International Journal of Informatics, Information System and Computer Engineering

# Impact of Number of Artificial Ants in ACO on Network Convergence Time: A Survey

*Samuel W Lusweti[1*], Collins O Odoyo, Dorothy A Rambim*

Department of Information Technology, School of Computing and Informatics, Masinde Muliro University of Science and Technology, Kenya
*Corresponding Email: lusweti015@gmail.com

## ABSTRACTS

Due to the dynamic nature of computer networks today, there is need to make the networks self-organized. Self-organization can be achieved by applying intelligent systems in the networks to improve convergence time. Bio-inspired algorithms that imitate real ant foraging behaviour of natural ants have been seen to be more successful when applied to computer networks to make the networks self-organized. In this paper, we studied how Ant Colony Optimization (ACO) has been applied in the networks as a bio-inspired algorithm and its challenges. We identified the number of ants as a drawback to guide this research. We retrieved a number of studies carried out on the influence of ant density on optimum deviation, number of iterations and optimization time. We found that even though some researches pointed out that the numbers of ants had no effect on algorithm performance, many others showed that indeed the number of ants which is a parameter to be set on the algorithm significantly affect its performance. To help bridge the gap on whether or not the number of ants were significant, we gave our recommendations based on the results from various studies in the conclusion section of this paper.

## ARTICLE INFO

***Article History:***
*Received 25 May 2022*
*Revised 30 May 2022*
*Accepted 10 June 2022*
*Available online 26 June 2022*

***Keywords:***
***Convergence time,***
***Ant Colony Optimization,***
***artificial ants,***
***networks,***
***parameter***

## 1. INTRODUCTION

A family of optimization techniques that have been applied as combinatorial problem-solving techniques form the widely known metaheuristics. Over the years, metaheuristics have been applied in many fields to solve complex problems (Liao et al., 2014). The ACO algorithm is one example of these metaheuristics. Others include Particle Swarm Optimization, Bee Colony Optimization, Bat Optimization among others. ACO was introduced in the early years of the 1990s (Blum, 2005) and has for many years been applied in various fields to tackle complex optimization issues that may be solved using basic methods. For example, the algorithm has been successfully applied in data compression, gaming theory, feature selection, dispatch problems, parameter estimation in dynamic systems, satellite control, job scheduling problems, congestion control, social graph mining, medicine for decision making, and target tracking (Kar, 2016). ACO has also been employed in computer networks to determine the shortest possible route for sending data from basis to destination, similar to how ants forage (Adiga et al., 2013), therefore becoming the lowest-cost route between any two connected nodes. This has increased network functioning while decreasing the latency that packets suffer in the absence of the ACO algorithm, when attempting to reach their target because computer networks are important to the running of any institution, they must be active at all times, regardless of the obstacles they may encounter. As a result, technical adjustments in their upkeep are required. ACO was employed in this example to make the network self-regulating and sustainable, so that any issues discovered in the network could be addressed by the network itself. Some academics have proposed using ACO to prevent the need for infrastructure such as nodes and switches, which might fail and cause the communication process to fail (Dressler et al., 2010).

## 2. METHODS AND MATERIALS

In this paper we adopted both qualitative research methods. Secondary data of 12 experiments published online was retrieved whereby results from various studies were compared and used to do the analysis using thematic analysis.

## 3. APPLICATION OF ACO IN COMPUTER NETWORKS

Ant Colony Optimization algorithm is a class of swarm intelligent systems that are applied in solving NP-hard problems. Swarm-intelligent systems have not been fully explored in literature till today (Gustov & Levina, 2021), but a lot is being done in ACO. This is because among the many Swarms Intelligent algorithms available, ACO is the most studied. Foraging ants in a true ant colony system produce pheromone trails along the pathways they traverse, which others may follow to the food source. An ant in the ACO algorithm is a mobile agent capable of dealing with computer network difficulties such as congestion and packet routing. This is made possible by the continuous and consistent modification of the routing tables by the agents in response to any congestion in the network (Sim & Sun, 2002). As the agents do this modification, they lay pheromone trails, which make clear routes between any nodes on the system. These pheromone trails acting as stigmergy would be handy in helping the future ants make a routing decision (Kamali & Opatrny, 2007). Other bio-inspired algorithms that have been successfully applied in computer networks for adaptive routing include the Artificial Plant Optimization (APO) algorithm for the implementation of the telecom sensor network, Artificial Neural Networks (ANN) for switching networks, the Genetic Algorithm (GA) for network path routing and the Leaping Frog Algorithm (LFA) for network designing and scaling problems (Kar, 2016). However, ACO has been the most often used and researched optimization algorithm since it has shown the highest performance and has been the most effective (Caro & Dorigo, 2004). As a result, this research concentrated on the design and use of ACO in many sectors, and more especially on its application in computer networks, and how it may even be improved in terms of the ideal number of ants required to be employed for better performance in these networks. Forward and backward ants are used in these algorithms (Jacobsen et al., 2011). The proactive ACO routing algorithm ANTNET has been used effectively in packet-switched networks (Caro & Dorigo, 1998). The ANTNET algorithm propels a forward ant from the nest or source node at regular intervals towards its objective of food. When a forwarding ant reaches at its destination, it uses the list to return to the nest or source and update the pheromone values deposited in the routes or connections. If ACO is applied in an ideal network, the ants are translated into packets, and the routes they use become the links between the nodes on that very network. If we have redundant links between the devices on the network, then the packets are expected to go through the shortest route to the destination if they are well optimized.

## 4. DRAWBACKS AND VARIATIONS OF ACO

The ACO has shown certain benefits, such as positive feedback for quick solutions, dynamic applications that adjust to changes such as additional distances, and intrinsic parallelism. It has, however, shown several limitations, such as probability distribution changes due to iterations and convergence time (Selvi & Umarani, 2010). More precisely, ANTHOCNET's disadvantage is the quantity of routing messages routing that must be delivered in the network before the formation of routes to the destination, the downside of ANTNET is the time necessary to build a route system between any two nodes in the network. This is referred to as the convergence time (Selvi & Umarani, 2010). Last but not least the stagnation of ants in the working process of the algorithm (Caro & Dorigo, 1998) is also another problem common in ACO algorithms. Parameter setting of a basic ant colony algorithm is mainly the cause of these variations and is still under experimental stage till today (Wei, 2014).

### 4.1 ACO Algorithm Parameter Setting

The following are the parameters under consideration (Wei, 2014).

m - number of ants.

α - pheromone relative importance.

β - relative importance of heuristic factor.

ρ- pheromone evaporation coefficient while (1-ρ) indicates the pheromone persistence factor.

Q - amount of pheromone released by ants

The following is the general formula used in the algorithm with the above parameters. The ants would independently select the next town or city to tour at time t, hence the probability of ant k to move from city i to j is given by (Caro & Dorigo, 1998):

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}(t)]^{\beta}}{\sum\limits_{x \in J_k(i)} [\tau_{ix}(t)]^{\alpha} \cdot [\eta_{ix}]^{\beta}}, & \text{if } j \in J_k(i) \\ 0, & \text{otherwise} \end{cases}$$

After all the ants have toured all the cities in the search space, each of the paths is updated according to Eq. (2) below.

$$\tau_{ij}(t+n) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}$$

Were,

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

and,

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k}, \text{if ant k passed the edge of ij in this tour} \\ 0, else \end{cases}$$

This study compares various researches on the impact of number of ants (m) shown in Eq. (3) on optimal solution and convergence time of the algorithm.

## 5. THE EFFECT OF THE NUMBER OF ANTS (M) ON ACO OPTIMIZATION

According to (Colorni et al., 1991) the number of ants is among the controllable parameters affecting the performance of ACO. We examine at 12 tests from three research to see how the number of ants in the ACO algorithm affects its convergence. Here, we look at whether or not an optimal solution is found and how long it takes for the solution to be found.

### 5.1. Experiment 1

This experiment was done by Aydin and Yilmaz (Sivagaminathan & Ramakrishnan, 2007). The two presented an investigation into the number of ants used in ACO in relation to the number of iterations, penalized objective function, and optimization time. For the purposes of this study, the results obtained from the number of iterations as well as time of optimization versus the number of ants are taken into account (See Figs. 1).

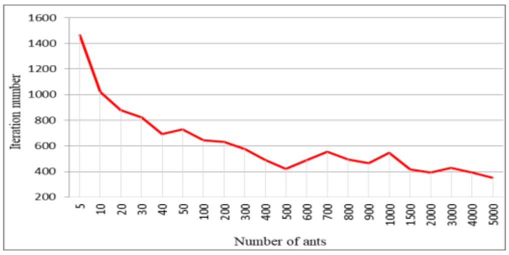

**Fig. 1. The Average Iteration Number Versus Number of Ants** (Sivagaminathan & Ramakrishnan, 2007)

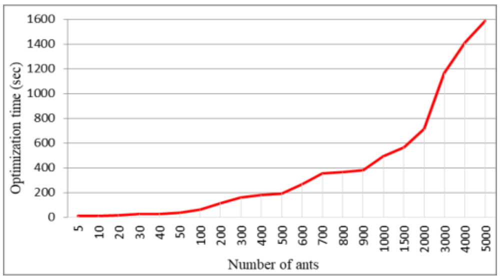

**Fig. 2. The Optimization Time Versus Number of Ants** (Sivagaminathan & Ramakrishnan, 2007)

From Fig. 1 above, as the number of ants rises, the number of iterations reduces. In contrast to Fig. 2 above, the optimization time increases quickly when the population of ants grows. In terms of the precise number of ants required for the optimum solution, this research does not give a direct answer, as it only suggests that it is critical to identify the appropriate number of ants in order to get the best solution in the shortest amount of time. However, from the experimental results in fig. 1 and 2 above, it is obvious that the fewer the ants, the greater the number of iterations, and hence the shorter the optimization time. In other terms, when the number of ants rises, the number of iterations decreases yet the optimization time increases since numerous ants take longer to converge. However, this experiment was limited to small problems, and the exact number of optimal ants could not be established clearly.

## 5.2. Experiment 2

We examine this experiment done by Alobaedy et al (Yilmaz & Aydin). In this experiment, the researchers categorized optimization problems into small and medium scales using data sets of 50 and 100 cities respectively. They were able to measure the execution time, best solution, total number of new solutions obtained among other metrics.  Fig. 3 below shows the results obtained in terms of execution time against the number of ants for small scale problem of 50 cities. Figure 3 shows that increasing the number of ants causes an increase in the execution time.

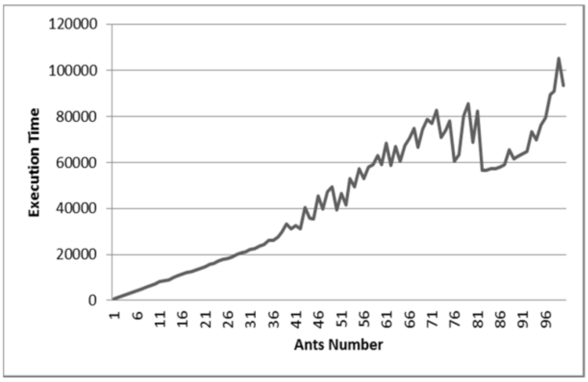

**Fig. 3. Execution Time Against the Number of Ants (50 Cities)** (Yilmaz & Aydin)

When the experiment with 100 ants (medium) was run as shown below in Fig. 4, a similar pattern was noted. However, due to the complexity of this issue, the execution time of the method increased in fig. 4.

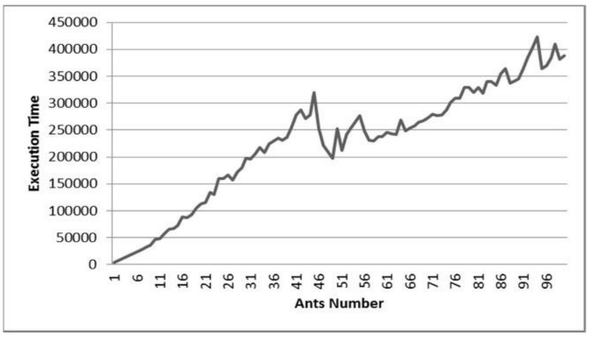

**Fig. 4. Execution Time Against the Number of Ants (100 Cities)** (Yilmaz & Aydin)

This research reveals that increasing the number of ants did not improve the algorithm's performance, but if the number is low, the performance was enhanced. Nevertheless, the experiment was performed under two problem variations that is small and medium sized problems. For small sized problem the execution time was small and difficult to determine the exact number of ants needed to optimize the solution. Whereas on the medium sized problem, the execution time doubled and the number of optimum ants for that solution was found to be 16 out of 100.

### 5.3. Experiment 3

In this experiment, Christoffer and Lars (Alobaedy et al., 2017; Stutzle & Hoos, 2000; Aydin) carried out comparisons on three ACO variations model (RankedAS EliteAS, and MMAS). The EliteAS relies on specialist ants to work. These secondary ants, or specialized ants, are utilized to impose the elitist approach. The other ants called the normal ants work differently. The specialist ants multiply the pheromone on the best solution found by normal ants (Petterson & Lundell, 2018) making it stay longer without decomposing unlike the normal routes in other ant systems. The RankedAS on the other hand also use specialist ants but these ants deposit pheromones on many good paths found instaed of depositing all pheromones on the best solution found (Petterson & Lundell, 2018). Every route is graded by length, with the best-ranked route getting the most pheromones and the worst-ranked route receiving the fewest. Lastly, the MMAS has no specialist ants which means it only uses the normal ants. Here, the pheromone deposited on a given path can never exit a maximum value or getlower than a given minimum value. This ensures that the pheromone level on a path does not get too low that the path is rendered unusable or the path should never be filled with the pheromone so much that it overshadows all the other routes (Bullnheimer et al., 1997). This happens through smoothing of the edges whenever pheromone concentration levels are going below or above the extremes (See Figs. 5-9).

### 5.3.1 EliteAS

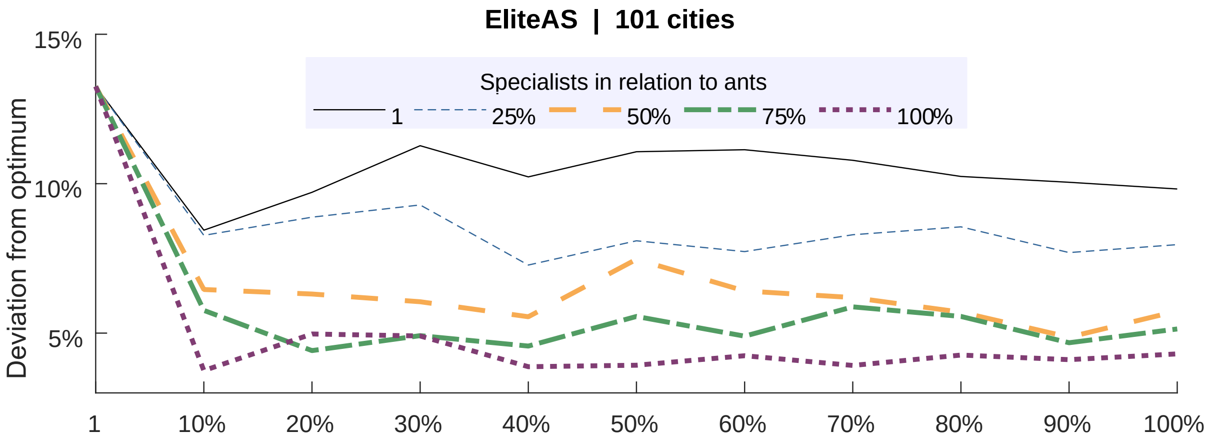

**Fig 5. Average Deviation from Optimum Ants in relation to 101 Cities** (Alobaedy et al., 2017)

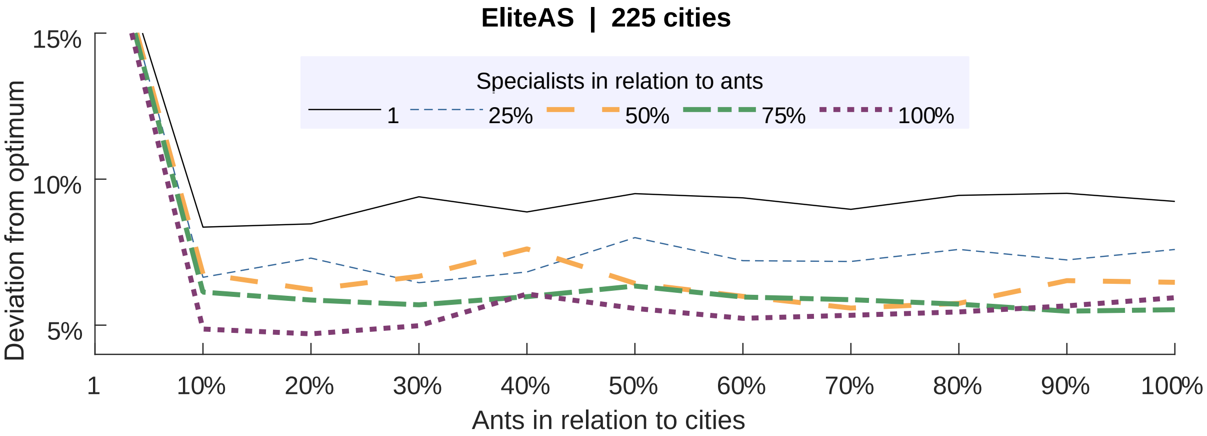

**Fig 6. Average Deviation from Optimum Ants in relation to 225 Cities** (Alobaedy et al., 2017)

In EliteAS as shown in figs. 5 and 6, only between 10-30% of the ants showed a better performance.

### 5.3.2. RankedAS

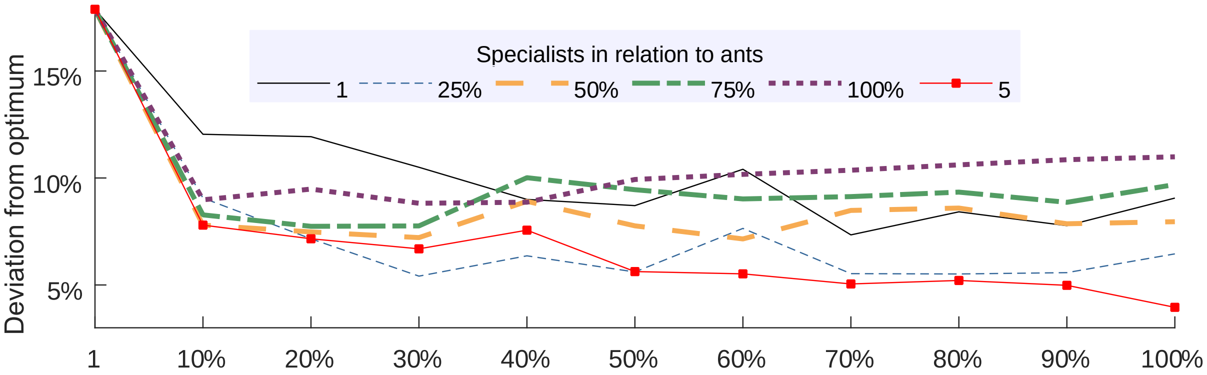

**Fig 7. Average Deviation from Optimum Ants in relation to 101 Cities** (Alobaedy et al., 2017)

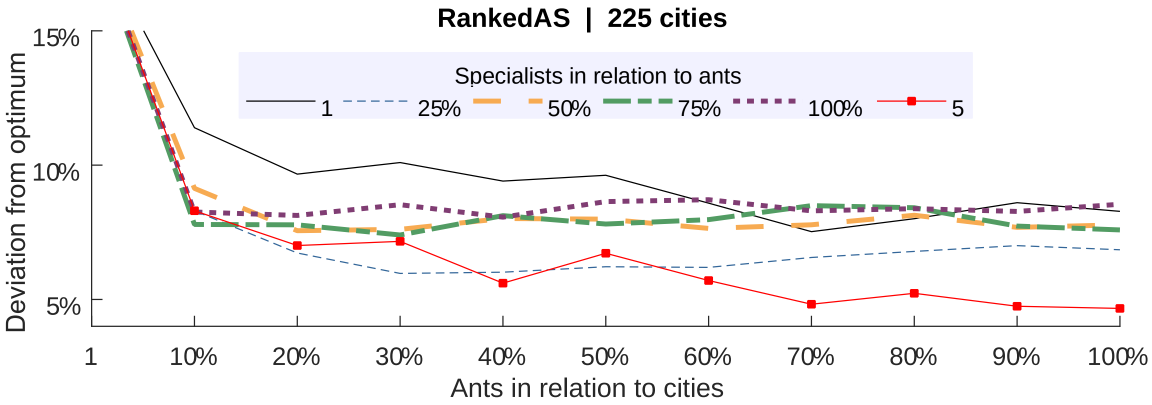

**Fig 8. Average Deviation from Optimum Ants in relation to 225 Cities** (Alobaedy et al., 2017)

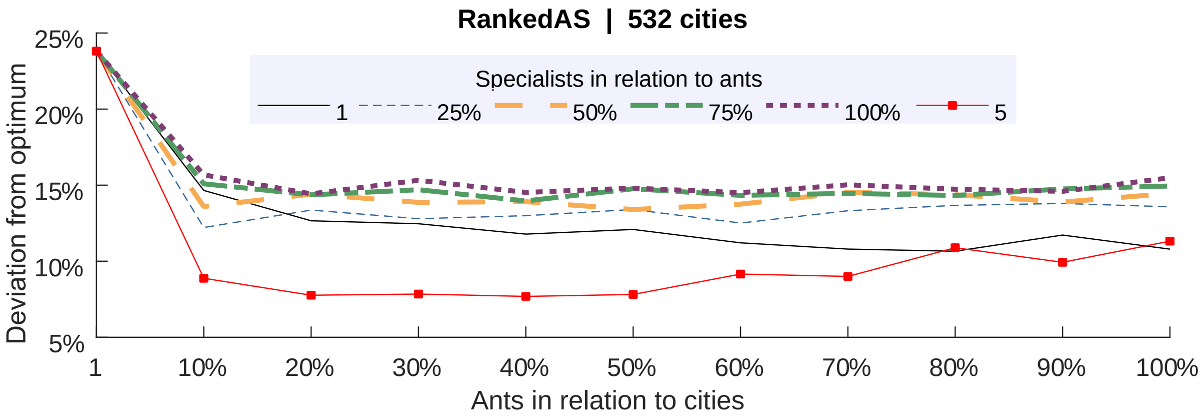

**Fig 9. Average Deviation from Optimum Ants in relation to 532 Cities** (Alobaedy et al., 2017)

As shown from the results in figs. 5,6,7,8 and 9 above, the results of RankedAS and EliteAS are contrasting each other. A large percentage of specialized ants degrades the solution more than a low proportion of specialized ants. In RankedAS, the number of normal ants has an effect of deviation from optimum solution when using 5 specialized ants but when using more specialized ants there is no effect. In this case, the optimal solution is obtained when 5 specialized ants and 100% regular ants are used in relation to cities (Alobaedy et al., 2017). Nonetheless, with figure 9 where we have 525 cities, using more than 50% of the normal ants brings about worse results as the deviation from optimum is high as shown by the red line. It found that, when implementing some RankedAS, the highest number of normal ants which is 100% of ants showed better performance in terms of convergence (See Figs. 10-12).

### 5.3.3 Min-Max ant System

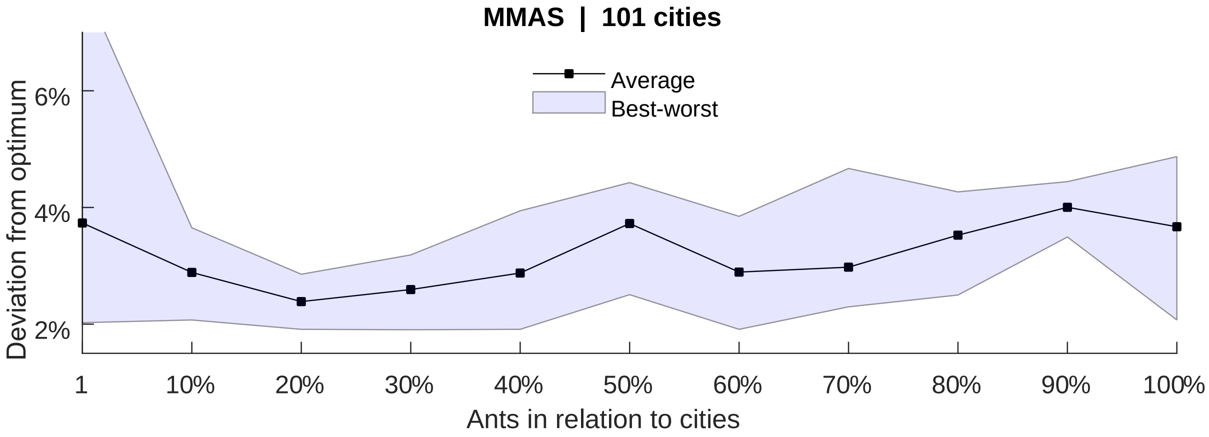

**Fig 10. Average Deviation from Optimum Against Ants in Relation To 101 Cities** (Alobaedy et al., 2017)

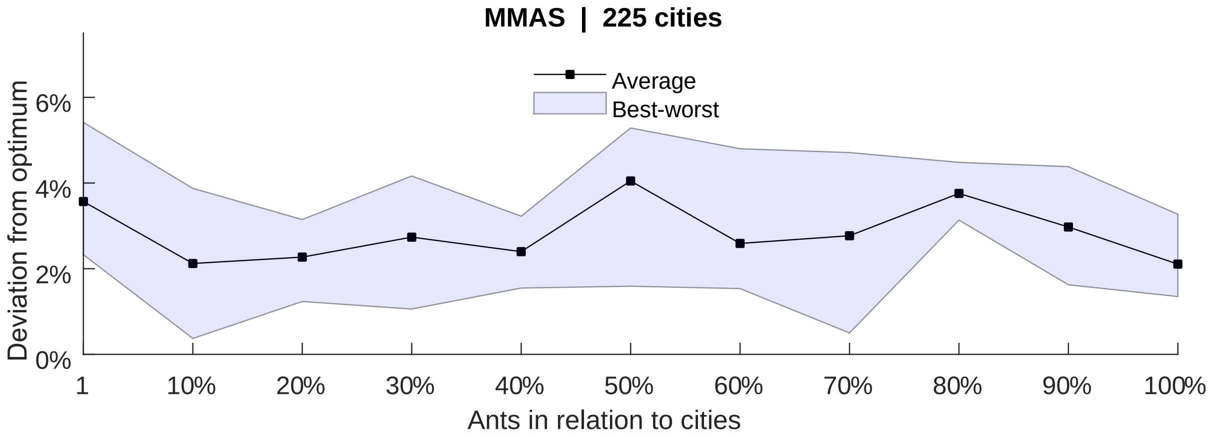

**Fig 11. Average Deviation from Optimum Against Ants in Relation To 225 Cities** (Alobaedy et al., 2017)

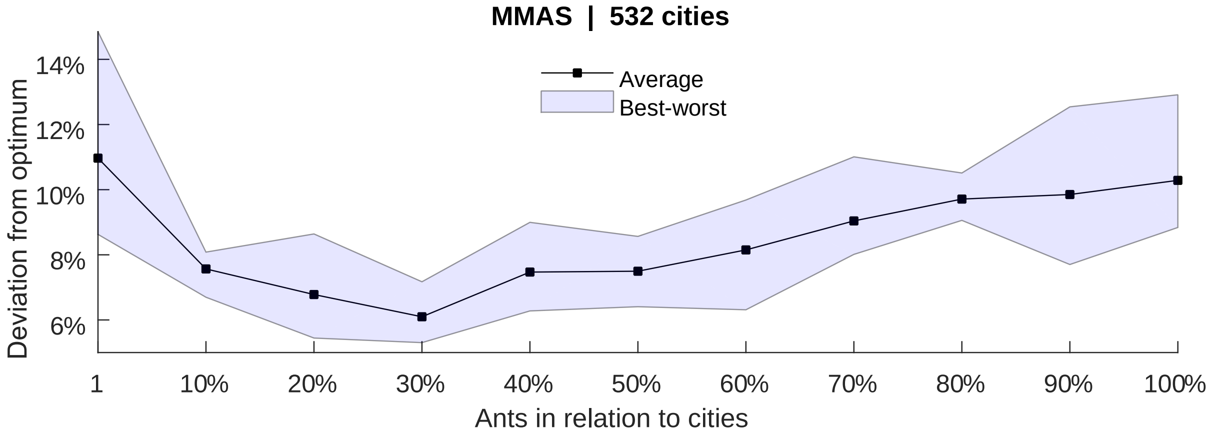

**Fig 12. Average Deviation from Optimum Against Ants in Relation To 532 Cities** (Alobaedy et al., 2017)

Fig. 10 and 11 show that increasing the number of ants in MMAS has no effect on the deviation from optimum results. Except for fig. 12 which shows some variation, overall performance of MMAS has is not affected by a large number of ants (Alobaedy et al., 2017).

## 6. RESULTS AND DISCUSSION

While some eight experiments in the data collected show that an increase in the number of ants degrades the optimization of various ACO versions, two of them indicate that it actually improves the optimization of the algorithm, yet two more reveal that it has no impact on the results of the algorithm.

To help understand the variation in EliteAS, we single out figure 9 that brought about worse results when the number of normal ants is increased. First, we look at the number of specialized ants kept at 5 in all the figures 7, 8 and 9. Calculating the ratio between the number specialized ants to that of normal ants in all the three experiments when the algorithm is at its optimum performance we find; 5:100, 5:225 and 5:266 (at 50% ants) for figures 7,8 and 9 respectively. When we simplify the ratio, we get 1:20, 1:45 and 1:53 respectively. If we take a look at the same figure 9, at 60% ants where the graph starts to deviate from the optimum, the ratio of the ants is 1: 63. From 60%, the graphs exponentially continue to deviate from the optimal solution till 100%. From this data we can see that for the optimal functioning of the algorithm the ratio of specialized ants and that of normal ants has to be considered. In figure 9, this ratio was not considered as the number of specialized ants remained to be 5. For instance, for every 1 specialized ant, there needs to be about 20 to about 50 normal ants to optimize the solution. When the ACO algorithms are optimized, they can then be applied in various fields of study. In this case when it is applied in computer networks, the packets would be translated into ants and made to be as intelligent as the ants of the algorithm, hence prevent packet looping and many other problems associated with unoptimized computer networks especially under dynamic situations. This helps improve on the network convergence time without making the time too short to cause premature convergence or too long to bring a lot of latencies in the network.

## 5. CONCLUSION

After identifying the main challenges faced by ACO which include stagnation of ants, actual number of routing messages that are needed, and convergence time and their main causes which are associated with parameter setting in the algorithm, we singled out one of the parameters which is the convergence time influenced by the number of ants in the solution space. However, the key problem is determining the ideal number of ants to utilize in the algorithm. It is difficult to quantify the number of ants necessary to solve a issue. First, some problems are less complex than others that means they need a smaller number of ants to be solved, and secondly, we have different types of ACO algorithms working differently. A well optimized algorithm will have a short convergence time. We therefore considered the results from the experiments done by various researchers as shown in the 12 figures above and came up with the following conclusions that can help determine the optimum number of ants needed. 1) The type of ACO algorithm in use, 2) The complexity of the problem under study, and 3) The ratio of the number of specialized ants to that of normal ants. Further research needs to be carried out to determine how best we can calculate and determine the optimal number of ants needed.

## REFERENCES

Liao, T., Stützle, T., de Oca, M. A. M., & Dorigo, M. (2014). A unified ant colony optimization algorithm for continuous optimization. *European Journal of Operational Research*, *234*(3), 597-609.

Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of Life reviews*, *2*(4), 353-373.

Kar, A. K. (2016). Bio inspired computing–a review of algorithms and scope of applications. *Expert Systems with Applications*, *59*, 20-32.

Adiga, C. S., Joshi, H. G., & Harish, S. V. (2013). Network routing problem-A simulation environment using Intelligent technique. *International Journal of Advanced Computer Research*, *3*(4), 166.

Dressler, F., & Akan, O. B. (2010). A survey on bio-inspired networking. *Computer networks*, *54*(6), 881-900.

Gustov, V., & Levina, A. (2021, April). Electromagnetic Fields as a Sign of Side-Channel Attacks in GSM Module. In *2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS)* (pp. 1-5). IEEE.

Sim, K. M., & Sun, W. H. (2002, November). Multiple ant-colony optimization for network routing. In *First International Symposium on Cyber Worlds, 2002. Proceedings.* (pp. 277-281). IEEE.

Kamali, S., & Opatrny, J. (2007, March). Posant: A position-based ant colony routing algorithm for mobile ad-hoc networks. In *2007 Third International Conference on Wireless and Mobile Communications (ICWMC'07)* (pp. 21-21). IEEE.

Kar, A. K. (2016). Bio inspired computing–a review of algorithms and scope of applications. *Expert Systems with Applications*, 59, 20-32.

Di Caro, G., & Dorigo, M. (2004). *Ant colony optimization and its application to adaptive routing in telecommunication networks* (Doctoral dissertation, PhD thesis, Faculté des Sciences Appliquées, Université Libre de Bruxelles, Brussels, Belgium).

Jacobsen, R. H., Zhang, Q., & Toftegaard, T. S. (2011). Bioinspired principles for large-scale networked sensor systems: An overview. *Sensors*, *11*(4), 4137-4151.

Caro, G. D., & Dorigo, M. (1998, September). Ant colonies for adaptive routing in packet-switched communications networks. In *International Conference on Parallel Problem Solving from Nature* (pp. 673-682). Springer, Berlin, Heidelberg.

Selvi, V., & Umarani, R. (2010). Comparative analysis of ant colony and particle swarm optimization techniques. *International Journal of Computer Applications*, *5*(4), 1-6.

Caro, G. D., & Dorigo, M. (1998, September). Ant colonies for adaptive routing in packet-switched communications networks. In *International Conference on Parallel Problem Solving from Nature* (pp. 673-682). Springer, Berlin, Heidelberg.

Wei, X. (2014). Parameters analysis for basic ant colony optimization algorithm in TSP. *International Journal of u-and e-Service, Science and Technology*, *7*(4), 159-170.

Colorni, A., Dorigo, M., & Maniezzo, V. (1991). Distributed Optimization by Ant Colonies. In proceedings of ECAL91–European Conference on Artificial Life.

Sivagaminathan, R. K., & Ramakrishnan, S. (2007). A hybrid approach for feature subset selection using neural networks and ant colony optimization. *Expert systems with applications*, *33*(1), 49-60.

YILMAZ, A. H., & AYDIN, Z. Examination of parameters used in ant colony algorithm over truss optimization. *Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *24*(1), 263-280.

Alobaedy, M. M., Khalaf, A. A., & Muraina, I. D. (2017, May). Analysis of the number of ants in ant colony system algorithm. In *2017 5th International Conference on Information and Communication Technology (ICoIC7)* (pp. 1-5). IEEE.

Pettersson, L., & Lundell Johansson, C. (2018). Ant Colony Optimization-Optimal Number of Ants.

Bullnheimer, B., Hartl, R. F., & Strauss, C. (1997). A new rank based version of the Ant System. A computational study.

Stützle, T., & Hoos, H. H. (2000). MAX–MIN ant system. *Future generation computer systems*, *16*(8), 889-914.

AYDIN, Z. IZGARA SİSTEMLERİN OPTİMİZASYONU ÜZERİNDEN KARINCA KOLONİ OPTİMİZASYON ALGORİTMASINDA KARINCA SAYISININ BELİRLENMESİ. *Uludağ Üniversitesi Mühendislik Fakültesi Dergisi*, *22*(3), 251-262. *and Signal Processing*, *2*(3), 1-10.