Computer Science and Information Systems 20(1):195–214 https://doi.org/10.2298/CSIS220820053B

Solution for TSP/mTSP with an Improved Parallel Clustering and Elitist ACO

Gozde Karatas Baydogmus

Department of Computer Engineering, Marmara University
Istanbul, 34722
gkaratas@marmara.edu.tr

Abstract. Many problems that were considered complex and unsolvable have started to solve and new technologies have emerged through to the development of GPU technology. Solutions have established for NP-Complete and NP-Hard problems with the acceleration of studies in the field of artificial intelligence, which are very interesting for both mathematicians and computer scientists. The most striking one among such problems is the Traveling Salesman Problem in recent years. This problem has solved by artificial intelligence's metaheuristic algorithms such as Genetic algorithm and Ant Colony optimization. However, researchers are always looking for a better solution. In this study, it is aimed to design a low-cost and optimized algorithm for Traveling Salesman Problem by using GPU parallelization, Machine Learning, and Artificial Intelligence approaches. In this manner, the proposed algorithm consists of three stages; Cluster the points in the given dataset with K-means clustering, find the shortest path with Ant Colony in each of the clusters, and connect each cluster at the closest point to the other. These three stages were carried out by parallel programming. The most obvious difference of the study from those found in the literature is that it performs all calculations on the GPU by using Elitist Ant Colony Optimization. For the experimental results, examinations were carried out on a wide variety of datasets in TSPLIB and it was seen that the proposed parallel KMeans-Elitist Ant Colony approach increased the performance by 30% compared to its counterparts.

Keywords: ACO, Parallel ACO, Parallel Kmeans, TSP.

1. Introduction

Traveling salesman problem (TSP) is an optimization problem, which is one of the Non-Deterministic Polynomial (NP) problems and is similar to problems such as GPS, UAV routing, logistics routing [17, 23]. The basic approach in TSP is to find the shortest path that a salesman will eventually return to the starting point by visiting n cities in a sequence at one time. There are many algorithms in the literature for solving this problem. In particular, optimization algorithms such as Genetic Algorithm, Artificial Bee Colony Optimization, Artificial Ant Colony Optimization (ACO), and Particle Swarm Optimization have been frequently preferred for the solution of this problem in recent years. ACO stands out as an algorithm that is widely applied in traditional path planning which is easy for GPU parallelization among other algorithms. Fig. 1 is an example showing the TSP solution on a graph. In Fig. 1, the figure on the left represents a graph, while the figure on the right offers graphs solution.

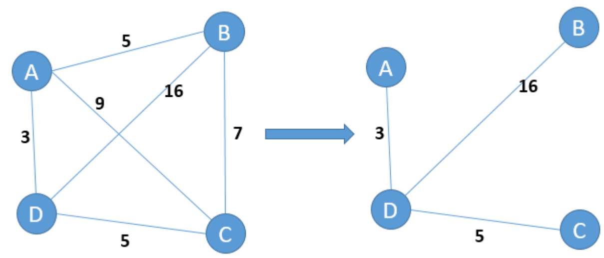

Fig. 1. The example of TSP on a graph without solution

ACO was developed based on the foraging instincts of ants in nature [10–12]. Accordingly, in order to find the shortest path to the food, the ants make use of the pheromone hormone released by the ant on the path it passed while searching for food. This hormone intensifies with each ant passed on and at the same time has the property of evaporation.

Finding a solution by combining clustering and artificial intelligence algorithms has become a favorite of many researchers. Accordingly, the points in the TSP are clustered with the help of the clustering algorithm, so that the closest points to each other are included in different clusters [34]. Then, a connection is made between the clusters created by determining the closest points, and in this way first, the clusters, which are the closest points to each other, are visited and then move to another cluster. The generally preferred algorithm for this is the K-means algorithm [15, 18]. A very easy solution can be found especially for the mTSP problem with this approach.

Parallelization, through the developing technology and GPUs, saves time for solving many complex problems. In this manner, the proposed model in this article designed a parallel approach for the TSP solution. Since the aim is to develop an innovative and successful approach, the literature has been examined in detail and a Parallelized Hybrid algorithm has been proposed considering some deficiencies.

In this study, a new, parallel, and powerful algorithm is proposed for the TSP solution. The strengths of K-means and Elitist ACO algorithms were utilized for this and a new perspective was brought to the studies in the literature. In the proposed method, the following operations carried out:

- The focus of the study is GPU parallelization, so the operations were carried out in parallel whenever possible.
- First of all, various datasets in TSPLIB were selected for the TSP solution and prepared for processing. Particular attention has been paid to the fact that the selected datasets have different numbers of nodes/points.
- Parallel clustering was performed with K-means in the TSP datasets. The K value is determined according to the m value by considering the mTSP problem. Calculation of the distances of other points to the center points determined for clustering was carried out in parallel. In this way, the processing load is reduced by 1/K.
- After clustering, the closest points of K clusters were determined and the clusters were connected to each other from these points.
- Another important part of clustering is the implementation of the Parallel Elitist ACO algorithm. Ants were sent to random points in the dataset for this. The ant will first

wander around in the cluster to which the point it is located, and then move on to another cluster from the connection point. The ant's circulation and the total distance achieved at the end of this circulation were calculated in parallel.
– After all the conditions are provided, a result that can be optimal for TSP is achieved through the parallel hybrid algorithm.

In the following part of the study, detailed information about the related work in Related Work, the methods used and the proposed parallel hybrid algorithm in Proposed Approach are given. In Proposed Approach section, a flowchart of the developed algorithm is given, and then all the steps are explained in detail. Experimental results on selected TSP datasets are shown in Experimental Results section. Discussion and Conclusion section contains the conclusion part of the study.

2. Related Work

In this section, studies which are examines clustering and metaheuristic algorithms in the literature are evaluated. As a result of the literature review, it was discovered that no research study uses both clustering and routing using the GPU parallelization aimed by this study. In this manner, the model proposed in this study brings a new perspective to the literature on GPU parallelization.

In 2022, researchers aimed to design an ACO that would solve TSP by utilizing algorithm improvements and multi-core CPUs and named the algorithm they designed as FACO [30]. Accordingly, at FACO the number of differences between the newly created and selected previous solution is checked and improvements made while maintaining the quality of the existing solution through a more focused search process. The results of the study were examined on many TSPs and it was seen that successful results were obtained with 8-core CPUs and problems with more than 100.000 nodes.

Dihn and others researched a study in 2021 to examine parallel drone scheduling using TSP, where deliveries are split between a truck and a fleet of drones [8]. The solution to the problem is to focus on the known TSP and develop a method. In this context, they proposed a hybrid ACO to solve the problem. The proposed algorithm focuses on a method that represents a TSP solution as a permutation of all data, and then performs the solution by dynamic programming. The hybrid algorithm designed by the researchers is completely based on dynamic programming. For the study, 90 samples were evaluated and it was seen that they gave better results than many studies recommended in the literature.

In 2022, researchers designed an algorithm that takes advantage of ACO to solve, improve overall performance, and shorten solution time in TSP [31]. For this, clustering on ACO parameters, dynamic pheromone evaporation and diversity of solutions in the population were used. In order to observe the working performance, a study was carried out on the problems with the number of nodes ranging from 51 to 2392 and it was observed that the proposed method was successful like the examples in the literature.

Rani et al. proposed an algorithm to determine the travel path in the most optimal way using the Traveling salesman problem and the K-means clustering technique in 2018. The purpose of the research is to develop a web-based application that can help people plan their travels. It proceeds on the assumption that the traveler determines the touristic places they want to visit and the number of days they will stay in the region. The proposed approach consists of two stages, macro grouping using k-means and micro tour editing using the traveling salesman problem. In the study, operations were carried out in the city of Yogyakarta, one of the touristic cities of Indonesia. As a result of the experiments it was seen that the proposed algorithm works well on a small to a medium number of points [28].

The approach developed by Cheng and Mao in 2007; aims to find the minimum cost path using time Windows for the traveling salesman problem. For this, they improved ant colony optimization, which is very popular among metaheuristic algorithms and is frequently used in solving the traveling salesman problem. In the study, they embedded two local heuristics in the ant colony algorithm to manage time window constraints. As a result of the experimental studies, it was discovered that the proposed algorithm solves the traveling salesman problem more efficiently than the standard ant colony algorithm [6].

The algorithm proposed by Stodola et al. designs three new techniques to reduce the negative effects associated with the ant colony optimization method on the traveling salesman problem, such as improving overall performance and reducing to a local optimum. These techniques are the concept of node clustering, adaptive pheromone evaporation, and diversity of solutions in the population. 30 benchmark data samples from well-known TSPLIB benchmarks were used to evaluate the performance of the proposed method and different comparisons with experimental results were performed. The proposed algorithm outperformed these competing methods available in the literature in most cases [31].

In 2014, Bora and Gupta observed the effect of different distance measurement techniques on clustering using the K-means algorithm. This study stands out, especially because distance calculation is very important in clustering algorithms. Manhattan, Euclid, and Cosine distance calculations were evaluated in the study. In addition, Matlab was preferred to improve the study. The values obtained as a result of this study are very important in terms of determining the algorithm to be selected according to the characteristics of the dataset to be used while clustering [3].

Researchers worked on K-Means and Cross Ant Colony Optimization methods to solve the multiple traveling salesman problem in 2020 [20]. They used the K-Means algorithm to determine the ant colony to determine the tour to find the area that each seller would visit. Experimental results were conducted on three different datasets chosen from the application TSBLIB and a different number of salesmen such as 2, 3, 4, and 8. As a result of the application, it was seen that the proposed algorithm gave better results than the K-Means and Ant colony optimization working alone. In addition, the effect of the number of vendors on the operation of the model was also observed.

In 2009, researchers conducted an optimized routing study for multiple Traveling Salesman Problem [26]. In the study, the K-means clustering algorithm was used to cluster the given cities depending on the number of a salesman. In this way, the m-Traveling Salesman Problem problem is reduced to a simple traveling salesman problem. After clustering, an optimized route is created for each salesman in its allocated cluster using metaheuristic algorithms. For this, ”Tabu Search” and ”Simulated Annealing” metaheuristic algorithms were preferred. In the experimental results, it has been seen that Simulated Annealing gives a shorter path than the Tabu search.

The Traveling salesman problem is a problem that can be summarized as finding the shortest path circulating between certain points by the salesmen. The m-traveling salesman problem is generalized to take into account more than one salesman. Latah conducted research in 2016 to find a solution to the m-Traveling salesman problem [21]. In the study, the author proposed a solution method by using ant colony optimization and a genetic algorithm. Accordingly, Latah proposed a new model by running K-means clustering algorithm on ant colony optimization. Popular datasets available on TSBLIB were used to test the study. As a result of the study, the author observed that the modified ant colony algorithm he proposed gave better results than the genetic algorithm.

In 2019, researchers proposed a solution to the traveling salesman problem using the Firefly Algorithm (FA) and K-means clustering [16]. This proposed approach consists of three steps: clustering the points in the dataset, finding the optimal path in each cluster, and creating a connection path between the clusters. In the first step, nodes are divided into sub-problems using K-means, in the second step, the optimal path in each cluster is found by FA, and finally, all clusters are reconnected. Experimental results showed that the proposed approach gave better results compared to other algorithms in the literature.

Chang aimed to increase the efficiency of ant colony optimization by using the K-means algorithm to offer a new perspective on the traveling salesman problem [4]. In the study, the city locations were divided into two or more groups according to the use of the K-means algorithm, and then circulation was carried out within these city groups with ant colony optimization. Finally, a connection was made between these clusters at the closest points. Experimental results showed that the proposed method reduces the computational cost by 32%.

3. Proposed Approach

In this section, the algorithms used in the study are given, and then the proposed model is explained in detail.

3.1. Traveling Salesman Problem

The Traveling Salesman Problem (TSP) is a problem that aims to find the shortest tour that passes through each of a certain number of points with known distances only once. It can be solved by calculating the permutations between all points while the number of points is low, but when the number of points increases, the cost of permutations will increase too much, and after a while, it begins to be unsolvable in polynomial time. For this reason, it started to attract the attention of researchers, and many methods have been developed for a fast and effective solution [17, 23].

TSP can be shown by a graph G=(V, E) are V is the set of nodes/points where V = 1,2,3,…,n and E is the set of edges (distance between two points) connecting the nodes in set V. Euclidean distance is mostly used to calculate the path length since TSP aims to find the shortest path to visit all points. This formula is shown by Equation 1.

$$dist(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{1}$$

where $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ equivalent to the coordinates of the points [1, 14].

3.2. K-Means Clustering

In human life, people do their work by grouping due to their nature. The librarian separates the books according to certain groups while organizing the library or the researcher organizes his/hers belongings according to the season, and groups them by examining the studies related to the subject while researching for his/her publications. Grouping similar objects together have always been in human life.

Computer science, which designs approaches by being influenced by human life, was also affected by this idea and put forward the clustering approach, the infrastructure of which was formed by this idea. Clustering is one of the most common statistical data analysis methods used to obtain a relationship about the structure of the available data [35]. Because we may not always have meaningful data when processing with computers, so knowing how to relate data increases the performance of the model [15, 19, 22]. Clustering, which is an unsupervised learning method, gives better results on such datasets. Clustering algorithms aim to establish relationships between such data. Fig. 2 shows the state of the data before and after clustering.

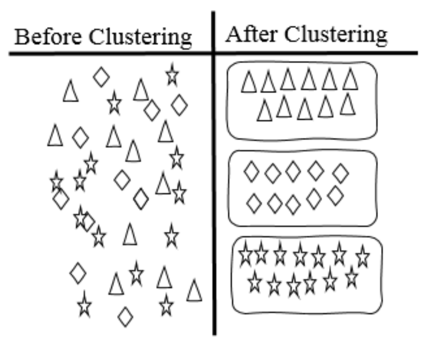

Fig. 2. Clustering

There are two types of clustering algorithms that are performed in Machine Learning; K-means and Hierarchical Clustering. In this work, the K-means algorithm, which is accepted as one of the most used clustering algorithms due to its simplicity and applicability, will be used to determine the relationship between places/locations. Application areas of the K-means algorithm; customer segmentation, game/player analysis, document classification, intrusion/fraud detection, etc. The main problem here is to choose the K value correctly. Because it is necessary to manually (intuitively) assign the K value. There are several methods for determining the appropriate K number, these are; Elbow Method, Average Silhouette Indices, and GAP statistic.

K-means algorithm's main purpose is to determine the center points by minimizing the distance within the cluster. For this, it assumes that the data at hand consists of K clusters and tries to minimize the distance of the points in the clusters to be formed from the cluster mean [18]. The algorithm is based on the Euclidean distance formula given by Equation 2;

$$\sum_{j=1}^{K}\sum_{i=1}^{N}\left\|x_i^j - C_j\right\|^2 \tag{2}$$

In Equation 2, N is the size of the dataset, $X = (x_1, x_2, \ldots, x_n)$ are the points an $C = (c_1, c2_2, \ldots, c_K)$ are centroids in the available dataset. The algorithm performs the following steps;

1. Start with randomly selected K (number of clusters) center points, $\mu_1, \mu_2, ..., \mu_K$.
2. Assign each point in the dataset to the cluster with its closest centroid (based on the distance calculated by the Euclidean distance formula).
3. Calculate the value of the cluster center by averaging all its points. It is checked with Equation 3 and Equation 4 whether the clustering is completed or not.

Adjust

$$C^{(i)} = argmin_j \left\|x^{(i)} - \mu_j\right\|^2 \tag{3}$$

For every i, and adjust

$$\mu_j = \frac{\sum_{i=1}^{m} 1\left\{C^{(i)} = j\right\} x^{(i)}}{\sum_{i=1}^{m} 1\left\{C^{(i)} = j\right\}} \tag{4}$$

for each j.

Hierarchical clustering was not preferred in this study. As it is known, in hierarchical cluster analysis, similarity and distance calculations between data are updated at every step. This causes it to be slower than the K-means algorithm in terms of time in large data sets. Therefore, K-means was preferred in the study.

3.3. Ant Colony Optimization and Elitist Ant Colony Optimization

Metaheuristic algorithms of artificial intelligence have started to gain attention after the increase in access to information and the researchers' turn of their observations into nature [5, 24, 33]. Many algorithms such as genetic algorithm, Artificial Bee colony, and particle swarm optimization have been developed by examining the existence of living things/matter in nature. The main method used to create a shortest path in this study, the Ant Colony Optimization (ACO) algorithm, was formulated on ants' finding the shortest path to food as a result of their wandering in nature.

ACO was first discovered by Marco Dorigo [10–12]. It has been observed that ants can find the shortest path to food as a result of their circulation to find food, and a related algorithm has been designed. Foraging for food in nature, the ant leaves a hormone called pheromone on its way to and from food. Since this hormone has an evaporating structure, it stays on the road for a certain period of time, the level of pheromone increases with each pass, and the level of pheromone decreases when there is no passage. Since the pheromone level on the road will increase over time because evaporation will be less in a short way, so the pheromone level will be higher, all ants will intuitively start to prefer the path with the highest pheromone level. The formulas developed by Dorigo are given by Equation 5 and Equation 6.

$$\tau_{ij}{}^{k} = (1-\rho)\tau_{ij}{}^{k} + \sum_{k=1}^{n} \Delta\tau_{ij}{}^{k} \tag{5}$$

$$p_{ij}^{k} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l\in\mathbb{N}}(\tau_{il})^{\alpha}(\eta_{il})^{\beta}}, \eta_{ij} = \frac{1}{L_{ij}} \tag{6}$$

where p_{ij}^{k} is the probability of the k^{th} ant moving from node i to node j, τ_{ij} amount of pheromone between nodes i-j, η_{ij} cost function between nodes i and j, α pheromone and β heuristic coefficient, n is ant colony size, ρ is the evaporation coefficient. Equation 5 determines the pheromone level between two points by considering the evaporation coefficient. The probability calculation made by Equation 6 is used to determine which of the path will be selected depending on the current point [7, 27, 32].

Elitist ant colony (EACO) is the first system developed by ACO, inspired by applications in genetic algorithm [9]. According to this approach, the pheromone amounts of the edges of the best round are subjected to a pheromone increase in addition to the standard pheromone update given by Equation 5 in the ACO at each cycle [13]. Elitist ACO's difference from the ACO is that the update of the pheromone trails is carried out by the elitist ants, which can be one or several. Equation 7 and Equation 8 are formulas for calculating Elitist ACO.

$$\tau_{ij}(t) = p.\tau_{ij}(t-1) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} + E.\Delta\tau_{ij}^{bs} \tag{7}$$

$$\tau_{ij}^{bs} = \frac{Q}{L^{*}} \tag{8}$$

In Equation 7 and Equation 8, E is the number of elitist ants which we will call E the elitist coefficient. , L^{*} is the length of the best round, τ_{ij}^{bs} is for the best solution and the amount of pheromone belonging to the edges on the best round is increased by $e\frac{Q}{L^{*}}$.

In EACO the elitist strategy for pheromone updating is also involved and Equation 7 is used for pheromone update. Apart from these, probability calculation and selection are exactly the same as ACO. The flow chart for the proposed algorithm is given in Fig. 3.

3.4. Parallel Clustered Elitist Ant Colony Algorithm for TSP

Graphics Processor Units are converters work between the screen and the processor during the creation of the text and graphics displayed on the computer. It undertakes the tasks of processing and reflecting graphics on many technological devices that you actively use in daily life, especially personal computers, smart phones, workstations, digital screens and game consoles.

In this section, information is given about the developed parallel model proposed in the study. When the literature is examined, it is seen that there is no study suggesting a new parallel approach by using clustering and EACO algorithms together. This study consists of three stages; Clustering with parallel K-means, interconnection between clusters at the closest points/nodes and routing with parallel EACO. In this way, a connection will be established between the shortest nodes between the clusters and the algorithm will find

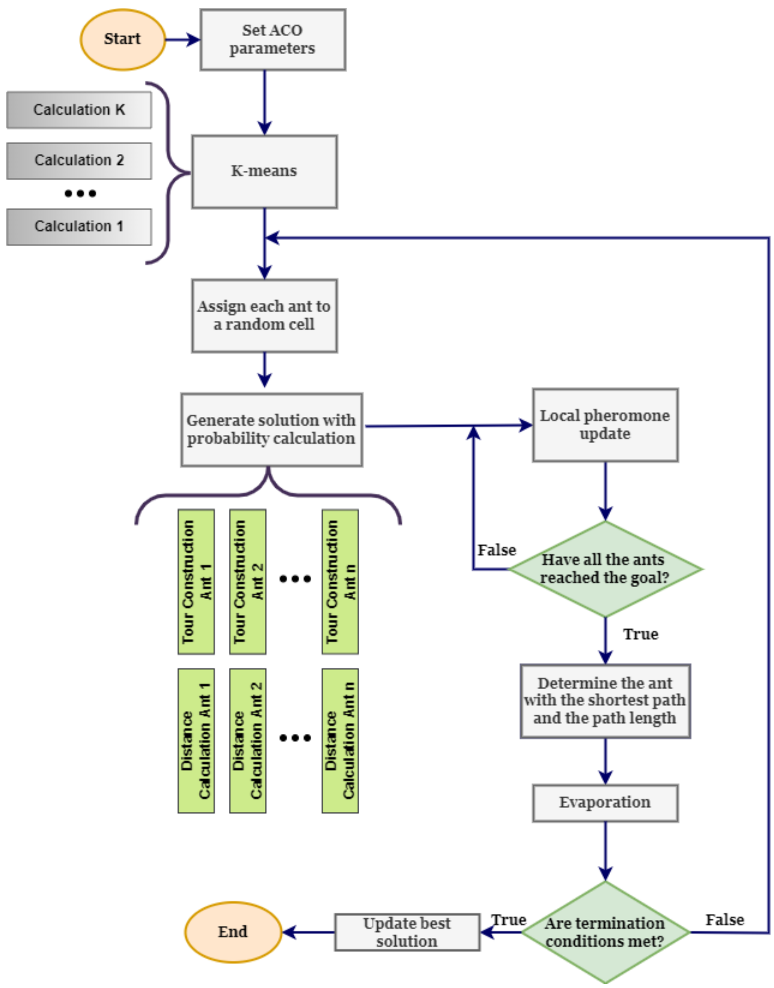

Fig. 3. Flow chart for Proposed Parallel K-EACO

the shortest path in the graph. In this paper, the GPU parallelization is used in calculating the distance of nodes belonging to each center in clustering, while in EACO it is used in determining the tour of the ants. The aim of the proposed study is to design a new optimization model with low computational cost. In the following part, the flow chart of the study is given, and then the step-by-step procedures are explained.

Set ACO Parameters; Determining the parameters in the code is one of the most important parts of the study and also be named Initialization. Critical formulas for both the K-means algorithm and the EACO algorithm were used in the study; these formulas are given in K-Means Clustering and Ant Colony Optimization sections preliminary work done to determine the values that the parameters used in these formulas will take. Especially the number of clusters used, the elitist coefficient in the EACO algorithm, the number of iterations, and colonies are the important ones. Since routing applications using K-means clustering have become popular in recent years, there are quality studies on this subject in the literature. In this manner, [20, 26] studies were examined and it was decided to take the K value of m which is the salesman number in mTSP. In this study, the K value was examined separately by taking 2, 4, and 8. In addition, since it is recommended to take the Elitist coefficient as 0.5 in the published books and articles for the Elitist coefficient, the study was carried out in that direction [25]. The researcher who developed this study has done various research and publication for ACO before, so a preliminary study was made for other important parameters and the parameters were determined accordingly. The values assigned to important parameters as a result of preliminary studies and literature searches are given in Table 1. It was seen that the average results were obtained with 100 iterations is good for the experiment and the results were interpreted over this number of iterations.

Table 1. Parameters

Parameter	Value
α	1.0
β	5.0
ρ	0.5
E_{val}	0.5
initial pheromone	0.01
number of iterations	10, 50, 100, 500, 1000
number of nodes (n)	52, 76, 127, 200, 318, 439
colony size	52, 76, 127, 200, 318, 439, 1002,1024

Parallel K-means Algorithm; It is the first important stage of the study. Clustering is an unsupervised learning algorithm that has been used in machine learning algorithm for years. It has been frequently used by researchers in new problems with its popularity. This study also wanted to benefit from the powerful computation of the K-means algorithm and the appropriate algorithm flow for parallelization. Though to K-means, it is aimed to create clusters consisting of the closest nodes to each other and allow the ant to circulate within these clusters, thus reducing both the ant's circulation and the cost/time. Since the main goal of the study is to calculate the shortest path in the shortest time, the K-means algorithm was also run in parallel.

Parallel K-means algorithm starts with the selection of the center nodes. According to Table I, there are 2, 4, and 8. For each K value, the operations mentioned in the following section were carried out separately. After the center nodes are selected, the distances of the other nodes to each center node are calculated in parallel. It means, the distance of each node to the selected center is performed simultaneously in a parallel manner. In this way, when one center is finished, the time lost by calculating the other is eliminated. The distance calculation was carried out with the Equation 2. After calculating the distances of the nodes to the central nodes, the process of determining the elements of the clusters sequentially was carried out. After all, when clusters are created, the central nodes are chosen again, the distances to these center nodes are calculated in parallel and the clustering process is performed sequentially. It is then compared with the previous clustering results. If there is no change, the K-means algorithm is completed, otherwise, the same processes are repeated by choosing the central nodes again.

– *Finding the closest nodes;* Since the aim is to perform a circulation on all the given nodes, when the K-means algorithm is done and the clusters determined, it is necessary to connect the clusters at the closest nodes. An example of this is given in Fig. 4. The Euclid algorithm given by Equation 2 was used to determine the closest nodes between the clusters. For this process, the distance between the points was calculated only between the clusters belonging to the closest center nodes. The calculation of the distances between the nodes was also carried out in parallel. Then, the closest nodes were determined and each cluster was connected as if there was a bridge between them.

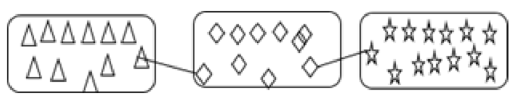

Fig. 4. Connecting Clusters

EACO operations performed with GPU parallelization are shown in Fig. 6.

– *Assign each ant to a random cell;* At this stage, all ants in the colony are randomly assigned to the existing nodes to solve the TSP problem, to start the circulation of finding the shortest path.
– *Generate Solution with Probability Calculation;* This is the second important part of the study. At this stage, three operation performed; Tour construction, finding the closest nodes between clusters, and total tour length of the path. Each of them has been examined separately.
– *Tour Construction;* Another important part of GPU parallelization is the tour construction part. In this part, every ant performs its circulation on the different GPU core. The ant first completes the circulation in the cluster it is in, then moves to the other cluster that closest to the cluster it is in and performs this operation until it completes all the circulations in all clusters. In this way, it performs a circulation between the closest nodes. For this circulation, given steps in Elitist Ant Colony Optimization section which called EACO are followed. Accordingly, the ant assigned to

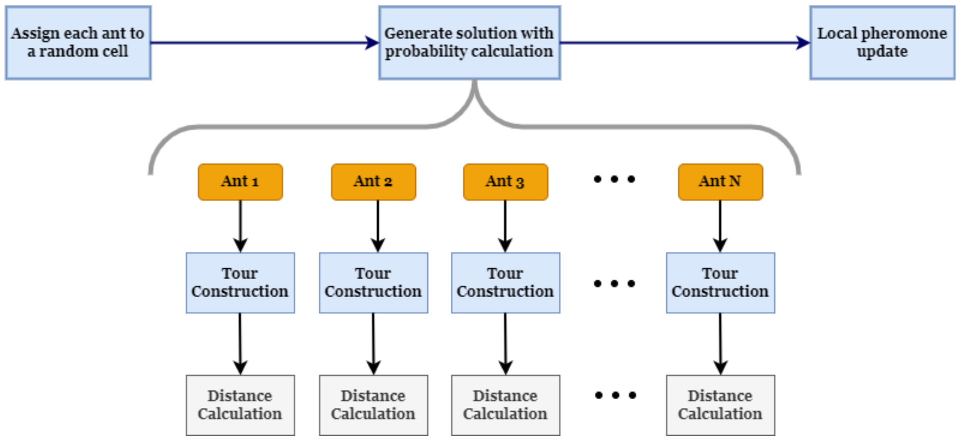

Fig. 5. Parallel Elitist ACO

a random node makes a probability calculation among the nodes it can choose, using the formula given in Equation 6. Then it chooses one of these possibilities with the Roulette Wheel Selection algorithm and moves to the next node. The important part is that the ant first circulates the nodes of the cluster it belongs to, and then moves to the next cluster. Since this process takes place in parallel an ant basis, it provides a great advantage in terms of cost which is time.

– *Distance Calculation;* After tour is completed it is necessary to calculate the fitness function of the ants (in TSP, this is the total path length or the total distance). For this calculation, the Euclidean distance algorithm given in Equation 1 was used and the distance information of each circulation was updated for all ants. Since the aim is to find the shortest path between both the current iteration and all iterations, the length of each circulation is needed. The operations was done with the parallelization on the GPU.

From here on, the following operations were carried out sequentially, meant without parallelization.

Pheromone Update; One of the most important information affecting operations is the pheromone value on the path used while applying algorithms. Through to the pheromone value, the ant decides on the node it will take, so after the circulation on the graph is completed, the pheromones of the paths must be updated again depending on the situation. For the EACO algorithm, calculations were done with Equation 7 and Equation 8.

Determining the shortest path; To determine the shortest path, the circulation paths of all ants are examined, the shortest path is selected and the ant that circulated this path is determined.

Evaporation; Evaporation is an important factor for ants, as the pheromone hormone has a structure that evaporates over time. Accordingly, while applying the EACO algorithm, it is necessary to add the evaporation rate for the work. Equation 5 was used to calculate the evaporation rate of the ants' paths.

4. Experimental Results

In this section, comparison of the developed algorithm with the existing standard method and the accuracy rate of the parallel hybrid algorithm are discussed. For improved Parallel Clustered EACO which is going to be called the K-EACO algorithm, the GPU parallelization which is given in Ant Colony Optimization section is tested and comparisons are made on various TSP libraries ranging in cities from 51 to 1002. Algorithms and parameters were applied as given in the initialization phase which is shown in Table 1. Python programming language have been used for experimental results. Only Standard Python libraries such as Numpy, Pandas, and Numba were used for the proposed algorithm, and the study was created entirely with my codes. Numpy and Pandas libraries are used for accessing dataset, and Numba is used for GPU parallelization. In addition, the ACO algorithm, which was chosen as the comparison algorithm, was also run in parallel. CPU/GPU benchmarks for this algorithm are available in another work by the author. Some of the environmental properties that need to be depicted are shown in Table 2.

Table 2. Environment

Hardware	Features
CPU	Intel(R) Core(Tm) I7-8700 Cpu @3192Mhz, 6 Cores
Op. Sys.	64 bit, Windows 10
Grap.card	NVIDIA GeForce® GTX 1080 Ti Founders Edition 11G
L1/L2/L3 Cache	384 KB/1.5 MB/12.0 MB
RAM	16.00 GB

GPU specifications of the computer can be seen in the Table 3.

Table 3. GPU Specifications

GPU Specifications	
CUDA Cores	3584
Graphics Clock (MHz)	1480
Processor Clock (MHz)	1582
Memory Clock (MHz)	1376
Memory Config	11 GB
Memory Interface Width	352 bit
Memory Interface	GDDR5X
Memory Bandwidth (GB/sec)	11 Gbps

Since the main purpose of this study is to focus on the TSP/mTSP solution, 7 popular datasets in the TSPLIB library were used [29]. These datasets and their optimal solutions are given in Table 4.

Table 4. TSPs and Optimal Solutions

Problem	Number of Cities	Optimal Solution
berlin52	52	7542
eli76	76	538
bier127	127	118282
kroA200	200	29368
lin318	318	42029
pr439	439	107217
pr1002	1002	118282

In the study, ant's circulation time which is the time taken to find the shortest path, and error rate were examined as evaluation criteria. The formula given in Equation 9 was applied for the error rate.

$$ErrorRate = \frac{d - d'}{d'} \tag{9}$$

which $d^{'}$ is the best solution length, d is the solution found by the algorithm. One of the important evaluations in the study is the value of K to be used in the K-means algorithm, that is, the number of clusters. When the literature is examined, the studies on this subject are investigated and seen that the value of K should be 2, 4, and 8 in most of them. In this context, the results of the serial and parallel average execution time (in millisecond) of the K-means algorithm according to the determined number of K values are given in Table 5.

Table 5. Sequential-Parallel K-means Time Comparison

	Time (ms)	
K	Sequential K-means	Parallel K-means
2	0.0150	0.0001
4	0.0245	0.0001
8	0.0309	0.0001

It was seen that clustering worked much faster with GPU parallelization when the results were evaluated. After the parallel K-means algorithm was completed, these clusters were connected at their closest points at each other. The results of the determination of the points and matching operations are added to the runtime results of the developed algorithm.

In the following part, comparisons were made choosing the K value of 8. Table 6 compares the times for finding shortest path of the existing ACO algorithm with the advanced parallel K-EACO algorithm. In Table 6, the evaluations are shown on a different row for each node.

Since the important thing in this study is parallelization, the results of the use of clustering and EACO without parallelization should be compared with those obtained as

Table 6. ACO and K-EACO Time Comparison

Number of	GPU Time (ms)	
Nodes	ACO	K-EACO
52	30.4	16.09
76	48.72	30.88
127	145.16	88.27
200	389.86	197.94
318	1125.96	601.42
439	2489.45	1709.61
1002	22056.11	17746.19

a result of working with GPU. Table 7 shows the running times of Sequential K-EACO (seq.) and Parallel K-EACO (par.) algorithms.

Table 7. Seq. and Par. Time Comparison

Number of	CPU vs GPU Time (ms)	
Nodes	Seq.	Par.
52	200.90	16.09
76	429.21	30.88
127	1301.90	88.27
200	4220.09	197.9
318	13420.64	601.42
439	31337.98	1709.61
1002	22734.96	302206.01

Fig. 6 shows the running time of the parallel and sequential model. When the figure is examined, it is seen that the parallel working model works much faster. The actual running time of the sequential K-EACO (seq.) and parallel K-EACO (par.) algorithms when the K value is 8 shown in Fig. 6. sequential K-EACO algorithm is applied in parallel, it does not give results as fast as K-EACO. The reason for this is that the ant move faster through in a set of points close to each other.

Another criterion as important as time is the error rate. Fig. 7 compares the error rates achieved by applying sequential K-EACO and improved parallel K-EACO to individual dataset. It is seen that the error rates are very close to each other, but the error rate is much lower with the proposed algorithm when the figure is examined.

Fig. 8 shows the shortest path graph reached as a result of circulating with sequential K-EACO the lin318 dataset, which has an average data number, and by running it with the advanced K-EACO algorithm in Fig. 9.

When all the given results and figures are examined, it is seen that the proposed improved algorithm gives much better results both in terms of error rate and time. Although the ACO algorithm was run in parallel, it could not pass the proposed algorithm, because it reduces the error rate and reduces the circulation time by circulating between related points through to clustering.

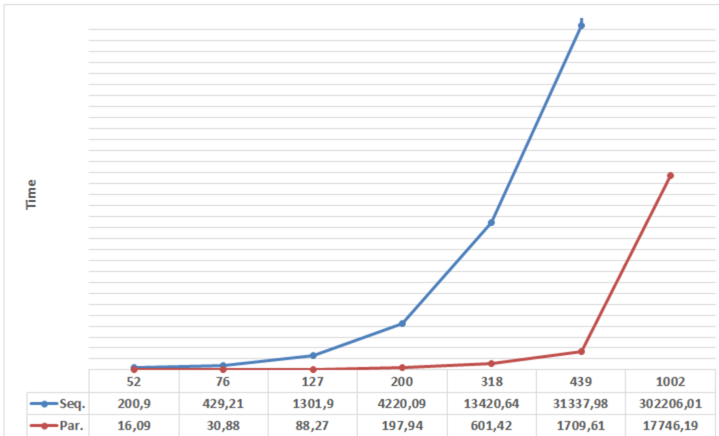

Fig. 6. Time Comparison for seq. and par.

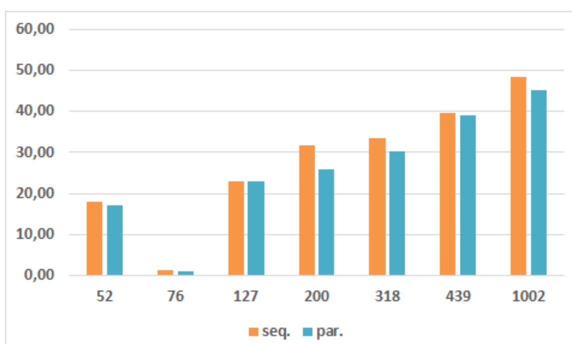

Fig. 7. Error Rate Comparison for seq. and par.

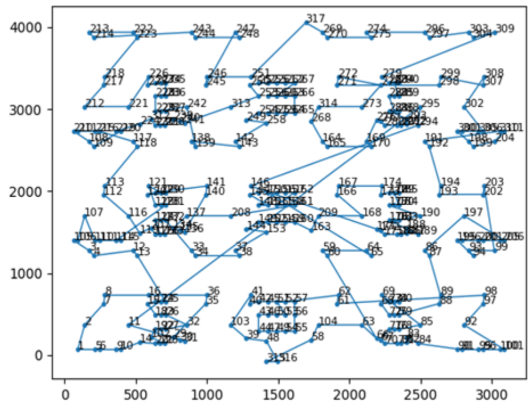

Fig. 8. lin318 with solution sequential K-EACO

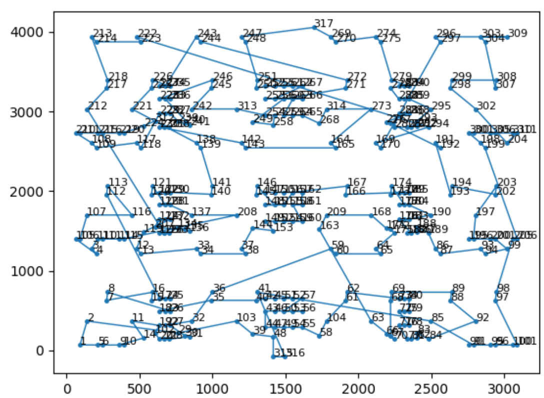

Fig. 9. lin318 with solution parallel K-EACO

5. Discussion and Conclusion

In the study, it is aimed to obtain a reliable algorithm that works faster by using the power of clustering in establishing relationships between data. In addition, since the performance of artificial intelligence algorithms in solving real world problems has increased in recent years, combining them with different algorithms provides more successful results. For this reason, an optimization algorithm that gives fast and successful results is designed by taking the strengths of both algorithms.

Shortest path-finding algorithms have started to draw attention due to the popularization of internet of things devices and the need for GPS use in almost every field. In this manner, TSP is a problem that attracts the attention of researchers and successful GPS suggestions can be made if a solution is found. There are many studies in the literature to solve this problem, but in these studies, there is no method that applies K-means and

EACO by using GPU parallelization like in this article. As a result of the preliminary studies, it has been seen that the parallelization on the GPU works much faster than the CPU and the shortest path on the ACO is found more optimal solution through clustering. Therefore, parallelization in the study was carried out in both K-means and EACO algorithms.

Experimental results were carried out on GPU using various TSP datasets available in TSBLIB. Results on the CPU were not included in the study because it was previously published as another academic study [2]. As a result of the study, it has been seen that the proposed new method has low time complexity and very successful in finding the optimal result, and it is 30% efficient This study will give an idea to other researchers in terms of providing a perspective on the EACO method and showing that it works efficiently with GPU, especially for researchers looking for a new study area. Other optimization algorithms of artificial intelligence can also be considered for the study. In the literature, it has been seen that the genetic algorithm is very successful in PEP studies. Therefore, the use of ACO and Genetic algorithm together can be evaluated in future studies. The research on this subject continues.

References

1. Applegate, D.L., Bixby, R.E., Chvátal, V., Cook, W.J.: The traveling salesman problem. In: The Traveling Salesman Problem. Princeton university press (2011)
2. Baydogmus, G.K.: A parallelization based ant colony optimization for travelling salesman problem. In: 2022 1st International Conference on Information System & Information Technology (ICISIT). pp. 166–169. IEEE (2022)
3. Bora, M., Jyoti, D., Gupta, D., Kumar, A.: Effect of different distance measures on the performance of k-means algorithm: an experimental study in matlab. arXiv preprint arXiv:1405.7471 (2014)
4. Chang, Y.C.: Using k-means clustering to improve the efficiency of ant colony optimization for the traveling salesman problem. In: 2017 IEEE international conference on systems, man, and cybernetics (SMC). pp. 379–384. IEEE (2017)
5. Chen, M.Y., Rubio, J.d.J., Sangaiah, A.K.: Guest editorial-pattern recognition, optimization, neural computing and applications in smart city. Computer Science and Information Systems 18(4), 0–0 (2021)
6. Cheng, C.B., Mao, C.P.: A modified ant colony system for solving the travelling salesman problem with time windows. Mathematical and Computer Modelling 46(9-10), 1225–1235 (2007)
7. Deng, W., Xu, J., Zhao, H.: An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem. IEEE access 7, 20281–20292 (2019)
8. Dinh, Q.T., Do, D.D., Hà, M.H.: Ants can solve the parallel drone scheduling traveling salesman problem. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 14–21 (2021)
9. Dorigo, M.: Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale. Ph.D. thesis, PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy (1992)
10. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE computational intelligence magazine 1(4), 28–39 (2006)
11. Dorigo, M., Blum, C.: Ant colony optimization theory: A survey. Theoretical computer science 344(2-3), 243–278 (2005)
12. Dorigo, M., Stützle, T.: Ant colony optimization: overview and recent advances. Handbook of metaheuristics pp. 311–351 (2019)

13. ESEN, H., Söyler, H., KESKİNTÜRK, T.: Global karınca koloni algoritmasının simetrik ve simetrik olmayan gezgin satıcı problemlerine uygulanması
14. Gutin, G., Punnen, A.P.: The traveling salesman problem and its variations, vol. 12. Springer Science & Business Media (2006)
15. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. Journal of the royal statistical society. series c (applied statistics) 28(1), 100–108 (1979)
16. Jaradat, A., Diabat, W., et al.: Solving traveling salesman problem using firefly algorithm and k-means clustering. In: 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT). pp. 586–589. IEEE (2019)
17. Jünger, M., Reinelt, G., Rinaldi, G.: The traveling salesman problem. Handbooks in operations research and management science 7, 225–330 (1995)
18. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An efficient k-means clustering algorithm: Analysis and implementation. IEEE transactions on pattern analysis and machine intelligence 24(7), 881–892 (2002)
19. Kodinariya, T.M., Makwana, P.R.: Review on determining number of cluster in k-means clustering. International Journal 1(6), 90–95 (2013)
20. Kusumahardhini, N., Hertono, G., Handari, B.: Implementation of k-means and crossover ant colony optimization algorithm on multiple traveling salesman problem. In: Journal of Physics: Conference Series. vol. 1442, p. 012035. IOP Publishing (2020)
21. Latah, M.: Solving multiple tsp problem by k-means and crossover based modified aco algorithm. International Journal of Engineering Research and Technology 5(02) (2016)
22. Likas, A., Vlassis, N., Verbeek, J.J.: The global k-means clustering algorithm. Pattern recognition 36(2), 451–461 (2003)
23. Lin, S.: Computer solutions of the traveling salesman problem. Bell System Technical Journal 44(10), 2245–2269 (1965)
24. Liu, Z., Jiang, G.: Optimization of intelligent heating ventilation air conditioning system in urban building based on bim and artificial intelligence technology. Computer Science and Information Systems 18(4), 1379–1394 (2021)
25. Merkle, D., Middendorf, M., Schmeck, H.: Ant colony optimization for resource-constrained project scheduling. IEEE transactions on evolutionary computation 6(4), 333–346 (2002)
26. Nallusamy, R., Duraiswamy, K., Dhanalaksmi, R., Parthiban, P.: Optimization of non-linear multiple traveling salesman problem using k-means clustering, shrink wrap algorithm and meta-heuristics. International Journal of Nonlinear Science 8(4), 480–487 (2009)
27. Paniri, M., Dowlatshahi, M.B., Nezamabadi-Pour, H.: Mlaco: A multi-label feature selection algorithm based on ant colony optimization. Knowledge-Based Systems 192, 105285 (2020)
28. Rani, S., Kholidah, K.N., Huda, S.N.: A development of travel itinerary planning application using traveling salesman problem and k-means clustering approach. In: Proceedings of the 2018 7th International Conference on Software and Computer Applications. pp. 327–331 (2018)
29. Reinelt, G.: Tsplib—a traveling salesman problem library. ORSA journal on computing 3(4), 376–384 (1991)
30. Skinderowicz, R.: Improving ant colony optimization efficiency for solving large tsp instances. Applied Soft Computing 120, 108653 (2022)
31. Stodola, P., Otřísal, P., Hasilová, K.: Adaptive ant colony optimization with node clustering applied to the travelling salesman problem. Swarm and Evolutionary Computation 70, 101056 (2022)
32. Xiao, J., Li, C., Zhou, J.: Minimization of energy consumption for routing in high-density wireless sensor networks based on adaptive elite ant colony optimization. Journal of Sensors 2021 (2021)
33. Yang, R., Li, D.: Adaptive wavelet transform based on artificial fish swarm optimization and fuzzy c-means method for noisy image segmentation. Computer Science and Information Systems (00), 39–39 (2022)

34. Yildiz, K., Çamurcu, A.Y., Dogan, B.: Comparison of dimension reduction techniques on high dimensional datasets. Int. Arab J. Inf. Technol. 15(2), 256–262 (2018)
35. Yıldız, K., Çamurcu, Y., Doğan, B.: Veri madenciliğinde temel bileşenler analizi ve negatifsiz matris çarpanlarına ayırma tekniklerinin karşılaştırmalı analizi. Akademik Bilişim 10, 248 (2010)

Gozde Karatas Baydogmus was born in Istanbul, Turkey, in 1991. She received the bachelor's degree from the Mathematics and Computer Science Department, Istanbul Kültür University, in 2009, the M.S. degree from the Computer Engineering Department, Istanbul Kültür University, in 2013 and Ph.D. degree the Computer Engineering Department from Marmara University. She continues to work in the field of computer security. In 2015, she completed her master thesis on NoSql Database Testing in Istanbul Kültür University and Ph.D thesis on Intrusion Detection Systems in Marmara University. During her master studies, she worked on distributed databases. She worked as a Research Assistant with the Department of Mathematics and Computer Science, Istanbul Kültür University. She is working as a Assistant Professor as Marmara University in Computer Engineering Department. Her research interests include computer networks and security, machine learning, deep learning, cryptography, python programming, statistics, and graph theory.

Received: August 20, 2022; Accepted: December 01, 2022.