Journal of Theoretical and Applied Information Technology
30th November 2021. Vol.99. No 22


ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# INTELLIGENT SYSTEM BASED ON K-MEANS AND ACO ALGORITHM WITH PARALLELISM FOR ROUTING IN LARGE AD HOC NETWORK

[1]HALA KHANKHOUR, [2]OTMAN ABDOUN, [3]JÂAFAR ABOUCHABAKA

[1,3]Ibn Tofail University. Department of Computer Science, Faculty of Sciences, Kenitra, Morocco
[2]Abdelmalek Essaadi University. Departement of Computer Science, Faculty of Polydisciplinary, Larache, Morocco

E-mail: [1]hala.khankhour@uit.ac.ma, [2]abdoun.otman@gmail.com , [3]abouchabaka3@yahoo.fr

**ABSTRACT**

With the appearance of new technologies: ICT, IoT, 5G, the spread of wireless internet, increasingly growing services and the current explosion of information systems on Ad Hoc networks, it is therefore necessary to have a new efficient routing strategy especially for large Ad Hoc networks, which takes into consideration the quality of service to increase the lifetime of the network by reducing energy loss while being responsive to changes in the environment. Thus, new emerging methods to save time during the journey of the message from the source node to the destination. In this context, this article focuses on developing a hybrid model between three approaches: the best known K-mean algorithm for clustering and the most popular Ant Colony Algorithm (ACO) and using the newest parallel optimization approach today. We will first take a quick look at the solutions existing for the programming of parallel machines; it introduces a new method of calculation by the processor. However, parallelism makes it possible to create a new model capable of finding the shortest path in terms of cost and time. So, our motivation is to find the shortest path from the source node to the destination with a rapid speed of heuristics and parallelism. Details of our new approach, methods and analyzes are clearly described, well presented and relevant to the objectives of the project. To assess our model, we compared it with another recent article. The results show that our new approach can be beneficial in solving routing problems in AD Hoc networks that are large (until 1000 sensors) and which meet the user need of modern life.

**Keywords:** *Artificial Intelligence, NP-Complete, Parallel Computer, Ant Colony Optimization, And Ad Hoc Network.*

## 1. INTRODUCTION

The increasing demand of the wireless network day by day, the rapid and evolutionary change of topologies of the Ad Hoc network and which does not have pre-existing infrastructure, the energy offloading of the transmission nodes, has become inactive due to long The distances traveled by packets on the network, these problems now pose a great challenge for researchers to optimize routing, especially for large Ad Hoc networks. So, an ideal algorithm should ensure the quality of service and thus find an optimal path for the transmission of packets from the source node to the destination in a short time [1][2]; this genre of complex problem is considered as a NP-hard problem, because it requires a lot of resources (time, memory, computational capacity, energy capacity of nodes), as best known example is the traveling salesman problem (TSP), researchers have been looking for a way to optimize and improve TSP, but it is difficult to solve this genre of TSP problem with polynomial time algorithms or with exact methods, this is also the case of our problem with large-scale Ad Hoc networks. As seen in Figure1.

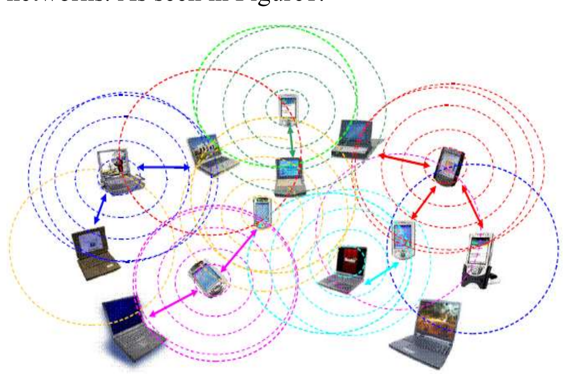

*Figure 1: Mobile Ad Hoc Network*

Most researchers seek to solve the routing problem in Ad Hoc network by using the NS-2 or NS-3 simulator, but this type of simulators used are limited to 50 nodes at most and it loads the bandwidth a lot by control messages [3] as showen in Figure 2. To solve this size of complex problems, we converge to use the meta-heuristic [4].

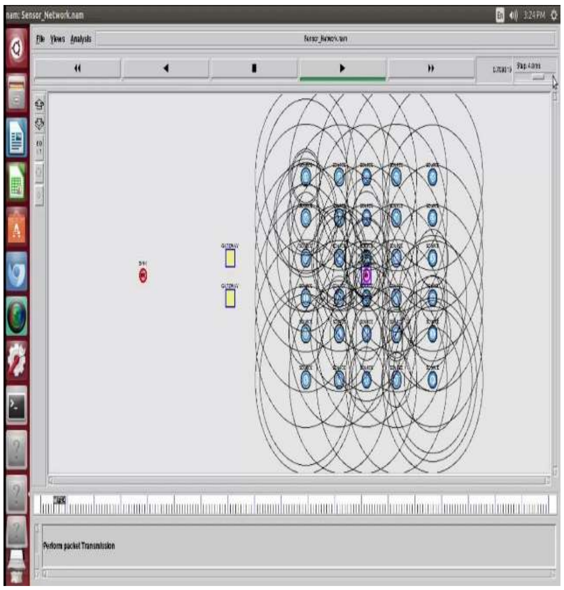

*Figure 2: Example of simulator network NS2*

Over the past two decades, several metaheuristics have ensured their performance for solving complex combinatorial problems, there are several metaheuristics approaches: the constructive approach, the local search approach (the threshold acceptance methods, the taboo method), the evolutionary approach (the genetic algorithm, the distributed search method, the algorithm ant colony optimization (ACO)) and the hybrid approach. These methods can solve the most complex problems within a reasonable time [5]. Currently, researchers have proven that algorithm ACO is powerful in terms of performance and scalability [3].

The main concept of ACO is simple, it consists of studying the behavior of ants in search of food, due to a substance (pheromone) deposited on the ground when the ants move, so the ants can choose the shortest path between their nest and the food, for example, the ants choose the next direction, the one with a higher concentration of pheromone as shown in Figure 3. So the first algorithm ( ant system ) was proposed by Dorigo in 1992, after that in 1997 Dorigo & Gambardella was proposed other algorithm ant colony system (ACS) as defined in this article [6].

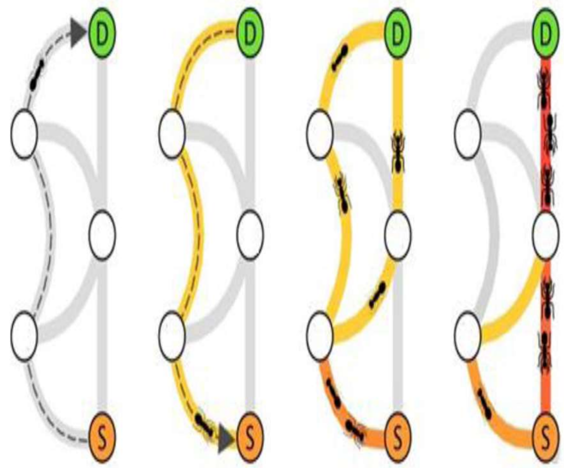


*Figure 3: How real ant find a shortest path*

In 2013, the work proposed in the article [7], uses city maps to have real simulations, and then they apply ACO to improve the quality of service of the network wireless, in the same context, the authors of article [8] in 2014, chose the VANET network model which has high mobility, they use ACO to find the best path by reducing collisions. In 2017 the authors proposed in this article [9] , an algorithm to solve the problem of the fast link between the source and the destination by using the ACO to find the shortest path. Another improvement, Majd Latah used an original idea in his article [10], he grouped the cities into a cluster, and then he applied the ACO algorithm to find the optimal path for each cluster. Today, there is a growing evolution of the Internet of Things (IoT) and its strong demand for resources and services of the Ad Hoc network, to meet the customer's need of modern life. In the same subject, in 2018, the authors of the article [11] propose a "Fast Ant Colony Optimization" algorithm to minimize the computation time of clusters, based on the concept of ACO. Given the evolution and dynamics of the network, in 2020 the authors of this article [12] have chosen to work on vehicles equipped with GPS to find the best path. In the same context in 2021, the researchers focused in this work [13] on solving the combinatorial optimization problem such as the traveling salesman problem (TSP) by using ACO but the speed time is very low. On the other hand, With the emergence of new technologies, it is necessary to have a new efficient routing method especially for large Ad Hoc networks, which takes into consideration the quality of service to increase

Journal of Theoretical and Applied Information Technology
30th November 2021. Vol.99. No 22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the lifetime of the network and decrease conflicts in short time. For this reason we propose to use a new technology in our approach: a parallel optimization approach,

In this paper, we first proposed to split the large Ad Hoc network map into a set of small towns by taking of the advantages of the most famous clustering algorithm is k-means algorithm, in order to prepare our cluster for parallel computing, to accelerate the resolution of the problem whatever the size of the Ad Hoc network to be solved, based on the concept of ACO. Our challenge is to change the design of the metaheuristics algorithm ACO to take advantage of GPU for solving large-scale complex problems in ad hoc network with a view to high effectiveness and efficiency in parallelism, so we use an hybrid method-based K-mean and ACO algorithm with parallel optimization approach (POA).

This work is structured as follows, in section II present the review of ACO, all the equations and methods used in this paper are presented in section III, the new approach proposed in this work: hyprid method-based k-mean and ACO algorithms with parallel optimization approach in Ad Hoc Network (PK-ACO) in section IV, after that, we discuss the experimental setup and results analysis by using PK-ACO to solve the problem of routing in large Ad Hoc networks until to 1000 sensors, and this paper is concluding in the last section.

## 2. REVIEW OF ACO

The original idea inspiration from the way that ants searching for food by selecting the shortest path between a food source and their nest. The path which is marked with the strongest pheromone concentration is the path that the ants choose [14]. The ant measures from 0.01 to 3 cm and weighs 1 to 150 mg, see the Figure 4.

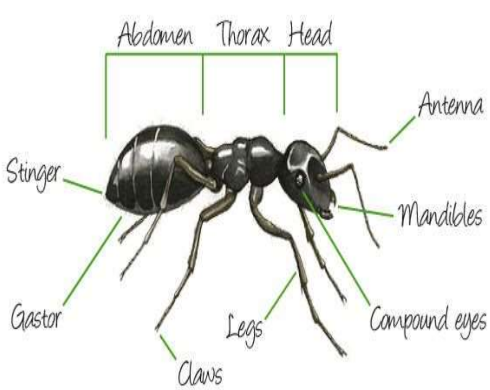


*Figure 4: Description Of The Ant*

Biologists have thus observed, in a series of experiments carried, that a colony of ants with two unequal length paths leading to a food source tended to use the shortest path [15].

In the beginning, the ants are all concentrated in the colony, and they have to find the location of the food and bring the food home. At first, all the ants move randomly. As the ant moves, it will establish pheromone tests. The pheromone is attractive to ants: when the other ants find a movement towards the position with the pheromone, they tend to move along the way instead of moving randomly. The pheromone evaporates over time, reducing the attractiveness of the ant [16].

Ants use the environment as a communication medium: they indirectly exchange information by depositing pheromones, all describing the state of their "work". The information exchanged has a local scope, only an ant located at the place where the pheromones were deposited has access to it. This system bears the name of "stigmergy" [17] and is found in several social animals (it has been studied in particular in the case of the construction of pillars in termite nests) [18].

Therefore, algorithms of Ant colony optimization are algorithms robust inspired by ant behavior and form a family of optimization meta-heuristics.

## 3. PRESENTATION OF THE ALGORITHM ACO

In general, an ant colony algorithm is an iterative population algorithm where all individuals share a common knowledge which allows them to guide their future choices and to indicate to other individuals the directions to follow or on the contrary to avoid [19].

### 3.1 Ant system

The first algorithm was proposed by Dorigo, is the ant system (AS), according to the literature, each ant that has completed the tour from the source to the destination updates the pheromones. in general, we define the solution by the edges of the graph $\tau(r,s)$, and the update of the pheromones for each edge $\tau(r,s)$ which contains the cities r and s, is presented as follows (1) [20] :

$$\tau(r,s) \leftarrow (1-\rho)\tau(r,s) + \sum_{k=1}^{m} (\Delta\tau(r,s)^k)$$

Where m is the number of ants and $\Delta\tau(r,s)^k$ is the quantity of pheromone placed on

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

edge $(r, s)$ by the k-th oven, and $0 < \rho < 1$ is the evaporation rate, (2) :

$$\Delta\tau(r,s) = \begin{cases} 1/l_k & if\ ant\ uses (r,s) in\ its\ tour \\ 0 & otherwise \end{cases} \quad (2)$$

Where $l_k$is the tour length of the k-th ant. During the construction of the solutions, the ants in AS cross the construction graph and take a probabilistic decision at each vertex. The transition probability of ant k going from city i to city j is defined as equation (3):

$$p\left(cr\middle|s(S_k^p)\right) = \begin{cases} \dfrac{t\,(r,s)^{\alpha}\ \ \mu\,(r,s)^{\beta}}{\sum_{c_i \in n\,(s_k^p)}\ t\,(r,l)^{\alpha}\ \ \mu\,(r,l)^{\beta}} & if\ (s \in (S_k^p)) \\ 0 & otherwise \end{cases} \quad (3)$$

Where $n(S_k^p)$ is the set of components that do not yet belong to the partial solution $(S_k^p)$ of ant k, and α and β are parameters that control the importance relative of the pheromone with respect to heuristic information $\mu(r,s) = 1/_d$ where d is the length of component $c(r,l)$(i.e., edge $(r,s)$).

### 3.2 Ant colony system

The ants must find the location of the food and bring the food home. At first, all the ants move around at random. As the ant moves, it will set up pheromone trials. The pheromone is attractive to ants: when other ants find movement to position with the pheromone, they tend to move along the path instead of moving randomly. The pheromone evaporates over time, reducing the attraction to the ant [14].

An ant is on a node i chooses to move to city j by applying the following rule as given in equation [6] (4):

$$s = \begin{cases} argmax_{\mu \epsilon j_r^k} \{[t_i].[\mu_{ij}]^{\beta}\} & if\ q \leq q0 \\ s \quad otherwise \end{cases} \quad (4)$$

Where q is a random number uniformly distributed in [0,1], and S equal to the equation below (5):

$$\rho_{ij}^k = \begin{cases} \dfrac{[t_{ij}]^{\alpha}[\mu_{ij}]^{\beta}}{\sum_{\mu \in j_r^k}\ [t_{ij}]^{\alpha}[\mu_{ij}]^{\beta}} & if\ j \in (j_r^k) \\ 0 \quad otherwise \end{cases} \quad (5)$$

Where $j_r^k$ is the list of possible displacements for an ant k, $\mu_{ij}$ the visibility, which is equal to the inverse of the distance of two cities i and $j(1/_{d_{ij}})$ and $t_{ij}$ the intensity of the track; the two main parameters controlling the algorithm are α and β, which control the relative importance of the intensity and visibility of an edge [21]; when the ant reaches its destination, she puts down a quantity of pheromone on each edge.

### 3.3 The updating rule of the local pheromone

The main goal is to reduce visited edges less attractive for future ants by using the local pheromone update rule [22]. This rule allows ants to revisit cities later in another tour: the local equation for updating the amount of pheromone is given by equation (6):

$$\tau(r,s) = (1-\rho).\tau(r,s) + \rho\Delta(r,s)\Delta(r,s) = \tau_0 \quad (6)$$

Where $\tau(r,s)$ is the amount of pheromone on the edge $(r,s)$ at time t.

$\rho$ : is a parameter governing the decrease of pheromones such that $0 < \rho < 1$

$\tau_0$: is the initial value of pheromones in all edges experimentally, the optimal value for $\rho$ was found at 0.1 and a good formulation for $\tau$ was found $\tau_0 = n/_{L_{nn}}$.

Where n is the number of in the graph and $L_{nn}$ is the length of the lap found by a heuristic of the nearest neighbor.

### 3.4 The updating rule of the global pheromone

We apply the rule of updating the global equation to update the level of pheromones, as shown in equation (7):

$$\tau(r,s) = (1-\alpha).\tau(r,s) + \alpha\Delta\tau(r,s)$$

Where,

$$\Delta\tau(r,s) = \begin{cases} (L_{gb})^{-1} & if\ (r,s) \in global\ best\ tour \\ 0 & otherwise \end{cases} \quad (7)$$

The global update reinforces the best path, which helps to use the best paths found.

## 4. PROPOSED NEW APPROACH: HYBRID METHOD-BASED ON K-MEANS AND ACO ALGORITHMS WITH PARALLELISM IN AD HOC NETWORK

All modern operating systems support concurrency across processes and threads. Processes are instances of programs that generally run independently of each other, for example, working with the Thread class can be very tedious and error-prone. For this reason, the Concurrency API was introduced with the release of Java 5.

The API is in the java package. Concurrent use and contains many useful classes for managing concurrent programming.

Since that time, the Concurrency API has been improved with each new version of Java, it provides new classes and methods for dealing with competition [23].

In this work, we focused on the advantage of the services concurrency API, the executor services. This allows knowing how to make a simple spot parallel for applied this method to our complex problem for big map Ad hoc network.

For using the advantage of parallel optimization approach on our problem, our work consists to use parallel computation on the Local update or global update of the ants part, because this stage of the algorithm depends on the number of ants chosen, so we considered the number of ants as a number of processors. The proposed new algorithm PACO for solving the problem of routing in Ad Hoc big map network in short time, with parallel optimization approach (POA), is shown in Figure 5:

```
Initialize the parameters.
Initialize the pheromone tracks.

Place the ants in the source.
Repeat.
call the parallel function according to the
processor architecture (number of threads),
   Construction of the road,
   For each ant
call the parallel function according to the
number of ants,
      Update local with the executor services.
   End for
   Update overall,
Update overall
```

*Figure 5:The proposed algorithm PACO for solving the routing in Ad Hoc networks big map with calcul computing*

Our challenge in this work is to found a new approach of metaheuristics algorithm to take advantage of GPU for solving the problems of routing in a large-scale Ad hoc network with a view to high effectiveness and efficiency in a short time and find the best shortest path, from the source node to destination node. For this reason, we have found another way, is to divide the given map into m groups of adjacent cities to solve this case of problem in speed time. We choose to use k-means algorithm because it is the most famous algorithm for clustering.

### 4.1 Hybrid method-based on k-means and ACO (PK-ACO)

One of the most popular grouping algorithm is the k-means classification algorithm. In the k-means, n observations will be divided into k groups so that each observation belongs to the cluster with the most average close. The first k is initialized according to the number of clusters desired. Each data point is assigned to the nearest centroid and the set of points assigned to the centroid is called a cluster. Each cluster centroid is updated accordingly with points allocated to the cluster. The process will be repeated until the centroids remain the same or no point changes cluster [24] [11], as shown in Figure 6. The Figure 4 below represents an algorithm of the K-Means method:

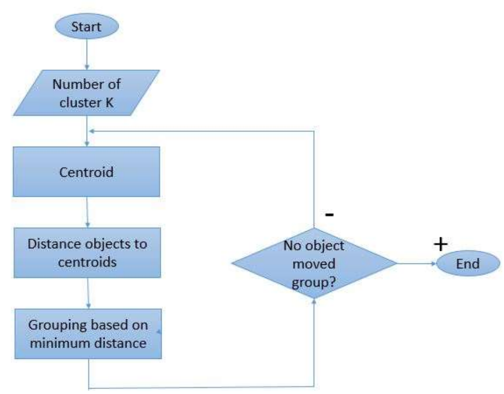


*Figure 6: Algorithm K-means*

For more details, in Figure 7 the pseudo code proposed of K-means algorithm

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
N: number of data objects
K: Number of clusters

Objects [N]: array of data objects
Clusters[K]: array of cluster centers
Membershinp[N]: array of object
membershinps

begin
Kmeans_clustering()
 While P/N > threshold
  P ← 0
   for i ← 0 to N-1
    for j ← 0 to K-1
    distance ← (objects[i] – clusters[j] )
    if distance < dmin
        dmin ← distance
         n ← j
    if membership[i] # n
         P ← P + 1
         membership[i] ← n
    new_clstr[n]←new_clstr [n] + objects[i]
    new_clstr_size[n]← new_clstr_size[n]+1
    end for
   end for
   for j ← 0 to K-1
   clusters[j][*] ← new_clusters[j][*]/
                              new_clstr_size[n]
   new_ clstr[j][*] ← 0
   new_clstr_size[n]← 0
   end for
end
```

*Figure 7: Pseudo-code of K-means algorithm*

We proposed a hybrid method based on K-means and PACO algorithms, is shown in figure 6.

```
Launch the k-means clustering algorithm
For each collection of cities
        Launch PACO (review Fig 2)
  Put together the solution
End for
```

*Figure 6: Pseudo-code of our new approachv PK-ACO algorithm*

In our new approach PK-ACO, you have to know that the performance of parallelism depends on the performance of the k-means algorithm, if it works well and divides the cities into cities almost equivalent, the parallelism works well. For this reason we tried to study the problem as mTSP to evaluate our part of our new approach “k-means algorithm”

## 5. EXPERIMENTAL SETUP AND ANALYSING THE RESULT BY USING PK-ACO

### 5.1 Solving mTSP problem by using PK-ACO

The Multiple Travel Traveler Problem (mTSP) is an extension of the traveling salesman problem (TSP) with a lot of sellers is authorized. Considering a set of cities, a depot where there are sellers, and a cost metric, the objective of (mTSP) is to determine a tour for each seller so that the total cost of the trip is minimized and let each city be visited once by a seller [25].

The requirements on all roads are:

- ✓ All ants must start and end in the same town,
- ✓ each city visited only once and by a single ant.

We compared our result with another article M. Latah (MK-ACO) [10], he proposed to use the k-means clustering and ACO algorithms to resolve the mTSP problem, the comparison seen in Table 1 with 4 vehicles (k=4).

*Table 1: The Computational Results Between PK-ACO and MK-ACO of Solving mTSP problem*

| TSPLIB | Optimal | MK-ACO | PK-ACO |
|---|---|---|---|
| Att48 | 10628 | 28510.82 | 10700 |
| Berlin52 | 7542 | 8820.24 | 7531 |
| Rat99 | 1211 | 2814.74 | 1211 |
| Bier127 | 118282 | 294273.77 | 118279 |

Moreover, we cite the paths of our result for each instance

✓ Att48 :
path=[1, 8, 9, 38, 31, 44, 18, 7, 28, 36, 30, 6, 37, 19, 27, 17, 43, 20, 33, 46, 15, 12, 11, 23, 14, 25, 13, 47, 21, 32, 39, 48, 5, 42, 10, 24, 45, 35, 26, 4, 2, 29, 34, 41, 16, 22, 3, 40, 1]}

✓ rat99:
path=[1, 2, 11, 12, 3, 4, 14, 5, 6, 7, 8, 9, 18, 17, 16, 15, 23, 24, 25, 26, 27, 36, 35, 44, 34, 33, 42, 43, 45, 54, 53, 52, 51, 50, 49, 58, 59, 60, 61, 62, 63, 72, 81, 80, 71, 70, 69, 78, 77, 76, 85, 86, 95, 96, 97, 98, 99, 90, 89, 88, 87, 79, 68, 67, 66, 65, 64, 74, 73, 82, 83, 93, 92, 91, 94, 84, 75, 56, 57, 55, 46, 47, 38, 39, 48, 40, 41, 32, 31, 30, 29, 37, 28, 19, 20, 21, 22, 13, 10, 1]}

✓ bier127:
path=[1, 16, 2, 51, 57, 54, 45, 103, 44, 40, 35, 36, 37, 41, 30, 43, 34, 39, 42, 38, 26, 25, 33, 122, 28,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

29, 32, 80, 79, 77, 18, 21, 17, 22, 4, 23, 24, 108, 20, 15, 106, 6, 114, 105, 7, 120, 10, 115, 13, 50, 121, 5, 56, 124, 52, 55, 66, 47, 53, 49, 118, 48, 46, 94, 112, 111, 107, 127, 93, 95, 123, 97, 98, 101, 102, 63, 119, 96, 109, 87, 86, 85, 88, 110, 104, 125, 89, 92, 99, 65, 113, 64, 58, 100, 90, 116, 60, 62, 61, 91, 59, 3, 11, 9, 8, 72, 19, 67, 73, 74, 68, 71, 70, 69, 75, 76, 78, 117, 84, 81, 126, 82, 83, 31, 27, 12, 14, 1]}

We notice from Table1, that for a small town like Berlin52, after several executions, the approximate solution found is 7531better than the other algorithm of MK-ACO is 8820. With regard to large-scale cities, we tested Bier127, we found so good result of our algorithm PK-ACO is equal at 118279 more better that MK-ACO is equal at 294273.77.

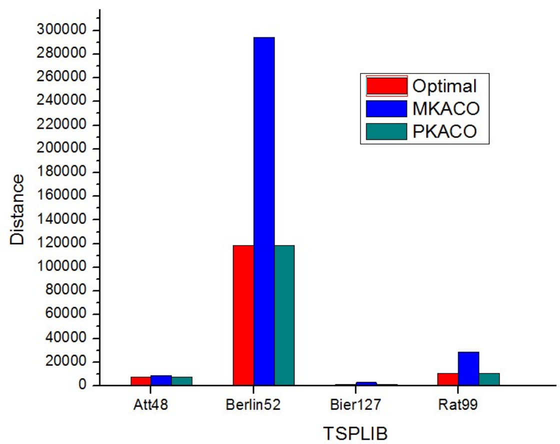


*Figure 8: Algorithm K-means Comparison of PK-ACO and K-ACO of solving MTSP problem with four instances*

Figure 8 show a comparison of PK-ACO and K-ACO for solving MTSP problem, the PK-ACO algorithm was closer to the K-ACO algorithm for the small instance Berlin52, and even for the medium size instance Rat99 there was medium acceleration but for the large instance like bier127 the program PK-ACO was faster with maximum acceleration. And always the PK-ACO program was closer to the optimal.

We notice that the PK-ACO algorithm does not improve the hig ht quality of the routing for small towns, but on the other hand, for large size problems it considerably improves the quality of service in large-scale network, specifically for big town as bier127

We conclude that parallel optimization approach is very powerful to find the shortest path in large scale town in a short time.

However, we suggest an another comparison with a recent algorithm in this article [26] to assess our work in terms of speed time, by using 10 TSP instances,

the authors used the Ant Colony Optimization algorithm to solve the traveling salesman problem (TSP), they proposed a self-adaptive ACO system to test the speed of convergence of his algorithm DEACO, the Table 2 present the comparison of the average of execution time between proposed algorithm PK-ACO and DEACO for 12 TSP instances.

*Table 2: The Computational Results Between Pk-Aco And Deaco For 12 Tsplib*

| TSPLIB | Optimal | DEACO | PK-ACO |
|---|---|---|---|
| Rad400 | 15,281.0 | 15,323 | 15.280 |
| Fl417 | 11,861.0 | 11,866 | 11,861 |
| Pcb442 | 50,778.0 | 50,964.5 | 50,760 |
| D493 | 35,002.0 | 35,100 | 35,002 |
| U574 | 36,905.0 | 36,937.0 | 36,915 |
| Rat575 | 6773.0 | 6773.0 | 6773 |
| P654 | 34,643.0 | 34,645.0 | 34643 |
| D657 | 48,912.0 | 49,124.0 | 48,915 |
| U724 | 41,910.0 | 42,038 | 41,910 |
| Rat783 | 8806.0 | 8916.0 | 8806 |
| KroA200 | 29,368 | 29,368 | 29,368 |
| Ch150 | 6528 | 6528 | 6528 |

We notice from Table 2, most of the results of the PK-ACO algorithm are closer to the optimal than the results of the DEACO algorithm, even some instances are equal to the optimal, for example: Fl417, D493, Rat575, P654, P654, U724, Rat783, KroA200, and Ch150. And next step we compare the speed time between PK-ACO and DEACO as shown in Table 3

*Table 3: The Computational Speed Time Between Pk-Aco And Deaco For 12 Tsplib*

| TSPLIB | DEACO (s) | PK-ACO (s) |
|---|---|---|
| Rad400 | 130 | 98,3 |
| Fl417 | 39,3 | 29,6 |
| Pcb442 | 129 | 90 |
| D493 | 130,12 | 95 |
| U574 | 110 | 80,7 |
| Rat575 | 127 | 90 |
| P654 | 100,2 | 73 |
| D657 | 126,5 | 85 |
| U724 | 130 | 91 |
| Rat783 | 140,81 | 83 |
| KroA200 | 166,3 | 87 |

The purpose of the presentation of Table 3 is to compare the previous DEACO studies with our PK-ACO work for 11 large cities, the results show that our new approach is faster than the DEACO algorithm in terms of time, for example, the instance Rat789, there is a big difference of 57 s between the two algorithms PK-ACO and DEACO. That mean the parallel optimization approach reduce the time execution because, the transmission time of messages between processes is lower.

The second phase of this research focuses on applying our new approach PK-ACO algorithm on the large-scale Ad hoc network until 1000 sensors.

### 5.2 Solving the problem of routing in Ad Hoc by using PK-ACO

The original idea used for this application between PK-ACO on Ad Hoc network, we considered the antennas of the ants as a sensor radius (in this work is equal to 20), each ant represents a node ID.

To solve the problem of routing for large-scale Ad hoc networks, we generated a large random topology until 1000 sensors, between two methods: sequential (K-ACO) and parallel (PK-ACO), on the other hand, before doing the local and global update we have looked at the spots on the ants, each ant is looking for its way to the desired destination, with Sensor radius equal at 20.

After carrying out several executions, we have chosen the best parameters shown in Table 4 that were used or solving the problem of routing in Ad Hoc network.

*Table 4: The Parameters Using By PK-ACO*

| Parameter name | Values |
|---|---|
| $\boldsymbol{\rho}$ | 0.1 |
| $\boldsymbol{\alpha}$ | 1 |
| $\boldsymbol{\beta}$ | 2 |
| $\boldsymbol{q_0}$ | 0.1 |
| Number of iterations | 100 |
| Source sensor | 3 |
| Destination sensor | 45 |
| Sensor radius | 20 |
| Number of ants | 8 |

In Table 5, the configuration used in this research is as follows:

*Table 5: The Configuration Computing*

| Configuration | Value |
|---|---|
| RAM | 8Go |
| Technology | LPDDR3 SDRAM |
| Vitesse | 1866MHz |
| Cache | 4Mo |
| CPU | Intel Core i7, 7600U/2.8 GHz |
| maximum speed in turbo mode | 3.9 GHz |
| Number of hearts | double heart |
| Technology platform | Technology Intel vPro |

The simulation results for random network topologies until 1000 sensor by using PK-ACO, the comparison seen in Table 6.

*Table 6: The Comparison Between Two Methods: Sequential (K-Aco) And Parallel (Pk-Aco) By Using Sensor Network Ad Hoc*

| Number of sensors | Execution time | |
|---|---|---|
| | Sequential K-ACO (ms) | Parallel PK-ACO (ms) |
| 20 | 1454 | 702 |
| 100 | 2487 | 1002 |
| 300 | 2945 | 1524 |
| 500 | 5568 | 2378 |
| 600 | 9425 | 3541 |
| 700 | 12456 | 5124 |
| 800 | 21886 | 7012 |
| 900 | 29631 | 10499 |
| 1000 | 99123 | 11661 |

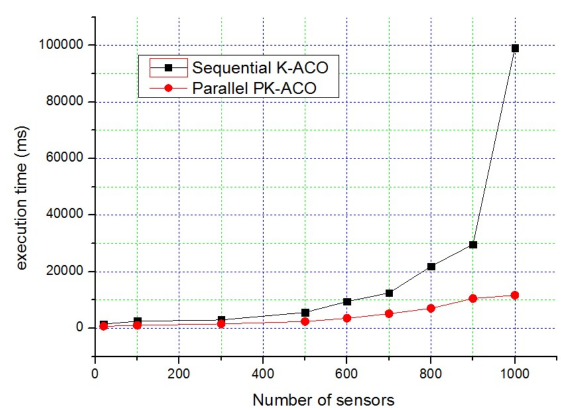


*Figure 9: Comparison between sequential K-ACO and parallel PK-ACO in Ad Hoc Network*

Figure 9 and Table 6 show the acceleration of time for the sequential K-ACO and parallel PK-ACO in Ad Hoc Network, we observe that for small networks about 20 sensors, the time is equal at 1454 ms for K-ACO and is equal at 702 for PK-ACO, there is not a remarkable time difference between them up to 300 sensors, but when we increase the size of the network we start to have the time difference between sequential K-ACO and parallel PK-ACO in Ad Hoc network, especially for the network of 1000 sensors we see the biggest difference in time, for PK-ACO is equal at 11661 ms but for K-ACO is equal at 99123 ms.

therefore, the sequential K-ACO algorithm finds difficulty finding the shortest path in a short time because of the memory usage which is more expensive in terms of calculation, on the other hand, with the use of parallel data programming in which a block of threads worked on the solution of a single ant, the solutions obtained for the PK-ACO program were significantly better quality than the sequential K-ACO program, he found the shortest path in speed time.

we notice from the interior work it was difficult to create simulations more than 50 nodes in the NS2 simulator, however, we can't deal with the problems of large networks [27][28], but with the use of methaheuristics and especially of the method ACO, with a new technology in our approach: a parallel optimization approach, we managed to generate simulations up to 1000 sensors, and the solutions obtained for our program PK-ACO were significantly better quality and it found the shortest path in speed time

## 6. CONCLUSION

This paper has presented a bio-research algorithm to improve the quality of service in large-scale ad hoc network. It presented a model combining three algorithms, based on a hybrid of K-means and ACO algorithms by using the latest technologies like a parallel optimization algorithm, to speed up the even more search mechanism; we have used new functions such as thread control, executor service, and others. The goal is to find the best path from the source node to destination, For the assessment of our model, firstly, we have applied our new approach in mTSP problem, after, we have applied on large-scale Ad Hoc network. It has been concluded from experimental work that our new approach has a strong ability to find the shortest path in speed time, therefore the packet delivery time is also very short and thus limit the energy consumption of the node, and subsequently increase the lifetime of the network, in order to satisfy the end user.

In the future, we propose to evaluate the performance of the Ad Hoc network by using another new method to solve the problem of routing in large-scale Ad Hoc network [29].

## REFERENCES

[1] M.Bagheri, A. Ebrahimnejad, S. Razavyan, et al.," Solving fuzzy multi-objective shortest path problem based on data envelopment analysis approach". Complex Intell. Syst. 7, pp 725–740 (2021).

[2] S.W.C.Huang, "Asynchronous distributed optimization via dual decomposition for delay-constrained flying ad hoc networks," Computer Communications, vol.137, pp. 70-80, March 2019,

[3] M. Ahsan, S. Zai, D. Kumar, A. Abbas, F. Ali, "Analysis of swarm intelligence routing protocols in large-scale mobile Ad Hoc network (MABET'S)", Journal of Theoretical and Applied Information Technology, 31st , Vol.99. No 2, January 2021.

[4] V. Leggieri, M. Haouari, "A matheuristic for the asymmetric capacitated vehicle routing problem," Discrete Applied Mathematics, 234, pp. 139-150. Cited 10 times, 2018.

[5] R. Skinderowicz, "The GPU-based parallel Ant Colony System", Journal of Parallel and Distributed Computing , Vol. 98, pp. 48-60, Elsevier December 2016,

[6] E.Neshati, A. A. P. Kazem, "QoS-based Cloud Manufacturing Service Composition using Ant Colony Optimization Algorithm," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 1, 2018.

[7] U. Mane and S. A. Kulkarni, "QoS realization for routing protocol on VANETs using combinatorial optimization," 4th Int. Conf. Comput, Commun. Netw. Technol. (ICCCNT), pp. 1–5, 2013.

[8] Er. GurpreetSingh, Dr. Vijay Dhir, "Open Vehicle Routing Problem by Ant Colony Optimization", IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 5, No. 3, 2014.

[9] S. Majumdar, Shivashankar, P. R. Prasad, S. S. Kumar and K. N. S. Kumar, "An efficient routing algorithm based on ant colony optimisation for VANETs," 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2016, pp. 436-440.

[10] M. Latah, "Solving Multiple TSP Problem by K-Means and Crossover based Modified ACO

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm," Department of Computer Engineering Ege University Izmir-Turkey. 2017.

[11] A. S. Girsang, T. W. Cenggoro, and K. W. Huang, "Fast Ant Colony Optimization for Clustering", Indonesian Journal of Electrical Engineering and Computer Science Vol. 12, No. 1, pp. 78-86, 2018.

[12] H. Du, Y. Hong, N. Xia, G. Zhang, Y. Yu and J. Zhang, "A Navigation Satellites Selection Method Based on ACO With Polarized Feedback," in IEEE Access, vol. 8, pp. 168246-168261, 2020

[13] T.C. Hung, P. Thi," The, Incorporate ACO routing algorithm and mobile sink in wireless sensor networks," International Journal of Electrical and Computer Engineering (IJECE), October 2021.

[14] Y.Liu, B. Cao, & Li, Henua.:" Improving ant colony optimization algorithm with epsilon greedy and Levy flight". Complex Intell. Syst. 7, pp:1711–1722, (2021).

[15] Z. Ali, W. Shahzad, "Comparative Analysis and Survey of Ant Colony Optimization based Rule Miners," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 1, 2017

[16] A. R. Mahlous, A. Zarrad, T. Alotaibi," State Transition Testing Approach for Ad hoc Networks using Ant Colony Optimization," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 6, 2018.

[17] A.Guerna, S. Bitam, & C. T. Calafate,: "a novel ant colony system for roadside units deployment in vehicular ad hoc networks". Peer-to-Peer Netw. Appl. 14, pp: 627–643 (2021).

[18] A. Roudane , M. Youssfi , K. Mansouri, "Semantic Micro-Services Model for Vehicle Routing using Ant Colony Optimization," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 11, 2019

[19] K.Ouaddi, Y. Benadada, F. Z. Mhada, "Ant Colony System for Dynamic Vehicle Routing Problem with Overtime," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 6, 2018.

[20] M. Dorigo, T. Stu¨ tzle, "Ant colony optimization: overview and recent advances," in Handbook of metaheuris- tics. Springer, pp. 227–263, 2010

[21] S. Akhter , M. N. Ahsan, S. J. S. Quaderi, "Modeling Ant Colony Optimization for Multi-Agent based Intelligent Transportation System," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 10, 2019

[22] S. Kannan, T. Kalaikumaran, S. Karthik and V.P. Arunachalam, "Ant Colony Optimization for Routing in Mobile Ad-Hoc Networks" Department of Computer Science and Engineering, SNS College of Technology, Sathy Main Road, Coimbatore 641035, Tamil Nadu, India, Intemational Joumal of Soft Computing 5 (6): 223-228, 2010

[23] R. Syah , M. K. M. Nasution, I. Husein , M. Elveny, "Optimization Tree Based Inference to Customer Behaviors in Dynamic Control System", International Journal of Advanced Science and Technology Vol. 29, No. 8, pp. 1102-1109, (2020).

[24] Y. Li, H. Wu, " A Clustering Method Based on K-Means Algorithm", Department of Computer Science Xinyang Agriculture College Xinyang, Henan 464000, International Conference on Solid State Devices and Materials Science, Physics Procedia 25 ( 2012 ) 1104 – 1109

[25] H. Khankhour, J.Abouchabaka, O. Abdoun, "A New Design of an Ant Colony Optimization (ACO) Algorithm for Optimization of Ad Hoc Network", Networking Intelligent Systems and security, pp (231-241),Niss2021.

[26] S. Ebadinezhad, "DEACO: Adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem", Engineering Applications of Artificial Intelligence, Volume 92, 10364992, Elsevier June 2020.

[27] H. Khankhour, J.Abouchabaka, O. Abdoun, Lecture Notes in Networks and Systems, 92, pp: 145-154 (2019).

[28] H. Khankhour, J.Abouchabaka, O. Abdoun, "Optimization the Ad Hoc network by using hybridization of genetic algorithm with two optimization algorithm",International Conference on Digital Technologies and Applications, pp:1073-1080, ICDTA'2021.

[29] H. Khankhour, J.Abouchabaka, O. Abdoun," Parallel genetic approach for routing optimization in large ad hoc networks", International Journal of Electrical and Computer Engineering (IJECE), Vol 12, No 1: February 2022.