As a result of the analysis of the three algorithms ACO, BCO and PSO, it was found that they are almost not suitable for dynamic environments, and if they are adapted, they use a significant amount of software resources and require appropriate modification to implement a swarm of drones.

Discussion of research results. Consider similar articles where current algorithms are compared and hybridized. In the article about the ACO – PSO hybrid algorithm [12], two algorithms are compared separately, and their hybrid algorithm, also for initialisation, a genetic algorithm (GA) is used. Based on the comparison, it is seen that the basic PSO is effective by 97.65 according to the Butt function and by 98.71 according to the Rosenbrock function 98.71. However, the hybrid algorithm provides improved efficiency results where the results are 98.83 and 99.89 and it provides a better ability to adapt to dynamic environments.

The following article analyzes the particle swarm algorithm (PSO), the bat algorithm (BA), and the gray wolf algorithm (GWO), where they calculate the quality of algorithms based on the interaction of drones and use a distributed version of each algorithm for the calculation, where drones can be developed at once in several different directions. The conclusion based on their comparison is that the basic PSO is the best with a small number of agents [4] and a short execution time, and the GWO is the best for a large number of agents where time is not important.

In the work of the authors Pratama and Suyanto, the Particle Swarm Optimization (PSO), Firefly Algorithm (FA), and Bat Algorithm (BA) algorithms are described, using the comparison by solving discrete optimization problems, in particular the traveling salesman problem (TSP) [14], it is therefore the best algorithm for dynamic environments among the above.

The article "Comparison of Swarm and Graph Algorithms for Solving Traveling Salesman" Problems is based on a comparison of swarm algorithms and graph algorithms in the traveling salesman problem (TSP) [3]. The authors use 22 different TSP problems from the TSPLIB95 library to compare the efficiency and the result of the comparison was that graph algorithms prevail over swarm theory algorithms such as ACO and PSO.

The following work examines swarm intelligence algorithms such as Particle Swarm Optimization (PSO) and Cuckoo Search (CS) and compares them using the following functions: Ackley, Rastrigin, and Rosenbrock. Based on the results the authors concluded that with a small number of agents, both algorithms work in the same way but when using more than 50 agents, CS will be significantly more efficient [1], which is important for dynamic environments.

So, based on the results of the work performed, it is possible to formulate the following scientific novelty and practical significance of the research results.

Scientific novelty of the obtained research results – for the first time, a comparative analysis of swarm intelligence algorithms such as Ant Colony Optimization (ACO), Bee Colony Optimization (BCO), Particle Swarm Optimization (PSO) was conducted, which made it possible to choose the most suitable ones for group adaptation.

Practical significance of the research results – an understanding of how and according to which parameters it is necessary to select algorithms for use in a group of unmanned aerial vehicles, which will make it possible to develop improvements and hybridization of swarm intelligence algorithms for use in a swarm of unmanned aerial vehicles using a minimum amount of resources.

Conclusions / Висновки

This paper investigated the suitability of swarm intelligence algorithms such as ACO, BCO and PSO for controlling swarms of unmanned aerial vehicles (UAVs) in dynamic environments in key areas such as the evaluation of basic algorithms, adaptation strategies, software tools, and comparative analysis. Such swarm intelligence algorithms as ACO, BCO, PSO, their possible adaptations and modifications to the dynamic environment, suitable for the management of a swarm of unmanned aerial vehicles, are analyzed. Based on the results of the research the following main conclusions can be drawn.

1. Three algorithms Ant Colony Optimization (ACO), Bee Colony Optimization (BCO), and Particle Swarm Optimization (PSO) have been studied. It was determined that they do not adapt well to dynamic environments, use a large number of resources for adaptation, and need modified versions to implement the swarm in unmanned aerial vehicles.
2. Algorithms ACO, BCO, and PSO were considered as well as their basic implementations and behavior on which they are based, and problems that may arise in a dynamic environment were analyzed. Furthermore, modifications of these algorithms for dynamic environments such as ACO (AACO, MAS-ACO, MLACO), BCO (GBA, ABC), PSO (DPSO-P, DMS-PSO-EL) are considered, as well as what methods they use for adaptation to changes, their advantages and disadvantages and software tools.
3. To conclude, the basic algorithms of ACO, BCO, and PSO are not adapted to dynamic environments, since they were developed for static environments. There are many improvements in the modifications, such as flexible interaction with dynamic environments, fast response to changes, and the use of a large number of different types of agents. However, there are also disadvantages, such as the complexity of implementation and the use of a large amount of computing resources, which leads to complexity and high cost of use in unmanned aerial vehicles.
4. Software tools that implement these algorithms were also considered. Most of them are open source and contain the implementation of these algorithms, but there are also those intended for training and development of new modifications of these algorithms. The research shows that the largest number of software tools exist for the ACO algorithm.
5. For a better understanding and the possibility of selecting an algorithm for a specific task, a comparison table of ACO, PSO, and BCO algorithms was created according to various properties (use of resources, resistance to environmental changes, complexity of implementation, etc.). Similar articles were also analyzed for a better understanding of the properties by which swarm intelligence algorithms were compared by other authors.

The next step of research is the development and improvement or hybridization of swarm intelligence algorithms for their use in swarms of unmanned aerial vehicles. It will be particularly focused on the use and construction of a swarm of FPV drones.

References

1. Bashath, S., & Ismail, A. R. (2018). Comparison of Swarm Intelligence Algorithms for High Dimensional Optimization Problems. *Indonesian Journal of Electrical Engineering and Computer Science*, 11(1), 300–307. https://doi.org/10.11591/ijeecs.v11.i1
2. Davidović, T., Teodorović, D., & Šelmić, M. (2015). Bee Colony Optimization. Part I: The algorithm overview. *Yugoslav Journal of Operations Research*, 25(1), 33–56. https://doi.org/10.2298/YJOR131011017D
3. Eggenschwiler, S., Spahic-Bogdanovic, M., Hanne, T., & Dornberger, R. (2020). Comparison of Swarm and Graph Algorithms for Solving Travelling Salesman Problem. In: *7th International Conference on Soft Computing & Machine Intelligence* (ISCMI), 1–7. Stockholm, Sweden. https://doi.org/10.1109/ISCMI51676.2020.9311558
4. Fan, J., Hu, M., Chu, X., & Yang, D. (2017). A comparison analysis of swarm intelligence algorithms for robot swarm learning. In: *2017 Winter Simulation Conference* (WSC), 3042–3053. Las Vegas, NV, USA. https://doi.org/10.1109/WSC.2017.8248025
5. Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3), 459–471. https://doi.org/10.1007/s10898-007-9149-x
6. Karatanov, O. V., Ustymenko, O. V., Yena, M. V., Bova, Y. A., & Kalashnikova, V. I. (2021). Application of swarm intelligence algorithms in the design of control systems for groups of unmanned aerial vehicles. *Young Scientist*, 24(10), 98–103. https://doi.org/10.32839/2304-5809/2021-10-98-24
7. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In: *Proceedings of ICNN'95 – International Conference on Neural Networks*, 4, 1942–1948. Perth, WA, Australia. https://doi.org/10.1109/ICNN.1995.488968
8. Liu, T., Yin, Y., & Yang, X. (2020). Research on logistics distribution routes optimization based on ACO. In: *Proceedings of the 5th International Conference on Information Science, Computer Technology and Transportation* (ISCTT), 641–644. Shenyang, China. https://doi.org/10.1109/ISCTT51595.2020.00122
9. Ma, T., Lu, P., Deng, F., & Geng, K. (2024). Air – Ground Collaborative Multi-Target Detection Task Assignment and Path Planning Optimization. *Drones*, 8(3), 110. https://doi.org/10.3390/drones8030110
10. Muruganantham, V., Rehan, M. Y. E. S., Srinivasan, R., Kavitha, M., & Kavitha, R. (2023). Traveling salesman problem with ant colony optimization. In: *Proceedings of the 2nd International Conference on Edge Computing and Applications* (ICECAA), 481–485. Namakkal, India. https://doi.org/10.1109/ICECAA58104.2023.10212262
11. Nasrinpour, H. R., Bavani, A. M., & Teshnehlab, M. (2017). Grouped Bees Algorithm: A grouped version of the bees algorithm. *Computers*, 6(1), 5–9. https://doi.org/10.3390/computers6010005
12. Pal, D., Verma, P., Gautam, D., & Indait, P. (2016). Improved optimization technique using hybrid ACO-PSO. In: 2016 *2nd International Conference on Next Generation Computing Technologies* (*NGCT*), 277–282. Dehradun, India. https://doi.org/10.1109/NGCT.2016.7877428
13. Pierobom, J. L., Delgado, M. R., & Kaestner, C. A. A. (2012). Particle swarm optimization applied to the dynamic allocation problem. In: *2012 Brazilian Symposium on Neural Networks*, 184–189. Curitiba, Brazil. https://doi.org/10.1109/SBRN.2012.35
14. Pratama, D. H., & Suyanto, S. (2020). Comparison of PSO, FA, and BA for Discrete Optimization Problems. In: *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems* (ISRITI), 17–20. Yogyakarta, Indonesia. https://doi.org/10.1109/ISRITI51436.2020.9315371
15. Shimko, S., Tkachuk, R., & Shulyka, Y. (2022). Investigation of swarm intelligence methods for the management of unmanned aerial vehicles. In: *Proceedings of the International Scientific and*

Technical Conference: Automation, Computer-Integrated Technologies, and Energy Efficiency Issues in Industry and Agriculture, 1–3. Kropyvnytskyi, Ukraine. https://doi.org/10.1109/NaBIC.2011.6089619

16. Tang, Y., Wei, B., Xia, X., & Gui, L. (2019). Dynamic multi-swarm particle swarm optimization based on elite learning. In: *2019 IEEE Symposium Series on Computational Intelligence*, 2311–2318. Xiamen, China. https://doi.org/10.1109/SSCI44817.2019.9002680
17. Wang, E., Sun, J., Liang, Y., Zhou, B., Jiang, F., & Zhu, Y. (2024). Modeling, Guidance, and Robust Cooperative Control of Two Quadrotors Carrying a "Y"-Shaped-Cable-Suspended Payload. *Drones*, 8(3), 103. https://doi.org/10.3390/drones8030103
18. Yang, L., Sun, X., Shen, Q., Zhang, B., & Chi, T. (2014). Combining multi-agent and ant colony optimization for endmember extraction. In: *Proceedings of the 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing* (WHISPERS), 1–4. Lausanne, Switzerland. https://doi.org/10.1109/WHISPERS.2014.8077592

І. О. Рабійчук, А. В. Фечан

Національний університет "Львівська політехніка", м. Львів, Україна

ОСНОВНІ ВИКЛИКИ АДАПТАЦІЙНОСТІ АЛГОРИТМІВ РОЙОВОГО ІНТЕЛЕКТУ

Проаналізовано три алгоритми ройового інтелекту: Ant Colony Optimization (ACO), Bee Colony Optimization (BCO), Particle Swarm Optimization (PSO) і їх адаптивність до динамічного середовища. Охарактеризовано алгоритм ACO, поведінку мурах у природі, призначення алгоритму і недоліки в динамічному середовищі, а також модифікації цього алгоритму до мінливих середовищ: ACO з динамічним оновленням феромонів (AACO), ACO з адаптивною пам'яттю (MAS-ACO), ACO з алгоритмами машинного навчання (MLACO). Проаналізовано переваги й недоліки цих модифікацій. Досліджено програмні засоби AntTweakBar, AntOpt і EasyAnt, які реалізують функціонал цього алгоритму, а також надають можливість розробляти нові модифікації алгоритмів ACO і досліджувати наявні. З'ясовано можливості алгоритму BCO і описано поведінку, параметри цього алгоритму, досліджено його переваги й недоліки в динамічному середовищі. Розглянуто модифікації Group Bee Algorithm (GBA), Artificial Bee Colony (ABC) і програмні засоби з відкритим кодом PySwarms і PyABC. Досліджено особливості роботи алгоритму PSO, його переваги й недоліки щодо пристосування до динамічних середовищ. Проаналізовано алгоритми Dynamic Particle Swarm Optimization with Permutation (DPSO-P), Dynamic Multi-swarm, Particle Swarm Optimization Based on Elite Learning (DMS-PSO-EL) як модифікації алгоритму PSO для адаптації до динамічних середовищ, а також бібліотеки для роботи з ними SciPy, DEAP, PyGAD, Particleswarm, Jswarm (має широке API і добре написану документацію) та Dlib. Розроблено порівняльну таблицю з найбільш значущими властивостями відповідних алгоритмів і їх модифікацій, таких як стійкість до зміни середовища, складність реалізації, можливість використання для рою БПЛА. Зроблено короткий аналіз схожих публікацій, де порівнюють алгоритми ройового інтелекту, наведено основні висновки щодо проведеного дослідження.

Ключові слова: оптимізація колонії мурах (ACO); оптимізація бджолиних колоній (BCO); оптимізація рою частинок (PSO); адаптація до динамічного середовища.