

## Recent research in finding the optimal path by ant colony optimization

Sari Ali Sari, Kamaruddin Malik Mohamad

Faculty of Computer Science and Information Technology Universiti Tun Hussein Onn Malaysia, Johor, Malaysia

---

### Article Info

#### Article history:

Received Sep 11, 2020  
Revised Nov 16, 2020  
Accepted Dec 8, 2020

#### Keywords:

Ant colony optimization  
File carving  
Graph theory  
Optimal path

---

### ABSTRACT

The computation of the optimal path is one of the critical problems in graph theory. It has been utilized in various practical ranges of real world applications including image processing, file carving and classification problem. Numerous techniques have been proposed in finding optimal path solutions including using ant colony optimization (ACO). This is a nature-inspired metaheuristic algorithm, which is inspired by the foraging behavior of ants in nature. Thus, this paper study the improvement made by many researchers on ACO in finding optimal path solution. Finally, this paper also identifies the recent trends and explores potential future research directions in file carving.

*This is an open access article under the [CC BY-SA](#) license.*



---

### Corresponding Author:

Sari Ali Sari  
Faculty of Computer Science and Information Technology  
Universiti Tun Hussein Onn Malaysia  
86400 Parit Raja, Batu Pahat, Johor, Malaysia  
Email: alsarisari46@gmail.com

---

## 1. INTRODUCTION

A graph comprises a set of edges and a set of vertices where each of the edges connecting among two vertices. Generally, the object is indicated as a vertex; meanwhile, the edge is indicated as a relationship between two objects. Therefore, a graph might be applied to represent information that can be modeled as objects and relationships among those objects. Considering that graph theory is a study of graphs. Graph theory is used in different practical applications [1] of the real world including image processing, file carving, and classification problem. In graph-theoretic (GT) carving, vertices are indicated as fragments and the edges are associated with weights which the value of the weights denotes the likelihood that in the original file there are two fragments are adjacent. The concept of utilizing a GT approach in the file carving had been explored in [2] and highlighted that the GT file carving had main shortcomings. Most of the time, it failed to find the optimal path. The reason is that they are not exhaustively performed on all the data. They decided too early to certain choices which then prevented them from finding the best path [2].

This paper structure is organized as follows; Section 2, a short overview of the optimal path. The optimization technique is discussed in section 3. Section 4 discusses the ant colony optimization (ACO). The ACO algorithm is discussed in section 5. The recent literature on the improvement of ACO is discussed in section 6. Finally, recommendations and conclusion are in section 7.

## 2. OPTIMAL PATH

In an undirected graph, a path is considered as an optimal path when it fulfills the requirement which all the edges and all vertices (excluding the beginning and the ending point) are different. An optimal path is the one that meets a specific requirement in the path; for instance, it will be termed as the shortest path when the sum of the weight of its unit edges is minimized. The calculation process to find the optimal path in the graph has been used in constructing the optimal path algorithm. Numerous researches have been done in finding the optimal path including simulated annealing algorithm, A\* algorithm, Dijkstra algorithm and Ant Colony Optimization (ACO). These algorithms are examined from various angles and continuously improved and enhanced in order to acquire an efficient optimal path algorithm. This paper focuses only on the ACO to solve the optimal path problem.

## 3. OPTIMIZATION TECHNIQUE

Optimization is the process of finding an optimal or best solution in solving a problem. Meanwhile, an optimization problem can be defined as a process to find the values of the variables which maximize or minimize an objective function and at the same time fulfill the constraints. 3 main factors have been highlighted in the optimization problem. The first factor is minimizing or maximizing an objective function. The second factor is variables or unknowns set which affecting the objective function and the third factor is a set of constraints allowing the unknowns to accept only particular values while excluding others. There is more than one local solution in most optimization problems. Thus, it is so important in selecting a good optimization method that will not be greedy and search just in the neighborhood area of the optimal solution; this will be caused by misleading the process of searching and trapped at a local solution. Nevertheless, the optimization algorithm must have a mechanism for balancing among local search and global search. Different approaches to solve mathematical and combinatorial in optimization problems are used [3]. Besides, in case the search space is large, or the optimization problem is complicated, the problem of optimization with traditional mathematics will be difficult to solve. For that purpose, several metaheuristic optimization techniques have been developed to resolve these tough problems regarding optimization [4].

The objective function value is controlled by a set of variables. These variables are very important for the optimization problem. Without these variables, the constraints and the objective function cannot be determined. A set of constraints accept the variables to pick several values and reject the remaining values. The constraints are considered as not important and their existence relies on the optimization problem requirements. The optimization problem is finding the values of the variables which maximize or minimize the objective function as well as satisfy the constraints. The formulation of the optimization problem for maximizing the objective function can be expressed as;

$$\text{To find } X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \text{ which maximizes } f(X) \text{ subject to the constraints:} \quad (1)$$

$$g_a(X) < 0, \quad a = 1, 2, \dots, m \quad (2)$$

$$h_b(X) < 0, \quad b = 1, 2, \dots, p \quad (3)$$

Based on the given formulation above, (X) is defined as an (n-dimensional vector) known as (design vector),  $f(x)$  is the objective function,  $g_a(X)$  is inequality constraints and  $h_b(X)$  is equality constraints. The number of n, m and p is not equivalent at all. This problem is categorized as a constrained optimization problem. The optimization problem may have only one objective function known as a single-objective optimization problem or many objective functions known as multiple optimization problems. An example given below is a clear example of a constrained optimization problem with single-objective.

Minimize cost,

$$Y_1 + f(x_1, x_2, x_3, x_4) = 2x_1 + 3x_2 + 1.5x_3 + 0.8x_4 \quad (4)$$

Subject to the constraints:

$$x_1 + 2x_2 + x_3 \leq 150 \quad (5)$$

$$x_1 + x_2 + x_3 + 0.3x_4 \leq 170 \quad (6)$$

$$3x_1 + 1.5x_2 \leq 128 \quad (7)$$

The ranges of the variables:

$$0 \leq x_1 \leq 50 \quad (8)$$

$$0 \leq x_2 \leq 32 \quad (9)$$

$$0 \leq x_3 \leq 44 \quad (10)$$

$$0 \leq x_4 \leq 18 \quad (11)$$

Based on the example given, the objective function and the constraints are indicated as linear equations in this optimization problem.

#### 4. ANT COLONY OPTIMIZATION

It is a very challenging task to find an optimal solution for the optimization problem. It relies on the correct use and the selection of a suitable algorithm. The selection of an algorithm is based on the computational resources, type of problem, time constraints and the availability of algorithms. However, for global optimization, non-linear, large-scale problems, usually, there is no agreed guideline for the selection of algorithm, and there is no efficient algorithm in many cases.

A metaheuristic is also known as a technique or procedure intended to find an approximate solution in a low computational time (short time). A diverse metaheuristic algorithm including stochastic in nature and nature-inspired has emerged. Mostly, metaheuristic algorithms are population-based algorithms. Yet one metaheuristic approach that gives us the opportunity for interactive learning is inspired by the foraging behavior of ants called ant colony optimization (ACO). Over time, ants have the capability in finding the optimal solution from the food source towards their nest through communication rather than visual cues. Ants deposit chemical information called pheromone when coming back once they find the object. Ants used these pheromones in attracting other ants to choose the same path, thus, all the colony searching for a solution in the space near to the previous good solution [5]. Deneubourg *et al.* [6] discovered by following the behavior of ants as well as pheromone laying. They conducted an experiment called the “double bridge experiment”. They found out that two bridges with the same length were connecting between the nest of the Argentine ants' colony and the food source as exposed in Figure 1(a). Then, ants reached their food source after exploring the surroundings of their nest. Besides, they deposited pheromone along their way between food source and nest. In the beginning, every ant chose one bridge randomly. But then, after some time, one of the bridges showed a higher pheromone concentration compared to the other bridge because of random fluctuations and consequently, attracted more ants until the whole colony converges to this bridge.

Later, the variation of the double bridge experiment has been carried out by Goss *et al.* [7] which one bridge is longer than the other bridge as presented in Figure 1(b). It was noticed that the stochastic fluctuations are much decreased in the initial choice of a bridge and a second method becomes a crucial role: the ants who chose the short bridge were the first ones to reach the nest. Thus, the short bridge received pheromone earlier compared to the long bridge. Therefore, it increases the probability that more ants choose the short bridge instead of the long one.

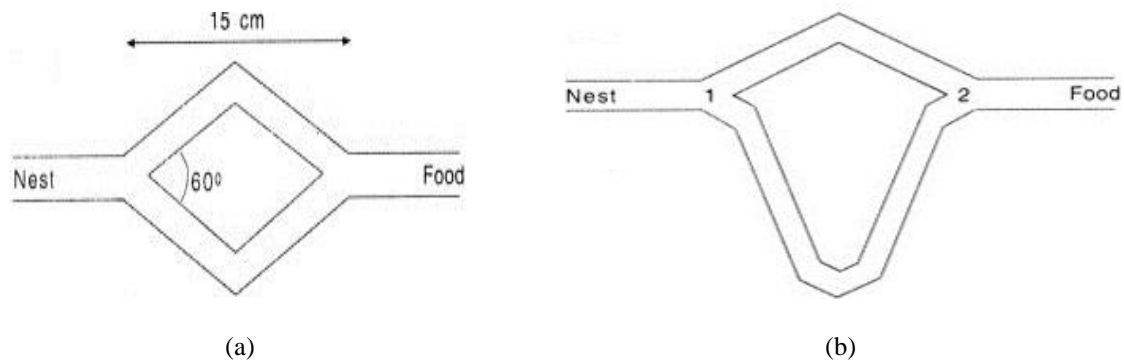


Figure 1. Double bridge experiment observations, (a) Equal lengths of a bridge, (b) Different lengths of bridge

Contrasting with other evolutionary algorithms, the ACO algorithm is a reactive search optimization method based on the “learning while optimizing” principle. ACO algorithm has been implemented due to the high performance in solving diverse combinatorial optimization problems, for example, image registration, the quadratic assignment problem, the traveling salesman problem (TSP) and other problems. These challenge problems are considered as highly efficient of the ACO algorithm compared to other metaheuristic algorithms.

Four main reasons have been highlighted that motivates us for choosing ACO [5]. Firstly, ACO effectively solves numerous NP-hard combinatorial optimization problems secondly, it has a high convergence speed and few parameters. Thirdly, it is simply still highly efficient in solving a problem with a large search space. Fourthly, during the optimization procedure, it still can keep learning. Basically, ACO is originally intended to optimize discrete problems compared to particle swarm optimization. It can prevent conversion between a continuous domain and discrete domain, which leads to efficiently performs and reducing computational complexity with large discrete in searching space. Besides, the strategy of solution construction arms ACO with high diversity as well, therefore ACO has the capability in handling complex optimization problems. All these reasons motivate us to propose ACO based on GT based carving. The advantages and disadvantages of ACO are tabulated in Table1.

Table 1. The advantages and disadvantages of ACO

No.	Advantage	Disadvantage
1.	ACO's efficiency is more prevalent when we do not know the particulars of a question or what kind of algorithm should be used in advance.	ACO is not found effective for dealing with large-scale combinatorial questions.
2.	It's easy to pair it with other algorithms, resulting in enhanced robustness and increased reliability.	Initial information of the algorithm is scarce, it is easy to fall into the local optimum.
3.	The ACO algorithm, one of the main yet incomplete algorithms, does not only boast the advantages such as high speed and high accuracy but it also searches for a quasi-optimal solution more quickly.	The option of Pheromone-related parameters in the formula is largely empirical; therefore, if the pheromone parameters are selected without experiment and theoretical arguments, chances are that the results obtained would be unclear or oblivious due to the inefficacy and poor convergence of the algorithm

## 5. THE ANT COLONY OPTIMIZATION ALGORITHM

Basically, the general scheme of the ACO algorithm for implementations on static combinatorial optimization problems [8] has been given in Figure 2, once the parameters, as well as pheromone trails, have been initialized. The main loop consists of 3 steps: The first step is that  $m$  ants are constructing solutions for the problem instance under consideration, possibly by the provided heuristic information and biased by the information of the pheromone. When their solutions have been accomplished, perhaps, these might enhance in the phase of an optional local search. Lastly, the pheromone trails are adjusted before proceed to the next iteration in reflecting the search experience of the ants.

```

Set parameters, initialize pheromone trails
while termination condition not met do
    ConstructAntSolutions
    ApplyLocalSearch (optional)
    UpdatePheromones
Endwhile

```

Figure 2. Ant colony optimization algorithm

### 5.1. Construct ant solutions

In order to tackle the problem instance, a solution is constructed by the set of  $m$  ants. Every ant is initialized with an empty solution  $s_p = \emptyset$  in the beginning. Then, an ant updated it extends the partial solution  $s_p$  at every iteration step by selecting one feasible component of the solution  $c_i^j \in N(s_p) \subseteq C$  and adding it into its current solution.  $N(s_p)$  is denotes as a set of solution components that need to be added whilst preserving feasibility and is implicitly defined by the process of solution construction implements by ants. Otherwise, if a partial solution unable to extend whilst preserving feasibility, then, it is dependent on the mechanism of a certain construction. The complete solution is constructed in case the solution construction is infeasible or abandoned. In a most recent case, based on a violation degree of problem constraints, the penalization can be applied to the infeasible solution.

## 5.2. Apply local search

When full candidate solutions have been obtained, they can be further improved by using local search algorithms. Besides, in combination with local search algorithms, ACO algorithms achieve the best performance for a large range of combinatorial optimization problems. In more general, One example of so-called daemon actions is local search. These are employed to centralized actions or adopt problem-specific which impossible for individual ants to perform.

## 5.3. Update pheromones

Generally, the pheromone update is given in the following equation:

$$\tau_{ij} = (1 - p)\tau_{ij} + \sum_{s \in Supd \setminus c_i^j \in s} g(s) \quad (12)$$

where *Supd* denotes as the set of solutions which are employed in order to deposit pheromone,  $p \in [0,1]$  denotes as a parameter known as evaporation rate,  $g(\cdot): S \rightarrow R^+$  denotes as a function such that  $f(s) < f(s') \Rightarrow g(s) \geq g(s')$ . Usually, it is known as the evaluation function and it also can verify the quality of the solution.

The goal of the pheromone update is to obtain the following iterations more suitable for solution components belonging to a better solution. Essentially, this goal can be achieved by two mechanisms. Firstly, a deposit of pheromones, when combined with a selected set  $S_{upd}$  of better solutions able to increases the pheromone level of solution components. Thus, ants become more attractive to these solution components in the subsequent iterations. Secondly, the evaporation of the pheromone trail is a method that reduces through time by the previous ants deposited the pheromone. Pheromone evaporation is essential from a practical point of view to prevent the algorithm converge too rapidly to a sub-optimal area. It applies an effective form of forgetting, favoring the exploration of the new areas of the search space.

# 6. IMPROVEMENT ON ANT COLONY OPTIMIZATION

In recent years with the progress of researches and applications, ant colony optimization (ACO) has been enhanced to improve the efficiency of ACO as explained in the next subsection.

## 6.1. Improvement by applying construct ant solutions

Liu *et al.* [9] have proposed a greedy-Levy ACO in order to solve complicated combinatorial optimization problems. This method employed epsilon greedy as well as Levy flight. Epsilon greedy is one of the important and extensively been applied in reinforcement learning as a policy-based exploration method. Besides, it has been utilized as the ‘pseudo-stochastic mechanism’ to improve the performance of the ACO algorithm. Meanwhile, ‘Levy flight’ is relying on the Levy distribution and also helps accelerate global optimization and balance the searching space. Then, a new ACO algorithm is proposed by Gao [10] which expands the search space for the ants a well as diversifies the solutions searched. This new algorithm integrates pairs of searching ants to enhance solution diversification.

Yi *et al.* [11] employed multiple pheromone change and heuristic functions methods. Moreover, they introduced mutation strategy and adaptive mechanism in order to reduce computational times of the ant colony (AC) algorithm, hasten up the convergence algorithm as well as increased the prediction accuracy. Long *et al.* [12] improved the AC algorithm for the mobile robot in solving complicated map problems, specifically in the shortest path length, improving the speed of convergence and bending the effect of suppression. An improved A\* algorithm estimated function has been employed as the heuristic function in order to enhance the smoothness of the path and search efficiency as well. Furthermore, they improved the global search ability of the algorithm as well as solved the ant deadlock problem by utilized the method of improved MIN-MAX Ant System and retraction mechanism. Whereas Qin *et al.* [13] enhanced the global convergence capability by the introduced rule of ant state transition in order to stimulate ants in the attempt the path which is less searched. Additionally, they improved the searching efficiency with designed a strategy for ant path compression.

## 6.2. Improvement by applying global search

An improved ACO-based method was suggested by Wang *et al.* [14], in order to avoid being stuck at a local optimal solution, to enhance the capability of global search and to improve the rate of convergence.

### 6.3. Improvement by applying local search

A hybrid fuzzy ACO (FACO) algorithm has been proposed by Jia *et al.* [15]. An optimistic coefficient is introduced in this algorithm to handle the processing time of fuzzy which leads to producing an algorithm more stable and feasible in a fuzzy scenario. Moreover, two candidates are integrated to reduce the search space which results in improving the searching efficiency. In fact, to further enhance the quality solution, a fuzzy local optimization approach is employed. The bi-objective ACO has been suggested by Jia *et al.* [16] to minimize make span and total energy consumption. In discrete optimization, finding a feasible solution is a very challenging problem owing to the complex constraints in the problem. Thus, an efficient method is provided to create feasible solutions, consequently, in the search space, the AC only requires focusing on the promising area. In fact, to construct the solutions, the user's preferences are incorporated. In addition, to improve the solutions, a neighborhood-based local optimization is utilized ensuring that the capabilities of AC's exploration and exploitation can be adequately exerted. Chen *et al.* [17] proposed the sequential variable neighborhood descent integrated with a memetic algorithm based on ACO. Moreover, to solve the bi-objective multiple TSP, a local search procedure for bi-objective problems has been proposed and implemented. Thammano and Onnsrikaw [18] improved the strength of the ACO and employed one or other of four algorithms such as simulated annealing (SA), similarity measure SA (Sim-SA), 2-Opt and 3-Opt to find near-optimal solutions and improving local search. Guan *et al.* [19] suggested an ACO with a local search for the bandwidth minimization problem on graphs. The primary novelty for this algorithm is the utilization of an efficient local search to increase ACO's intensification ability. The local search is integrated with the first and best improvement strategy. The first improvement employed hill climbing method in finding better solutions till the current solution unable to be further improved. The best improvement is then performed a specified number of iterations for visiting a new search space.

### 6.4. Improvement by applying global search and local search

Srichandum and Pothiya [20] have suggested an improved ant colony optimization (IACO) in order to ensure the diversity of ants. The additional procedures include a particular improvement of the algorithm (known as neighborhood searches) and re-initializations. In the searching process, the solution being repeated for a long time. This means that the process failed to escape with this solution or unable to find better solutions. Consequently, this solution is possible either locally or globally. The search process will be interrupted if the global solution is known because the best solution has been found. In contrast, if the global solution is unknown, then, the algorithm assumes it is a local solution. Meanwhile, if the process is a blow to the local solution in a very long time, the process of re-initialization is utilized. Subsequently, this mechanism helps the search process to continue and find better solutions. Later, Hong *et al.* [21] improved the AC algorithm that employed the internet of things (IoT) to crawl and collect cluster information. Based on the trajectory information of AC, For the cluster resource indexing of the IoT, the location and velocity are updated and the balanced AC algorithm is used to perform the global and local search for resources and initialise the clustering center, and the goal function is constructed for the cluster resource indexing of IoT and the problem of the optimization parameter has been solved by the constraint condition of the minimum variance of the whole fitness. The strong global optimization capability of the AC algorithm is utilized in order to realize resource indexing optimization.

### 6.5. Improvement by applying pheromones update

Ding *et al.* [22] suggested IMCA-BS (Improved AC algorithm-based band selection) algorithm in order to solve the drawbacks of the conventional AC algorithm. In the IMACA-BS, the initialization of pheromone on each path is based on the heuristic desirability amongst each pair of nodes with the means of a pre-filter. The information amongst the potential node and all the preferred nodes is employed during the searching process in order to lead a search ant to perform a state transition with the pseudo-random rule. Besides, an adaptive information strategy is proposed to avoid the searching ants leaving the pheromone with just several (instead of all) optimal paths. The proposed IMACA-BS avoids the AC being trapped in local optima by accelerating the speed of convergence as well as maintaining the diversity of the population.

Luo *et al.* [23] proposed an improved pheromone method, which calculates the initial pheromone based on the relative distance between the current node, next node and the starting to ending point connection. The initial pheromone with unequal allocation has been produced to prevent the blindness search at early planning. For path selection, a pseudo-random transition rule is applied, the probability of state transition is being calculated based on the number of iterations and current optimal solution, the determination of proportion or random selections is changing adaptively. Besides, the worst solution and optimal solution are suggested in improving the global pheromone update approach. Moreover, the dynamic punishment method is suggested as well in solving the deadlock problem. This study intends in solving the problems including low search efficiency, slow speed of convergence and local optimum in the AC algorithm.

Deng *et al.* [24] introduced ICMPACO (new multi-population co-evolution ACO) which based on co-evolution mechanism, multi-population strategy, pheromone diffusion mechanism and pheromone updating strategy in order to balance the solution diversity and speed of convergence as well as enhance the performance of optimization to solve the large scale optimization problem. Besides, in the ICMPACE algorithm, the optimization problem is segregated into numerous sub-problems, meanwhile, the ants in the population also segregated into common ants and elite ants to avoid stuck at local optimal and enhance the rate of convergence. Furthermore, in order to improve the ability of optimization ability, the pheromone updating strategy is utilized. Whilst to make ants released the pheromone at a certain point, the pheromone diffusion mechanism is employed. This affects gradually a certain range of adjacent regions. Also, the co-evaluation mechanism is employed to exchange information between several sub-populations in implementing information sharing.

Deng *et al.* [25] proposed a new pheromone updating rule, known as SPB (sub-path-based pheromone modification) 'rule and suggested an improved ACO based on the SPB rule, naturally named' as SPB-ACO. The SPB rule comprises an adaptive part and an iterative updating part. The adaptive part regulates the updating strength by the gradual iteration. Meanwhile for the iterative updating part is utilized to update certain selected pheromone of sub-paths trace, known as r-rank sub-paths chosen with calculating the crossed ants of a sub-path in each iteration. Subsequently, with the combination of the iteration-best update rule and SPB rule, an adaptive SPB ACO is applied to solve the standard of TSP.

Qian *et al.* [26] suggested an improved ACO, which enhanced the pheromone update method and the heuristic function of the ACO algorithm. In the heuristic function, the influence factor of path distance is proposed. The algorithm search has some direction, in the meantime, to enhance the speed of convergence of the algorithm. Furthermore, the pheromone's penalty mechanism is employed and it merged with the factor of self-adapted volatility. Table 2 tabulated the summarization of improvement ACO optimization.

Table 2. Summary of improvement in ACO optimization

No.		Proposed Method	Comments
1.	Improvement on construct ant solution	<ul style="list-style-type: none"> <li>- Greedy-Levy ACO [9].</li> <li>- A new ACO algorithm [10].</li> <li>- Employed multiple pheromone change and heuristic function methods [11].</li> <li>- Improved AC algorithm for the mobile robot [12].</li> <li>- Enhanced global convergence capability [13].</li> </ul>	<ul style="list-style-type: none"> <li>• Helps accelerate global optimization and balance the searching space.</li> <li>• Expands the search space for the ants a well as diversifies the solutions searched.</li> <li>• Reduce computational times of the ant colony (AC) algorithm, hasten up the convergence algorithm as well as increased the prediction accuracy.</li> <li>• Improving the speed of convergence and bending the effect of suppression.</li> <li>• Enhance the smoothness of the path and search efficiency.</li> <li>• Improved the global search ability of the algorithm as well as solved ant deadlock problem</li> <li>• Stimulate ants in an attempt at a path that is less searched.</li> <li>• Improved the searching efficiency with designed a strategy for ant path compression</li> </ul>
2.	Improvement on global search	<ul style="list-style-type: none"> <li>- An improved ACO-based method [14].</li> </ul>	<ul style="list-style-type: none"> <li>• Avoid being stuck at the local optimal solution, to enhance the capability of global search and to improve the rate of convergence.</li> </ul>
3.	Improvement of apply local search	<ul style="list-style-type: none"> <li>- A hybrid Fuzzy ACO (FACO) algorithm [15].</li> <li>- The bi-objective ACO [16].</li> <li>- The sequential variable neighborhood descent is integrated with a memetic algorithm based on ACO [17].</li> <li>- Improved the strength of the ACO [18].</li> <li>- ACO with local search for the bandwidth minimization problem on graphs [19].</li> </ul>	<ul style="list-style-type: none"> <li>• Reduce the search space which results in improving the searching efficiency.</li> <li>• Minimize makespan and total energy consumption.</li> <li>• Solve the bi-objective multiple TSP.</li> <li>• Improved the strength of the ACO.</li> <li>• Find near-optimal solutions and improving local search.</li> <li>• Increase ACO's intensification ability.</li> </ul>
4.	Improvement of global search and apply local search	<ul style="list-style-type: none"> <li>- Improved ant colony optimization (IACO) [20].</li> <li>- Improved AC algorithm that employed the Internet of Things to crawl and collect cluster information [21].</li> </ul>	<ul style="list-style-type: none"> <li>• Ensure the diversity of ants.</li> <li>• Helps the search process to continue and find better solutions.</li> <li>• Realize resource indexing optimization.</li> </ul>

Table 2. Summary of improvement in ACO optimization (*continue*)

No.	Proposed Method	Comments
5.	Improvement of update pheromones	
	- IMCA-BS (Improved AC algorithm-based band selection) algorithm [22].	<ul style="list-style-type: none"> <li>• Avoid the AC being trapped in local optima by accelerating the speed of convergence as well as maintaining the diversity of the population.</li> </ul>
	- An improved pheromone method [23].	<ul style="list-style-type: none"> <li>• Solving the problems including low search efficiency, slow speed of convergence and local optimum in the AC algorithm.</li> </ul>
	- ICMACO (New multi-population co-evolution ACO) [24].	<ul style="list-style-type: none"> <li>• Balance the solution diversity and speed of convergence.</li> <li>• Solve the large-scale optimization problem.</li> <li>• Avoid stuck at local optimal and enhance the rate of convergence.</li> </ul>
	- SPB-ACO [25].	<ul style="list-style-type: none"> <li>• Improve the ability of optimization ability.</li> <li>• Solve the standard of TSP.</li> </ul>
	- An improved ACO, which enhanced the pheromone update method and the heuristic function of the ACO algorithm [26].	<ul style="list-style-type: none"> <li>• Enhance the speed of convergence of the algorithm.</li> </ul>

## 7. CONCLUSION

In this research work, first, we gave a brief description of the graph and graph theory. After that, we defined graph-theoretic (GT) carving and highlighted the main shortcoming in GT. We then gave a further explanation of the optimal path. In order to find an optimal solution, optimization provides a better solution. Furthermore, we highlighted three main factors in optimization. Therefore, it is so important in selecting a good optimization method. One of the metaheuristic optimization techniques is Ant Colony Optimization (ACO). We outlined the general framework of the ACO algorithm. For this reason, it gives us the idea to propose ACO based on GT based carving. Moreover, we classified the improvement of ACO based on the general scheme of the ACO optimization for implementations on static combinatorial optimization problems. In future works, we intend to integrate ACO with file carving techniques in carving files such as office documents, emails and images. This technique will give an optimal solution to solve fragmentations.

## REFERENCES

- [1] M. A. Saare, A. B. Ta'a, S. A. Lashari, and S. A. Sari, "Mobile system for managing and mitigating the accommodation problems," in *Journal of Physics: Conference Series*, vol. 1019, 2018.
- [2] S. A. Sari and K. M. Mohamad, "A review of graph theoretic and weightage techniques in file carving," in *Journal of Physics: Conference Series*, vol. 1529, 2020.
- [3] M. Maqsood, S. A. Lashari, M. A. Saare, S. A. Sari, Y. M. Hussein, and H. O. Hatem, "Minimization response time task scheduling algorithm," in *IOP Conference Series: Materials Science and Engineering*, vol. 705, 2019.
- [4] T. Dokeroglu, E. Sevinc, and Ahmed Cosar, "Artificial bee colony optimization for the quadratic assignment problem," *Applied Soft Computing*, vol. 76, pp. 595-606, 2019.
- [5] Y. Wu, M. Gong, W. Ma, and S. Wang, "High-order graph matching based on ant colony optimization," *Neurocomputing*, vol. 328, pp. 97-104, 2019.
- [6] J.-L. Deneubourg, S. Aron, S. Goss, and J. M. J. J. o. i. b. Pasteels, "The self-organizing exploratory pattern of the argentine ant," *Journal of Insect Behavior*, vol. 3, no. 2, pp. 159-168, 1990.
- [7] S. Goss, S. Aron, J.-L. Deneubourg, and J. M. J. N. Pasteels, "Self-organized shortcuts in the Argentine ant," *Naturwissenschaften*, vol. 76, pp. 579-581, 1989.
- [8] M. Dorigo and T. Stützle, "Ant colony optimization: overview and recent advances," in *Handbook of metaheuristics: Springer*, pp. 311-351, 2019.
- [9] Y. Liu, B. Cao, and H. Li, "Improving ant colony optimization algorithm with epsilon greedy and Levy flight," *Complex & Intelligent Systems*, 2020.
- [10] W. Gao, "New ant colony optimization algorithm for the traveling salesman problem," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 44-55, 2020.
- [11] N. Yi, J. Xu, L. Yan, and L. Huang, "Task optimization and scheduling of distributed cyber-physical system based on improved ant colony algorithm," *Future Generation Computer Systems*, vol. 109, pp. 134-148, 2020.
- [12] S. Long, D. Gong, X. Dai, and Z. Zhang, "Mobile robot path planning based on ant colony algorithm with A\* heuristic method," *Frontiers in neurorobotics*, 2019.
- [13] W. Qin, J. Zhang, and D. Song, "An improved ant colony algorithm for dynamic hybrid flow shop scheduling with uncertain processing time," *Journal of Intelligent Manufacturing*, vol. 29, no. 4, pp. 891-904, 2018.
- [14] J. Wang, J. Cao, R. S. Sherratt, and J. H. Park, "An improved ant colony optimization-based approach with mobile sink for wireless sensor networks," *Journal of Supercomputing*, vol. 74, no. 12, pp. 6633-6645, 2018.
- [15] Z. Jia, J. Yan, J. Y. Leung, K. Li, and H. Chen, "Ant colony optimization algorithm for scheduling jobs with fuzzy processing time on parallel batch machines with different capacities," *Applied Soft Computing*, vol. 75, pp. 548-561, 2019.

- [16] Z.-h. Jia, Y. Wang, C. Wu, Y. Yang, X.-y. Zhang, and H.-p. Chen, "Multi-objective energy-aware batch scheduling using ant colony optimization algorithm," *Computers Industrial Engineering*, vol. 131, pp. 41-56, 2019.
- [17] X. Chen, P. Zhang, G. Du and F. Li, "Ant colony optimization based memetic algorithm to solve bi-objective multiple traveling salesmen problem for multi-robot systems," in *IEEE Access*, vol. 6, pp. 21745-21757, 2018.
- [18] A. Thammano and Y. Oonsrikaw, "Improved ant colony optimization with local search for traveling salesman problem," *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, Toyama, Japan, pp. 22-27, 2019.
- [19] J. Guan, G. Lin, and H.-B. Feng, "Ant colony optimisation with local search for the bandwidth minimisation problem on graphs," *International Journal of Intelligent Information Database Systems*, vol. 12, no. 1-2, pp. 65-78, 2019.
- [20] S. Srichandum and S. Pothiya, "Multiple plants multiple sites ready mixed concrete planning using improved ant colony optimization," *International Journal of GEOMATE*, vol. 19, no. 72, pp. 88-95, 2020.
- [21] Y. Hong, L. Chen, and L. Mo, "Optimization of cluster resource indexing of Internet of Things based on improved ant colony algorithm," *Cluster Computing*, vol. 22, no. 3, pp. 7379-7387, 2019.
- [22] X. Ding, H. Li, J. Ynag, P. Dale, X. Chen, C. Jiang, and S. Zhang, "An improved ant colony algorithm for optimized band selection of hyperspectral remotely sensed imagery," *IEEE Access*, vol. 8, pp. 25789-25799, 2020.
- [23] Q. Luo, H. Wang, Y. Zheng, and J. He, "Research on path planning of mobile robot based on improved ant colony algorithm," *Neural Computing Applications*, vol. 32, no. 6, pp. 1555-1566, 2020.
- [24] W. Deng, J. Xu and H. Zhao, "An Improved ant colony optimization algorithm based on hybrid strategies for scheduling problem," in *IEEE Access*, vol. 7, pp. 20281-20292, 2019.
- [25] X. Deng, L. Zhang, and J. Feng, "An Improved ant colony optimization with subpath-based pheromone modification strategy," in *International Conference on Swarm Intelligence, Springer*, pp. 257-265, 2017.
- [26] W. Qian, L. Zhou, L. Yang and W. Xu, "An improved ant colony algorithm of three dimensional path planning," *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, pp. 119-122, 2017.

## BIOGRAPHIES OF AUTHORS



**Sari Ali Sari** is a Ph.D candidate at Universiti Tun Hussein Onn Malaysia (UTHM), Malaysia. He holds a master's degree in computer science (internetworking technology) at Universiti Teknikal Malaysia Melaka (UTeM), Malaysia. His research interest includes Metaheuristic, File carving, Digital Forensics.



**Dr. Kamaruddin Malik bin Mohamad** is a senior lecturer in UTHM. His research interest includes File Carving, Steganography, Secure Data Wiping, Digital Forensics Triage, Digital Forensic Analysis, Metadata Visualization and Data Redact.