Received: August 21, 2022. Revised: September 12, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Improved Artificial Bee Colony Algorithm-based Path Planning of Unmanned Aerial Vehicle Using Late Acceptance Hill Climbing

Bakr S. Shihab[1,2] **Hadeel N. Abdullah**[1]* **Layth A. Hassnawi**[3]

[1]*Electrical Engineering Department, University of Technology, Iraq*
[2]*Electromechanical Engineering Department, University of Samarra, Iraq*
[3] *Ministry of Higher Education and Scientific Research and Science and Technology, Iraq*
* Corresponding author's Email: hadeel.n.abdullah@uotechnology.edu.iq

Abstract: The unmanned aerial vehicle's primary goal (UAV) in route planning is to create a flight path from starting point to the ending point in space to avoid obstacles. The UAV path planning needs a real-time adaption and rapid reaction to the dynamic nature of the operational area since the path computing time, and average path length is very important factors over some cost function that reflects its effectiveness, such as power consumption or average trip time. Creating an ideal path must embrace a particular standard to reduce the distance the drone must travel. The artificial bee colony (ABC) represents one of the most important global search algorithms. The main problem with ABC is that it neglects the local search factor and uses the total search to find the optimal path to the target point since it searches the path to the target in remote areas rather than dealing with the nearby or neighboring areas, and this requires more time to find the path to reach the goal. On the other hand, late acceptance hill climbing (LAHC) algorithm uses the local search operator, which searches for the target point in the areas close to the starting point, thus providing a shorter path to reach a goal. This paper proposes and evaluates a new drone path planning algorithm, hybrid artificial bee colony (HABC). HABC algorithm design is based on modifying the ABC algorithm by cross-layer design between ABC and LAHC algorithms. The HABC is different from the original ABC algorithm in that it modifies the original one to reduce the total search by 3% and pushes ABC's search agent to use local search by 3% to start the process of rediscovering a new path to the target. The evaluation and analysis were performed for several performance metrics under different static evaluation scenarios. It has been observed from the results that the HABC outperforms the original ABC concerning average path length and standard deviation by a reduction reach to 25%, which leads to improved path planning.

Keywords: Path planning, Avoid obstacles, UAV, Artificial bee colony, Late acceptance hill-climbing.

1. Introduction

UAVs recently rose to the status of one of aviation's most challenging and advanced technology [1]. UAVs exhibit several benefits in both military and civilian uses, including low power consumption, unmanned operation, superior camouflage, low price, and great maneuverability. UAVs have been used for precision agriculture, science, and business [2, 3], surveillance, delivery of goods, aerial photography, monitoring of earth resources [4], and border protection [5]. Although UAVs have been widely employed in outdoor applications, recent technological advancements have made it possible to deploy UAVs for indoor navigation applications. Flying an indoor drone can be done for a variety of purposes, including training, competition, and business user.

Professional indoor drones are used for security checks, 3D modelling, and inspections. A variety of drone models utilized for numerous indoor applications are depicted in Fig. 1.

There are many path planning algorithms, and the artificial bee colony (ABC) is one of the most important global search algorithms. The ABC is based on a group of solutions explored by a swarm of bees.

Received: August 21, 2022. Revised: September 12, 2022.

Figure. 1 Several types of UAVs

Thus, ABC explores the solution space effectively, looking for the best food source. However, the ABC is less competitive in terms of exploitation in comparison to local search algorithms late acceptance hill climbing algorithm (LAHC) is one of the important local search algorithms that depend in its search for the target on only one solution in the areas near and adjacent to the starting point. The proposed method in this study aims to modify the ABC algorithm's search engine by balancing exploration and exploitation. This improves its performance in finding the shortest and collision-free path to the target.

Many search algorithms use a series of presumptions based on the features of the indoor environment investigated while using UAVs for mission planning; the most important conditions are as follows:

1- None of the basic path points fall within an obstacle.
2- The line segment joining two successive points of the cardinal path does not intersect with an obstacle.
3- That the straight line representing the path does not go beyond the limits of the environment used.

The main contribution of this work is to provide an effective hybrid algorithm for UAV path planning. Also, we propose an efficient objective function for evaluating the solutions of the proposed algorithm.

A bee colony can be considered a dynamic system was gathering information. The adjustment behavior of bees in the colony to explore an environment suits the task of robotic path planning. Besides the excellence of ABC in exploring an environment, we hybridize it with a local search to intensify the search around potential solutions. This helps the ABC find a shorter path than one found in the bee colony, which directs the colony to investigate new food sources of better quality. The local search used in this study is the LAHC, which has a benefit over other local search algorithms by memorizing previous paths' lengths. This enables the algorithm to investigate new paths of shorter paths than previous ones, even if they are temporarily worse than the current ones. That is, the algorithm possibly considers long paths for the purpose of skipping local optima. Finally, the algorithm will provide the best path that enables the UAV to save energy and time.

The remainder of this paper is organized as follows: previous path planning work is presented in section 2. Section 3 demonstrates the idea of path planning, its importance and problem, and how heuristics can be used to solve algorithms and exploratory aids. The proposed algorithm is presented in section 4. The results obtained for ABC and HABC algorithms are presented in section 5. Finally, the conclusion and future work are presented in section6.

2. Related work

Many techniques were proposed to develop an optimal UAV path for various missions.

The belief roadmap algorithm (BRM), which incorporates a prediction model to detect the surroundings, was created by the robust robotics group at MIT for path planning with obstacles in an indoor setting [6].

In [7], the author introduced a novel ant colony optimization (ACO) algorithm for collision-free path planning that considers dynamic threats and static barriers.

In [8] introduces an original multi-destination path planning approach for unmanned aerial vehicles (UAVs) named MQTPP (multi Q-table path planning). MQTPP aims to reduce the computational burden of cyclical/continuous path planning through a Q-learning planning process while overcoming the fixed path origin problem.

In [9] has been introduced to address this problem using a mixed integer linear programming (MILP) formulation for FBS path optimization in terms of traveling time considering co-channel interference and time windows in serving ground users (GUs) at cluster points (CPs) for constructing the multiple UAV paths in both a single-cell and a multi-cell wireless network. In the proposed technique, we assume that all FBSs depart and return to a single depot, considered a terrestrial 5G/6G base station (BS).

In [10], The author presented a global path planning method based on improved ant colony optimization. The method achieves fast convergence

Received: August 21, 2022. Revised: September 12, 2022.

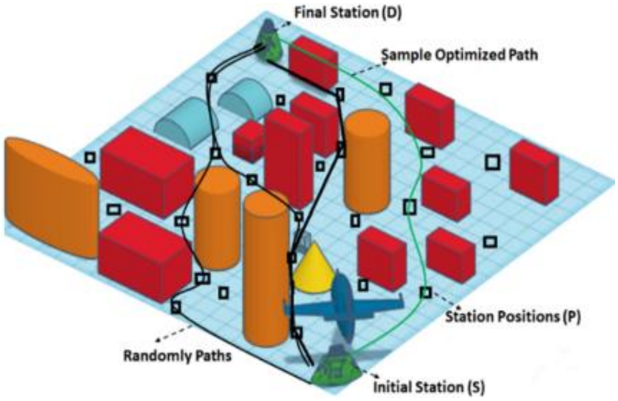


Figure.2 path planning and obstacle avoidance

and guarantees global search capability but neglects the local search factor, which is one of the disadvantages of this method. A new pheromone update rule is presented using fuzzy control to change the value of the pheromone heuristic factor and expectation heuristic factor, adjusting the evaporation rate in stages.

In [11], the authors used optimization methods to delete useless nodes, shorten path length, and reduce the number of inversion points in both directions. Finally, a cubic spline function is introduced to smooth the path simulation. The results show that the optimized A* algorithm is effectively optimized in terms of path length, The number of inflection points, and smoothness. But the path remains long compared to other algorithms.

In [12] studies on the problem of generating feasible cooperative paths for formation rendezvous of unmanned aerial vehicles (UAVs) using particle swarm optimization, one of the problems with this algorithm is that it takes a longer time to calculate the path, and this results in longer paths than the paths of the proposed algorithm.

In [13], a new method of smooth path planning is proposed based on a genetic algorithm. It is applied to solve the problem of redundant nodes and peak inflection points in the path planning process of traditional algorithms. Still, the paths it produces are long and unsafe, which is one of its disadvantages.

The authors changed the artificial bee colony algorithm's path planning in [14] to address the algorithm's weak local search capability and slow convergence time. Based on a balanced search factor, their job involves doing a depth search while preserving a specific level of population variety. Both the capacity for local growth and global search are equal. They noted from the data that the robustness of the path, the average accuracy of path planning, and the capability to find better path options are all improving.

3. The basic principles

3.1 Path planning and obstacle avoidance

Finding the route from one point to another is the process of path planning. Path planning issues are frequently encountered in logistics transportation, automobile navigation systems, computer communication networks, and decision-making systems for individuals or the general public fleeing calamities [15]. Finding an optimal or nearly optimal path is crucial for both network design and graph theory applications [16]. The optimum and heuristic algorithms are the primary classifications of optimal path algorithms [17]. Although heuristic and metaheuristic algorithms can more effectively traverse the search space and discover an optimal path than the best algorithm, the optimum algorithm is inefficient for real-time navigation systems [18] because it increases the time complexity when using a complex PRM. Algorithms for local multiresolution search and the best-first search algorithm are only a few examples of the numerous heuristic algorithms available in [19-21].

On the other hand, obstacle avoidance is one of the most important applications that avoids collision and keeps the vehicle on course to the target. Obstacle avoidance usually differs from path planning since it's usually performed in an interactive environment. In contrast, path planning involves pre-calculating an obstacle-free route in which the controller guides the vehicle.

With the recent advances in the UAVs sector, there is an urgent need for obstacle avoidance features for unmanned vehicles and, therefore, optimal obstacle avoidance [22]. Fig. 2 shows path planning and obstacle avoidance Concepts.

This study proposes a fitness function that calculates the path between source and destination. Practically, this function calculates the Euclidian distance between every two points of the obtained path. Also, this function takes into account the obstacles between the two points. Also, this function penalizes each point in the path that hits an obstacle, as shown in Eq. (1).

$$f(x) = (1 + \mathrm{E}) \times \sqrt{\sum_{i=1}^{n-1}(x_i - x_{i+1})^2} \quad (1)$$

Where x is a solution (path) produced by the proposed algorithm, and n is the number of points that constitutes the path. E is the error portion caused by hitting obstacles, which is calculated as follows:

$$E = \sum_{i=1}^{n} k/d \quad (2)$$

Where k is a scaling factor of an obstacle, and d is the distance between the obstacle and the *ith* point.

3.2 Artificial bee colony

The foraging behavior of natural bees served as the model for the artificial bee colony algorithm. Bee colonies are made up of employed, onlooker, and scout bees. They work together to identify the finest nectar source or find the best solution [23]. It is frequently utilized in many fields because of its few parameters and versatility [24]. The algorithm starts by initializing the first response.

$$(x_1, x_2, \ldots, x_n, x_{SN})\ x_i = \left(x_i^1, x_i^2, \ldots, x_i^D\right) \quad (3)$$

Which represents the candidate location searched by the i bee in the space ($i = 1, \ldots, SN$) , and the number of employed bees is SN. An employed bee modifies a food source as follows:

$$x_i^j = x_{min}^j + rand(0,1) \times \left(x_{max}^j - x_{min}^j\right) \quad (4)$$

Where $j = 1,2, \ldots, D$

In Eq. (4), x_{min}^j is the simplest possible answer; x_{max}^j is the highest possible value for the answer, while rand (0,1) is a random number between (0,1); $i \in \{1,2,3, \ldots, SN\}$, SN Each answer is a D-dimensional vector, where is the total number of food sources. By using a cross-and-variation method, the hired bees communicate with a partner who is chosen at random and works in a new place during each iteration x_i^{*j}Search to search the present site.

$$x_i^{*j} = x_i^j + rand(-1,1) \times \left(x_k^j - x_i^j\right) \quad (5)$$

The first employed bee in this equation shares information with the other k employed bees in the j element; j is a random integer of $(1, D)$, k is a random integer of $(1, SN)$, and satisfies the condition of $k \neq i$. The created food sources are assessed using the evaluation tool. The assessment function is:

$$fitness\ (x_n) = \begin{cases} \frac{1}{1+f(x_n)} & f(x_n) \geq 0 \\ 1 + f(x_n) & f(x_n) < 0 \end{cases} \quad (6)$$

A greedy method was chosen after cross-variation of the employed bees, and if the new location x_i^{*j} is preferable because it would keep the previous location in its original location. The employed bee will provide the following information on the source of the honey bee after all of the employed bees have finished their search. Then an onlooker bee will use Eq. (6) to choose an employed bee to follow:

$$P_i = \frac{fitness(i)}{\sum_{j=1}^{SN} fitness\ (j)} \quad (7)$$

Each onlooker bee chose an employed, when $P_i \geq\ rand$ (0,1), the onlooker chooses the i employed bee and searches nearby:

$$Y_i^{(*j)} = x_i^j + rand(-1,1) \times \left(x_k^i - x_i^j\right) \quad (8)$$

Suppose a food source (solution) reaches a limit cycle during an iteration and is not updated further. In that case, it is assumed that the solution has reached the local optimum, and the food source shall be discarded. After that, a new food source will be generated to take its place, the employed bee connected with this food source will be converted to a scout bee, and the employed bee search process will repeat [25]. Using the greedy selection technique once more, if the employed bee finds a spot faster than the one being followed, it travels there right away or remains as it is.

The main steps of the algorithm are given below [23]:

Step 1: Initialize a set of food sources randomly.
Step 2: REPEAT.

a. Put the remembered employed bees on the food sources.
b. Keep in mind the observer bees on the food sources.
c. Scouts should be dispatched to the search area to look for fresh food sources.

Step 3: UNTIL (conditions are fulfilled).

Each cycle of the search in the ABC algorithm consists of three steps: sending the employed bees onto the food sources and then measuring their nectar amounts, and selecting the food sources by the onlookers after sharing the information of the employed bees and determining the nectar amount of the foods. The last step is identifying the scout bees and sending them to potential food sources.

The ABC algorithm mainly focuses on exploring the problem space, and thus it does not effectively exploit the neighborhood of a given solution. This study integrates the ABC with a local search algorithm to obtain a balanced search procedure [26].

Algorithm 1: Pseudo code of the late acceptance hill climbing (LAHC)

$X \leftarrow$ *Initial CRNN structure*
$L \leftarrow$ *obtained from the SSA length of the list*
for i=1 to L do
  fi = f(X)
X = X*
for i=1 to Max_iterations do
  X'=NS(X)
  v = i mod L
  if f(X') ≤ fv||f(X') ≤ f(X) then
    X = X'
    if X' < X then*
      X* = X'
    fv=f(v)
*Output: X**

3.3 The late acceptance hill-climbing (LAHC)

Burke and Bykov [27] proposed a straightforward, practical, and efficient local search method in 2016 under late acceptance hill-climbing (LAHC). It can be viewed as an improved form of the hill climbing (HC) algorithm. The main distinction between the two is the acceptance criterion used to compare the new solutions to those found through earlier iterations. In LAHC, the searching process accepts a new solution even if it is worse than the current solution. However, the solution must be better than the one obtained in previous iterations. This helps the algorithm to escape local optima, which has been proved in recent studies [28]. The LAHC begins with a random initial solution and accepts or rejects new solutions until the stopping condition is satisfied. History length **(L_h)** is a fixed length of the internal list which LAHC maintains to memorize prior fitness values. The state of the LAHC algorithm is represented during LAHC execution by the variable Idle, which is increased by one if there is no improvement and decreased to zero if there is. If a new solution is not better than the current one, the LAHC acceptance criterion will be invoked. This consists of comparing the earliest fitness value in the LAHC list -with respect to the current iteration- to a newly generated solution. The latter will be accepted as a new search point if it is better than the one on the list. Practically, the LAHC will establish a hypothetical beginning of the list indexed as v. The new solution is accepted and moved up to the top of the list if it is superior to the ***v*** position in ***L_h***, and the previous solution is removed from the bottom of the list. The LAHC pseudo-code is presented in Algorithm 1. An LAHC algorithm represents the process of calculating the best path:

$$v = i \bmod L_h \tag{9}$$

When **v** is the item we want to compare with it; ***i*** number of iterations; ***L_h*** is a list length.

The user chooses a single algorithmic value for the listed size. Setting the list length to a high value can improve LAHC's accuracy at the expense of the computation time [27]. Hence, we adopt the exact value of the list length, which has been proved by the original LAHC study [27]. The main advantage of this algorithm is that it uses the local search operator, which searches for the source of food in the areas close to the starting point, thus providing a shorter path to reach a goal.

4. Proposal algorithm: hybridized ABC with LAHC

The proposed HABC is a new algorithm designed for drone path planning. The HABC is based on the notion of sharing information inspired by the bee's foraging behavior which is enforced by a local search algorithm. The HABC in this study searches the problem space for the best food source. A local search is used within 3% to improve the best food source. Then, other food sources will be affected when employed bees share information. These operations lead the algorithm to converge toward the shortest path of the given environment.

The steps of the proposed HABC are presented in Algorithm 2.

4.1 Performance metrics

In this study, the proposed algorithm is run thirty times. Then the evaluation is based on the following metrics:

- **Minimum path length (Min. length):** This value shows the minimum path length obtained over the thirty runs.
- **Average path length (Ava. PL):** This value shows the average path length over the thirty runs, which is calculated as follows:

$$Ava.PL = \frac{\sum_i^m L_i}{m} \tag{10}$$

  where L_i is the best path length obtained at an *ith* run. *m* is the number of runs.
- **Standard deviation (SD):** This metric shows the variance among the thirty runs based on the best path length of each run.

Algorithm 2: Pseudo code of hybrid artificial bee colony (HABC)

Input: f(x)
initialize food sources X, x_i (i=1,2, ..., N)
for i=1 to N, do
fi = f(xi)
x^{best} = $argmin_x$ f(x), $x \in X$
while (t< max cycles) do
for each Embloyed_bee, i do
Select a neighbor food source x_d, where d = i
Select a jth decision variable from x_i
Generate a new food source s based on x_{ij} and x_{dj} using Eq. 5.
Replace xi by s if the latter has a minimum path length.
for each Onlooker_ bee, i do
Select x_i within p_i, where p_i is calculated in Eq. (4).
Select a neighbor food source x_d, where d = i
Select a jth decision variable from x_i
Generate a new food source s based on x_{ij} and x_{dj} using Eq. (5).
Replace x_i by s if the latter has a minimum path length
if a food source is abandoned, then
Send a Scout bee to explore a new food source
update x^{best} if any new food source is better
if rand < lp, then
x^{best} =LAHC (x^{best}). Call the local search algorithm
t = t + 1
Output: G^{best}

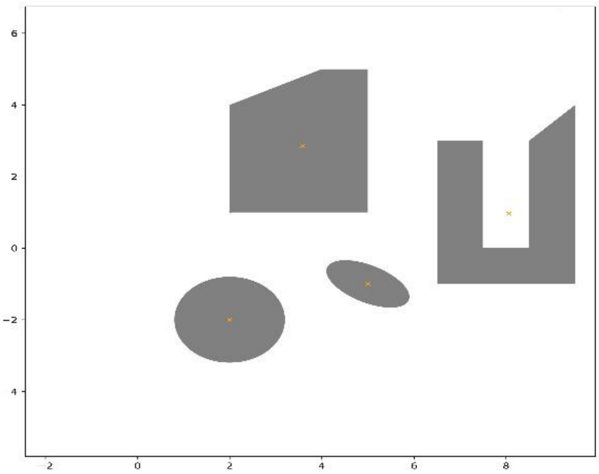


Figure. 3 environment used in this paper

$$SD = |Ava.PL - PL| \tag{11}$$

4.2 Evaluation scenario

The evaluation scenario and analysis experiments of the proposed HABC are simulated and evaluated using the MSI KATANA GF76 11UE laptop simulator. It has both Windows and Linux systems and uses Python 3.7. The evaluation scenario is modeled with the following specifications:

1. A static indoor computer environment of 4 obstacles of different sizes and shapes is used.
2. Eleven cases are implemented. Each case represents one starting and one target point, as shown in Fig. 3.
3. Thirty runs are made for each case.
4. Five hundred iterations for each run are made to reach the best track.
5. Twenty bees (20 solutions) are sent to each case to search for a food source (the target point).
6. HABC and standard ABC algorithm are applied to the proposed environment shown in Fig. 3.
7. Simulation of the new algorithm HABC using the simulator program to prove the results in practice.

5. Result and discussion

Different types of experiments were carried out to evaluate and analyze the HABC performance with respect to the performance of the traditional ABC algorithm. The results show that the performance of HABC outperforms the performance of ABC with respect to path length, average path length, and standard deviation for all evaluation cases. Table 1 shows a performance comparison between ABC and HABC.

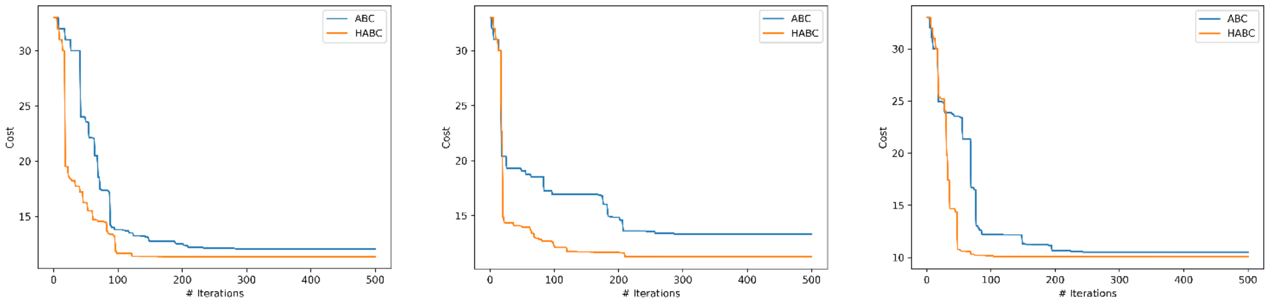
If we take the first case (start point (0.5), goal point (8, -2)), we can note that the path length using the traditional ABC algorithm is 11.667 m, while the path length using HABC is 11.269, which indicates an improvement in the performance of HABC concerning path length. Fig. 6 compares ABC and HABC in an environment shown in Fig. 3.

On the other hand, the results demonstrated that for the same case 1, the average path length using ABC is 16.672m, while the average path length using HABC is 11,682 m, representing a significant improvement. HABC is very close to the shortest path, unlike the standard algorithm ABC which was far from the shortest path, as shown in Fig. 6. This result is due to using local search in addition to the global search of the traditional ABC.

This improvement, in turn, affects the cost, time of arrival, and UAV energy consumption. Table.1 compares the new algorithm HABC and the standard ABC algorithm. For example, we also note case 1, that the proposed algorithm found the shortest path in

Table 1. Comparison ABC and HABC in same environment

Cases	ABC					HABC				
	Start	**Goal**	**Min. path length (m)**	**Average length(m)**	**Standard Deviation (S.D) (m)**	**Start**	**Goal**	**Min. Path length(m)**	**Average length (m)**	**Standard Deviation (S.D) (m)**
1	(0,5)	(8, -2)	11.669	16.672	3.475	(0,5)	(8,-2)	11.269	11.682	0.393
2	(2,5)	(6, -5)	12.791	16.418	4.100	(2,5)	(6,-5)	11.231	11.335	0.118
3	(5,6)	(6, -4)	10.107	12.213	1.370	(5,6)	(6,-4)	10.057	10.063	0.005
4	(8,7)	(2,-4)	14.501	17.365	1.732	(8,7)	(2,-4)	14.382	16.014	1.530
5	(7,-4)	(-2,5)	13.323	18.145	4.267	(7,-4)	(-2,5)	12.728	12.737	0.017
6	(10,0)	(-2,0)	13.265	17.406	3.199	(10,0)	(-2,0)	13.035	13.809	0.649
7	(0,-3.5)	(8,2)	18.665	22.348	3.308	(0,-3.5)	(8,2)	12.296	15.598	2.809
8	(2,-3.5)	(8,1.5)	20.864	26.524	6.761	(2,-3.5)	(8,1.5)	11.434	16.941	3.724
9	(0,3)	(8,1)	13.882	17.044	3.158	(0,3)	(8,1)	12.828	14.436	2.0233
10	(3.5,-1.5)	(8,1.5)	9.7751	17.118	4.520	(3.5,-1.5)	(8,1.5)	8.643	11.906	2.9277
11	(3.5,5)	(8,2)	9.2807	24.401	12.936	(3.5,5)	(8,2)	6.341	8.0183	1.0380



(a) (b) (c)

Figure. 4 Average solution convergence of 30 runs by two algorithms ABC & HABC for: (a) case. 1, (b) case. 2, and (c) case. 3

Table 2. Comparing the effectiveness of the new algorithm HABC with other algorithms

cases	**Cost (path length) (m)**								
	ABC	**HABC**	**LAHC**	**Ant colony ACO [7]**	**Improve ACO [10]**	**A-star [11]**	**PSO [12]**	**Genetic GA [13]**	**ACO-GA [13]**
1	11.669	11.269	17.146	30.02	30.384	63.038	35.06	33.796	32.968
2	12.791	11.231	12.742	29.65	30.970		35.03	32.8663	31.796
3	10.107	10.057	10.589	18.72	32.141				
4	14.501	14.382	14.458	15.91	39.698				

120 iterations, while the standard algorithm delayed to 280 iterations. Despite that, the proposed algorithm path was the shortest and best. This situation is also noted in case 2, case 3, and all other cases, as shown in Fig. 4.

In Table 2, a comparison was made of the proposed algorithm with the previous path planning methods. The comparison showed a clear superiority of the proposed algorithm in the length of the path significantly, which confirms the effectiveness of the proposed algorithm in this paper. Table. 2 Comparing the effectiveness of the new algorithm with other algorithms. Fig. 5 shows a lower standard deviation (SD) in the hybrid algorithm HABC.

The shortest path was drawn from the starting point to the target point using the new algorithm. The vehicle traveled according to the path set for it, bypassing all obstacles without hitting any obstacle

Received: August 21, 2022. Revised: September 12, 2022.

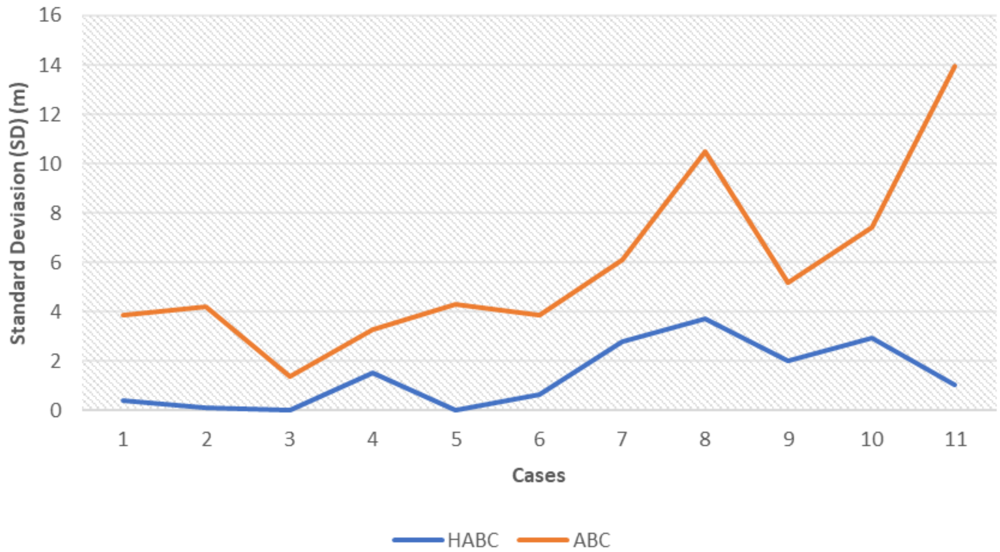
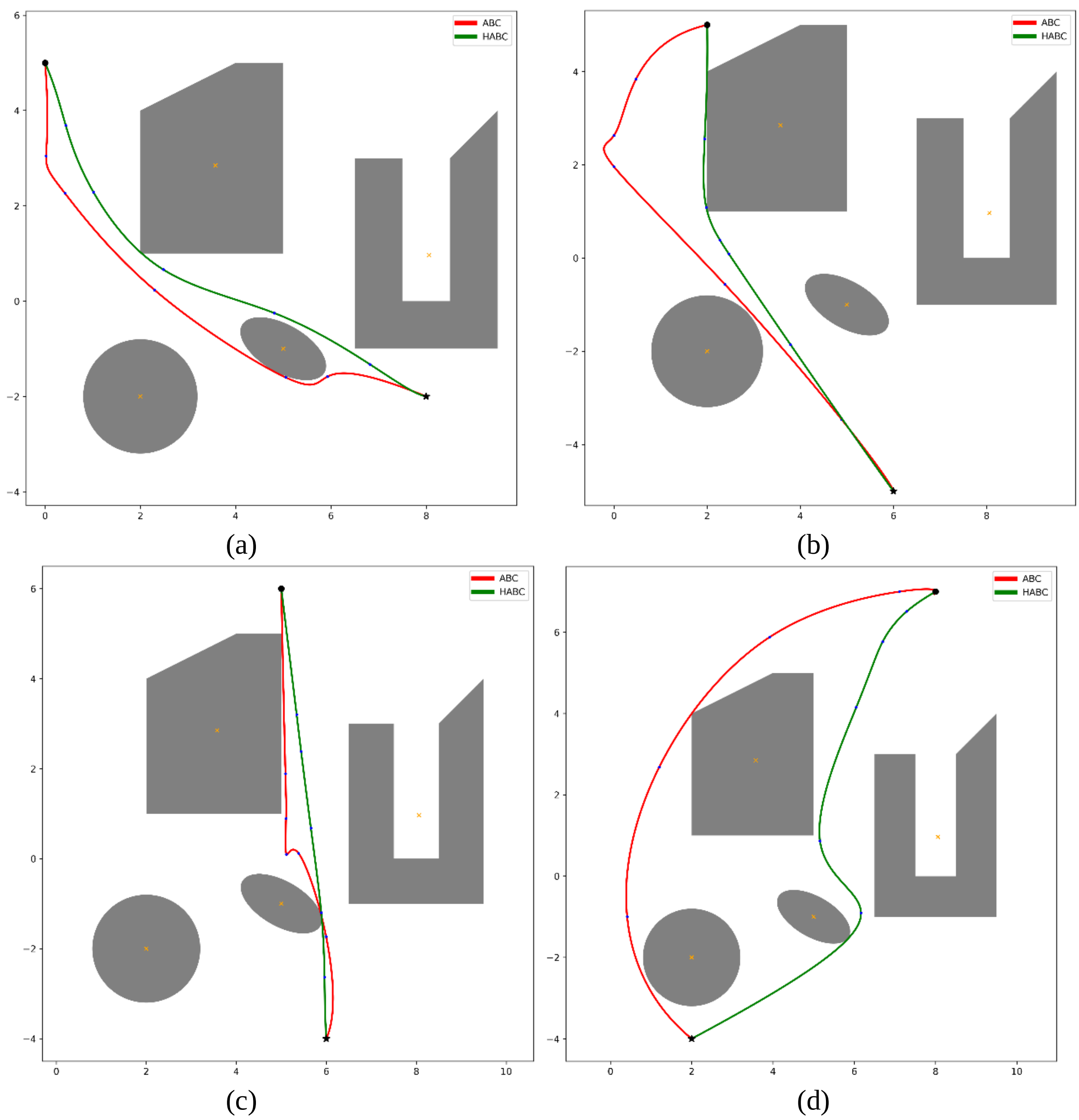
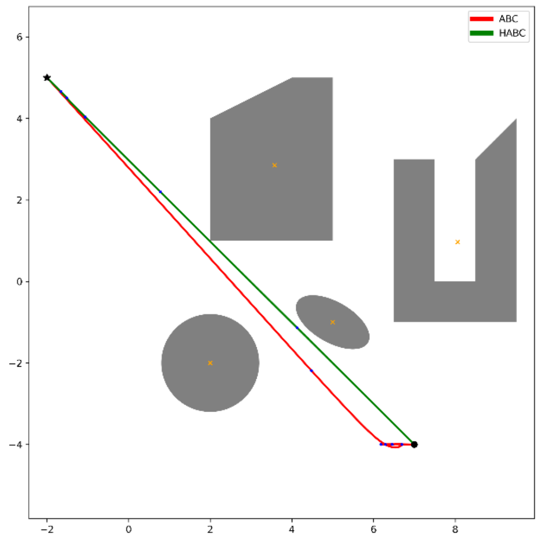


Figure. 5 Comparison the standard deviation (SD) for each case of HABC and ABC algorithms

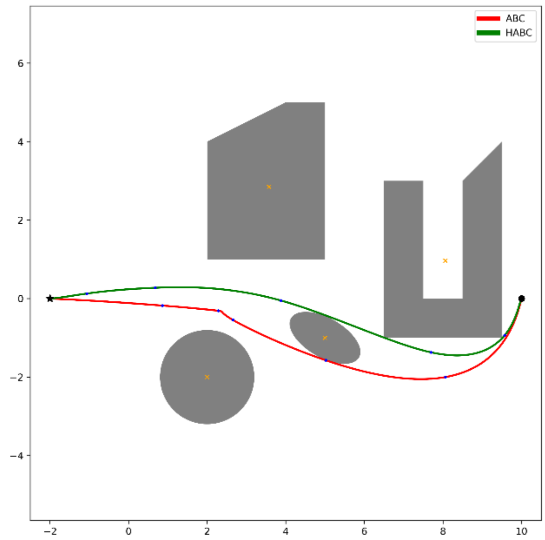


 DOI: 10.22266/ijies2022.1231.39

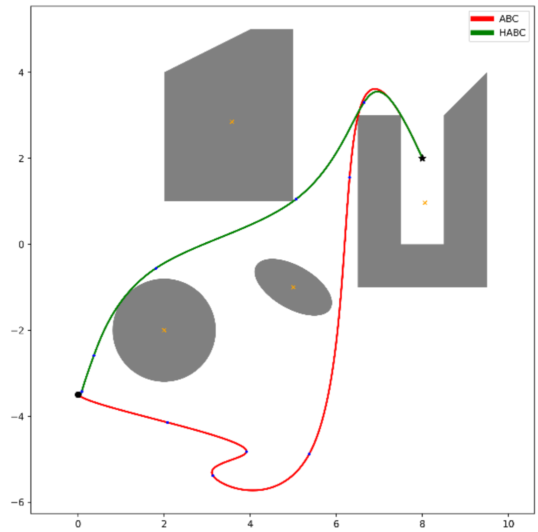
Received: August 21, 2022. Revised: September 12, 2022.



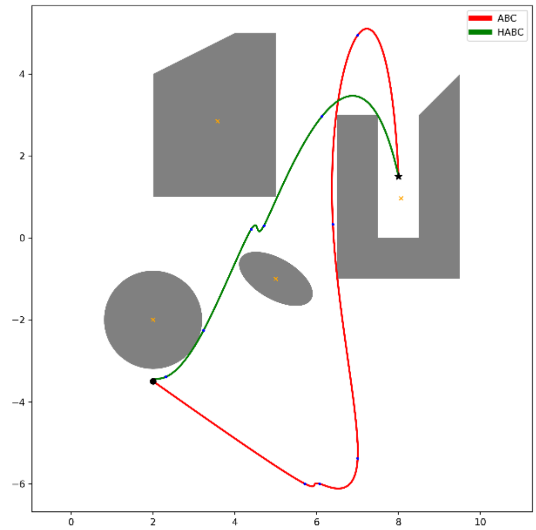
(e)



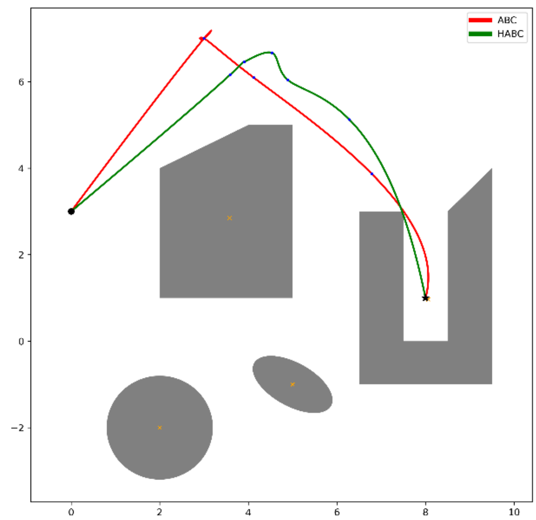
(f)



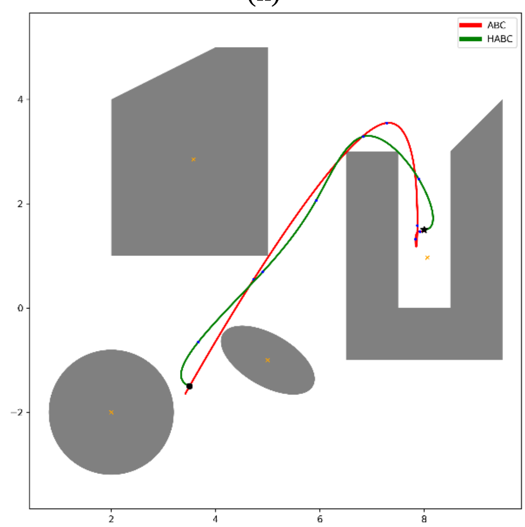
(g)



(h)



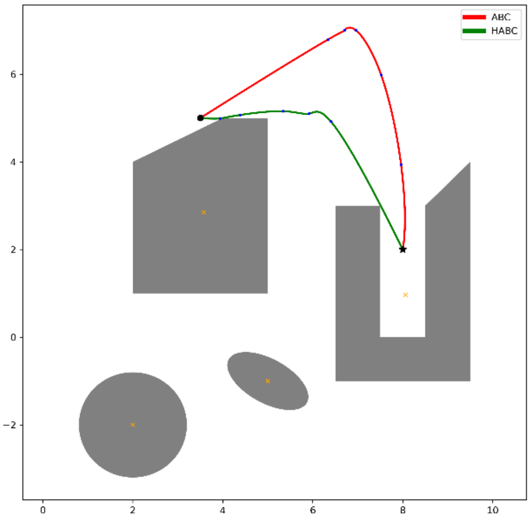
(i)



(j)

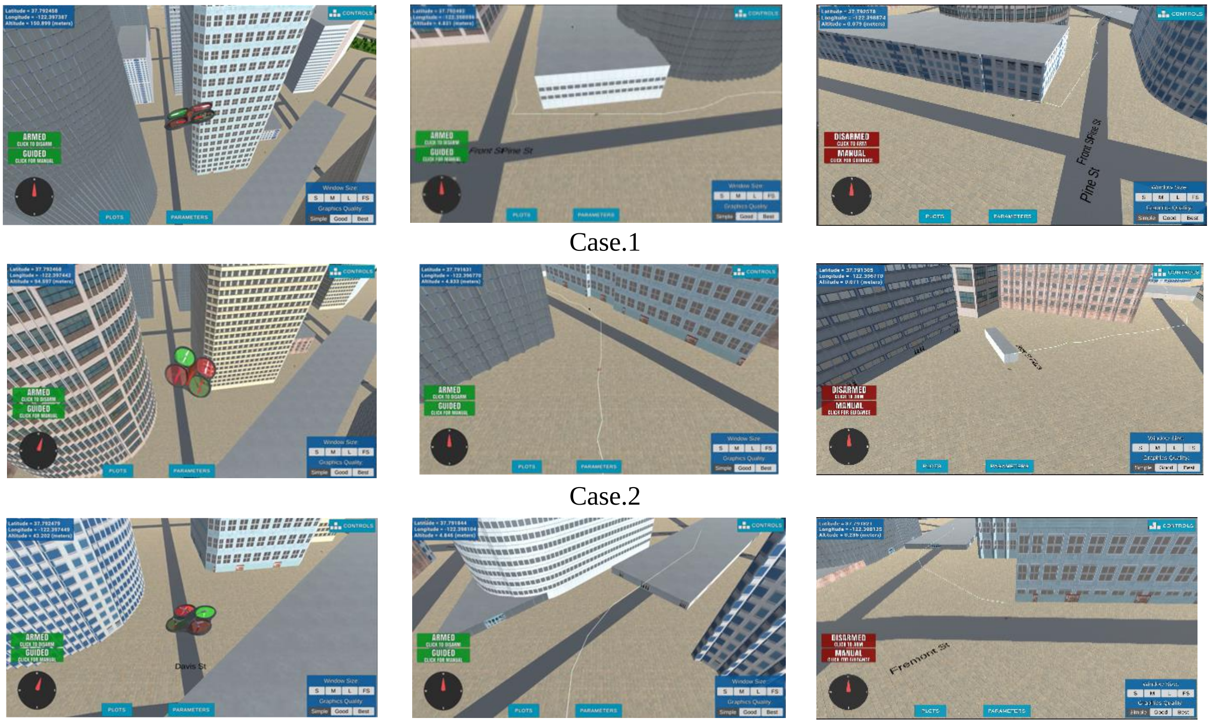
 DOI: 10.22266/ijies2022.1231.39

Received: August 21, 2022. Revised: September 12, 2022.



(k)

Figure. 6 Comparison ABC and HABC algorithms to find the minimum path in a same environment: (a) case. 1, (b) case. 2 (c) case. 3, (d) case. 4, (e) case. 5, (f) case. 6, (g) case. 7, (h) case. 8, (i) case. 9, (j) case. 10, and (k) case.



Case.1

Case.2

Case.3

(a) (b) (c)

Figure.7 Applying the Hybrid algorithm HABC to a real environment using a FCND-simulator (Flying car simulator) program: (a) start, (b) during the path and (c) goal

and then reaching the target point, as shown in Fig. 6.

To verify the effectiveness of the new algorithm in practice. The first three points were applied to a natural environment using FCND-simulator (Flying car simulator), as shown in Fig. 7.

6. Conclusion

According to a comparison of 30 runs and 500 iterations, the modified artificial bee colony algorithm's search results are much improved regarding average path length. The updated algorithm's ideal value is based on the local search factor suggested in this study. It increases the track layout's average length, robustness, the algorithm's capacity to find better path solutions, and the ease with which satisfying results can be attained after numerous iterations. The complexity of the problem grows exponentially as the variable's dimension rises, and the path calculation time also rises. As a result, the algorithm struggles to solve high-dimensional optimization problems. Therefore, the next area of research will be how to make the algorithm address complicated issues.

Conflicts of interest

We confirm that there is no conflict of interest in this work.

Author contributions

Conceptualization, and Methodology, L. A. Hasnawi, and H. N. Abdullah; Software, B. S. Shihab; Validation, B. S. Shihab and H. N. Abdulla; Formal analysis, B. S. Shihab; investigation, B. S. Shihab; resources, B. S. Shihab; data curation B. S. Shihab; Writing—original draft preparation, B. S. Shihab; Writing—review and editing, H. N. Abdullah and L. A. Hassnawi; visualization, H. N. Abdullah; supervision, H. N. Abdullah and L. A. Hassnawi; project administration, H. N. Abdullah and L. A. Hassnaw

References

[1] T. G. Grabowski and M. Rodzewicz, "Design of UAV for photogrammetric mission in the antarctic area", In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 231, No. 9, pp. 1660–1675, 2017.

[2] J. Kim, S. Kim, C. Ju, and H. I. Son, "Unmanned aerial vehicles in agriculture: A review of the perspective of platform, control, and applications", *IEEE Access*, Vol. 7, pp. 105100-105115, 2019.

[3] P. R. Grammatikis, P. Sarigiannidis, T. Lagkas, and I. Moscholios, "A compilation of UAV applications for precision agriculture", *Computer Networks*, Vol. 172, p. 107148, 2020.

[4] H. T. Berie and I. Burud, "Application of unmanned aerial vehicles in earth resources monitoring: focus on evaluating potentials for forest monitoring in Ethiopia", *European Journal of Remote Sensing*, Vol. 51, No. 1, pp. 326–335, 2018.

[5] R. Koslowski and M. Schulzke, "Drones along borders: Border security UAVs in the united states and the European Union", *International Studies Perspectives*, Vol. 19, July 2017.

[6] R. He, S. Prentice, and N. Roy, "Planning in information space for a quadrotor helicopter in a GPS-denied environment", In: *Proc. of 2008 IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, pp. 1814–1820, May 2008.

[7] C. Huang, Y. Lan, Y. Liu, W. Zhou, H. Pei, L. Yang, Y. Cheng, Y. Hao, and Y. Peng, "A new dynamic path planning approach for unmanned aerial vehicles", *Complexity*, Vol. 2018, pp. 1–17, November 2018.

[8] M. R. Jones, S. Djahel, and K. Welsh, "MQTPP – Towards Multiple Q-Table based Path Planning in UAV Environments", In: *Proc. of 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, pp. 457-460, February 2022.

[9] J. Lee and V. Friderikos, "Interference-aware path planning optimization for multiple UAVs in beyond 5G networks", *Journal of Communications and Networks*, Vol. 24, No. 2, pp. 125-138, April 2022, doi: 10.23919/JCN.2022.000006.

[10] Y. Tao, H. Gao, F. Ren, C. Chen, T. Wang, H. Xiong, and S. Jiang, "A mobile service robot global path planning method based on ant colony optimization and fuzzy control", *Applied Sciences*, Vol. 11, No. 8, p. 3605, 2021.

[11] L. Zhang, and Y. Li, "Mobile Robot Path Planning Algorithm Based on Improved A Star", In: *Proc. of Journal of Physics: Conference Series*, Vol. 1848, No. 1, p. 012013. IOP Publishing, 2021. https://doi.org/10.1088/1742-6596/1848/1/012013.

[12] Z. Shao, F. Yan, Z. Zhou, and X. Zhu, "Path planning for multi-UAV formation rendezvous based on distributed cooperative particle swarm optimization", *Applied Sciences*, Vol. 9, No. 13, p. 2621, 2019.

[13] J. Ma, Y. Liu, S. Zang, and L. Wang, "Robot path planning based on genetic algorithm fused with continuous Bezier optimization", *Computational Intelligence and Neuroscience*, Vol. 2020, 2020. https://doi.org/10.1155/2020/9813040.

[14] W. Hao, B. Luo, and Z. Zhang "Application study of UAV path planning based on the balanced search factor artificial bee colony

algorithm", *Journal of Physics: Conference Series*, Vol. 2083, No. 3, p. 032064, 2021.

[15] M. Xue, "Optimal path selection of vehicle route scheduling under bad weather factors", *Computer Simulation*, Vol. 32, No. 2, pp. 210–212, 2015.

[16] R. Sotnezov, "Genetic algorithms for problems of logical data analysis in discrete optimization and image recognition", *Pattern Recognition and Image Analysis*, Vol. 19, No. 3, pp. 469–477, 2009.

[17] B. Yan, T. Chen, X. Zhu, Y. Yue, B. Xu, and K. Shi, "A comprehensive survey and analysis on path planning algorithms and heuristic functions", In: *Arai, K., Kapoor, S., Bhatia, R. (eds) Intelligent Computing. SAI 2020. Advances in Intelligent Systems and Computing*, Springer, Cham., Vol. 1228, https://doi.org/10.1007/978-3-030-52249-0_39, 2020.

[18] W. Qiuping, S. Rong, and Z. Qi, "Optimal path selection of slow traffic based on gis network analysis", *Journal of Xi'an University of Architecture & Technology*, Vol. 45, No. 5, pp. 668–674, 2013.

[19] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot", *Procedia Engineering*, Vol. 96, pp. 59–69, 2014.

[20] S. Behnke, "Local multiresolution path planning", In: *Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds) RoboCup 2003: Robot Soccer World Cup VII. RoboCup 2003, Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, Vol. 3020, 2003, https://doi.org/10.1007/978-3-540-25940-4_29.

[21] C. Dellin and S. Srinivasa, "A unifying formalism for shortest path problems with expensive edge evaluations via lazy best-first search over paths with edge selectors", In: *Proc. of the International Conference on Automated Planning and Scheduling*, Vol. 26, pp. 459-467, 2016.

[22] Q. Zhu, Y. Yan, and Z. Xing, "Robot path planning based on artificial potential field approach with simulated annealing", In: *Proc. of Sixth International Conference on Intelligent Systems Design and Applications*, IEEE, Vol. 2, pp. 622–627, 2006.

[23] J. H. Liang, and C. H. Lee, "Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm", *Advances in Engineering Software*, Vol. 79, pp. 47-56 2015.

[24] M. Tareq, S. A. Abed, and E. A. Sundararajan, "Artificial bee colony for minimizing the energy consumption in mobile ad hoc network", In: *Herawan, T., Chiroma, H., Abawajy, J. (eds) Advances on Computational Intelligence in Energy*, Green Energy and Technology, Springer, Cham,z, 2019 https://doi.org/10.1007/978-3-319-69889-2_2.

[25] A. P. Ning and X. Y. Zhang, "The convergence analysis of artificial bee colony algorithm", *Control and Decision-Making*, Vol. 28, No. 9, pp. 1554-1558, 2013.

[26] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm", *Journal of Global Optimization*, Vol. 39, No. 3, pp. 459-471, 2007.

[27] E. K. Burke and Y. Bykov, "The late acceptance hill-climbing heuristic," *European Journal of Operational Research*, Vol. 258, No. 1, pp. 70 – 78, 2017.

[28] S. A. Abed, H. M. Rais, J. Watada, and N. R. Sabar, "A hybrid local search algorithm for minimum dominating set problems", *Engineering Applications of Artificial Intelligence*, Vol. 114, p. 105053, 2022.