Solving Traveling Salesman Problem using Firefly algorithm and K-means Clustering

Ameera Jaradat
Department of Computer Sciences
Yarmouk University
Irbid-Jordan
ameera@yu.edu.jo

Bara'ah Matalkeh
Department of Computer Sciences
Yarmouk University
Irbid-Jordan
bfmatalkah@hotmail.com

Waed Diabat
Department of Computer Sciences
Yarmouk University
Irbid-Jordan
aldiabatw@gmail.com

***Abstract*—Traveling Salesman Problem (TSP) is a significant combinatorial optimization problem. TSP is NP-hard problem which involves finding the minimum tour length among a given set of nodes and return to the starting node knowing that each node must be visited once. This paper proposes a solution to TSP using Firefly Algorithm (FA) and k-means clustering. The proposed approach comprises three major steps: cluster the nodes, find the optimal path in each cluster, and reconnect the clusters. The first step uses k-means clustering to divide the nodes into sub-problems, the second step utilizes FA to find the optimal path in each cluster, finally reconnect all clusters and return the path between them. The experimental results show that the proposed approach provides better results when compared to other algorithms from the literature.**

Keywords—traveling salesman problem,firefly algorithm,k-means clustering, NP-hard problem

I. Introduction

Travelling Salesman Problem (TSP) is a classical combinatorial optimization problem, easy to describe but hard to solve [1]. TSP consists of salesman and N cities, the goal of this problem is to visit each city by the salesman once with minimum tour length. The challenge in TSP is to find the route that has a minimum distance.

TSP has many applications in vehicle routing and communication networks [2]. Therefore, finding a solution to the TSP is of a great importance. TSP is classified as a hard problem, since finding the shortest route over a set of n cities requires the total comparisons of (n-1)! Permutation vectors [2]. Researchers have attempted to apply variety of heuristic and Meta heuristic approaches. Meta heuristic attempts to find optimal solutions through iteratively enhancing a candidate solution taking in respect a measure of solution quality [2, 3]. Although optimal solution is not guaranteed by these heuristics, a near optimal results can be produced.

The goal of TSP is to find the minimum route among all provided cities, visiting each city exactly once and returning to the point of departure. TSP seems to be a simple problem, but it is one of the most significant classical optimization problems and it has been verified difficult to solve [1, 2]. This paper proposes a new approach to solve the TSP. First, the approach assembles the nodes into a number of clusters. Then, the firefly algorithm is applied to find the minimum rout length in each cluster. Finally, the clusters are grouped together through one global route.

Clustering involves partitioning or grouping a given set of data into disjoint clusters. K-means is a simple unsupervised learning algorithm that was developed to the problem of clustering [6]. K-means has been widely implemented in clustering applications due its effectiveness in producing good results [8, 9].

Firefly algorithm is heuristic practices swarm optimization algorithm that was formulated by Xin-She Yang in 2008 [4]. The firefly algorithm stimulates the flashing activities of fireflies and the phenomenon of bioluminescent communication in solving problems. Firefly algorithm has proven to be effective in solving many optimization problems [5, 10].

TSP, classified NP complete, has attracted many researchers of this field to find optimal tour. Recently, researchers approaches the problem by developing heuristic algorithms [8, 9].

Saloni Gupta and Poonam Panwar [7] provided a solution to TSP using Genetic Algorithm (GA). The proposed approach calculates the distances between all cities to be visited and form a matrix of these distances. Then, the initial city is selected randomly, which is followed by sequence of genetic steps to extract the best path.

Li-Zhuang *et al*. [8] combined Genetic Algorithm (GA) with k-mean clustering method to solve the TSP. The approach involves dividing the cities into several groups by using k-mean clustering. Then, apply Genetic Algorithm on each group to find the sub-path.

Tanasanee Phienthrakul [9] proposed clustering techniques (Gaussian mixer and k-means) to divide the cities to smaller problems by grouping the nearest nodes with each other. Then the optimal path is found in each cluster by applying GA and Ant Colony Optimization (ACO). Finally, clusters are connected to find the optimal path.

The main goal of this work is to propose a new approach for solving travelling salesman problem (TSP). The approach divides the graph into groups of nodes. Then, it uses FA algorithm to generate best routs for each group. Later, these groups are connected together in one global rout.

The rest of this paper is organized as follows: Section II illustrates the firefly algorithm. Section III describes the k-means clustering technique. The proposed method in Section IV. Experimental results in Section V. Finally conclusion in Section VI.

II. FIREFLY ALGORITHM

Firefly algorithm is a meta-heuristic algorithm based on flashing patterns and behavior of fireflies, which was introduced by Xin-She Yang in 2008 [4].Yang applied the following assumptions regarding fireflies [4, 5]:

978-1-5386-7942-5/19/$31.00 ©2019 IEEE

- Any Firefly will be attracted to all other fireflies as fireflies are unisexual.
- The brighter firefly will attract the less bright firefly. However, the light intensity decreases as the distance between the fireflies increases.
- The firefly will move randomly if there is no brighter firefly.

Next, we refer to the Pseudo code of the firefly algorithm that was proposed by Yang, X. S [4] for more detailed description of the firefly optimization. Fig. 1 summarizes the basic steps of the firefly algorithm.

1. Data input and initialization:

Generate an initial population of firefliesXi$(i = 1, 2\ldots m)$, and import the array of dataset. Each firefly generates an initial solution randomly.

The initial value of the light intensity parameter I is initialized with initial path length. Firefly i selects the next city to fly to, its selection probability is associated with the light intensity and path lengths between cities. Firefly i will choose a shorter path, and a city with higher intensity. γ (Gamma) is the absorption coefficient and its value = 1, and r is the distance between two positions which is calculated using Eq. (2).

2. Light Intensity and Attractiveness:

The light intensity l decreases when a firefly moves, and consequently its attractiveness will change. Attractiveness β of a firefly is determined by its brightness which is associated with the encoded objective function, light intensity decreases with the distance from its source, varying with the distance rij between firefly i and firefly j and light that is absorbed in the media.

Light intensity $I(r)$ varies according to the inverse square law $I(r) = Is / r2$, is the intensity at the source.

$$I\,(r) = I_0 e^{-\gamma r^2} \qquad (1)$$

3. Distance and Movement:

The distance between any two fireflies i and j at xi and xj, respectively, is the Cartesian distance:

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad (2)$$

The firefly with the best fitness value is selected to participate in the next phase of optimization.

In case of equal brightness, the Firefly move randomly. The firefly modifies its position using the current position. The movement of a firefly i attracted to another more attractive (brighter) firefly j is determined by:

$$x_i = x_i + \beta_0 e^{\gamma r_{ij}^2}(x_i - x_j) + \alpha(\text{rand} - \tfrac{1}{2}) \qquad (3)$$

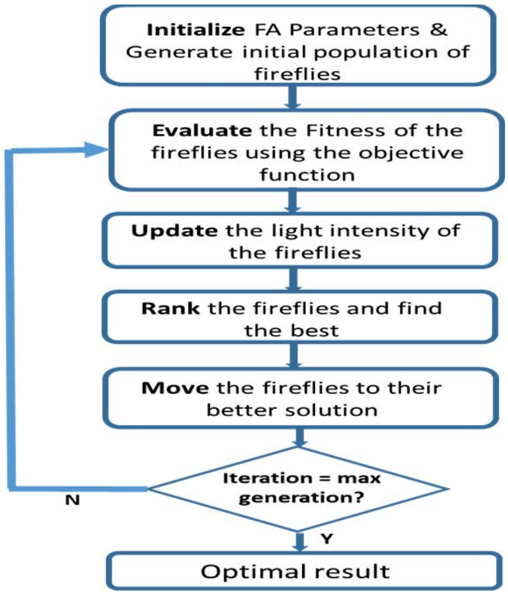

Fig.1. Steps of the Firefly Algorithm

III. K-MEANS CLUSTERING

K-means is considered among the simplest unsupervised algorithms to group a given set of patterns into disjoint clusters [6].The algorithm categorizes the data set into k groups of similarity. Similarity is measured using the Euclidean distance [6, 8].

The main steps of k-mean algorithm can be summarized as follows: first, define k centers, one for each cluster or partition. Next, associate each point in the given data set to the nearest center. Then recalculate k new centroid of clusters resulting from the previous step. Now, apply a new binding between the data set points and the nearest new centers. This step will be repeated until centers do not move any more [6]. Fig.2 illustrates the algorithmic steps of k-means clustering.

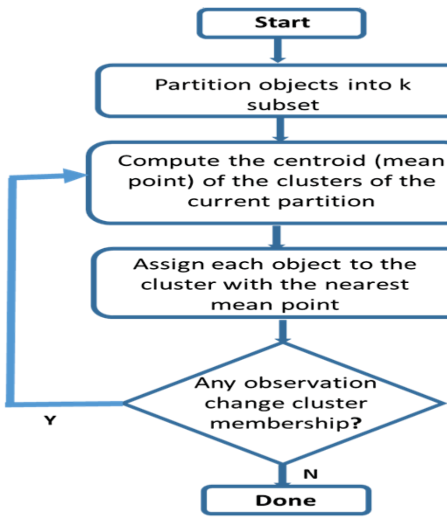

Fig.2. Flowchart of the k-mean clustering method.

IV. PROPOSED METHODOLOGY

In this work, we are proposing a new heuristic approach that combines K-mean clustering algorithm with firefly algorithm to solve TSP.

Let $G(V,E)$ be a weighted graph of nodes representing the cities and edges with weights representing the distances between these cities. The proposed algorithm can be summarized in three basic steps:

- Apply k-means clustering on nodes, this step will divide the nodes into k clusters.
- Apply Firefly algorithm to find the minimal tour in each cluster.
- Connect the clusters with each other and locate the final solution.

Next, we provide a detailed explanation of each step of the proposed algorithm through an illustrative example.

A. K-means Clustering on Nodes

This paper uses k-means clustering because it is a distance-based clustering algorithm. The first step of this method has grouped the nodes in the TSP using k-means clustering technique. The number of clusters is defined as:

$$k = \sqrt{N/2} \tag{4}$$

Where k is the number clusters, N is the number of nodes for TSP. Initially, the centroids are selected randomly from TSP nodes. Each node in the TSP is assigned to the nearest cluster based on the minimum distance between it and the centroids then recalculate the centroid of clusters, as shown in Fig. 3 This will be repeated until the values of centroids are not changed.

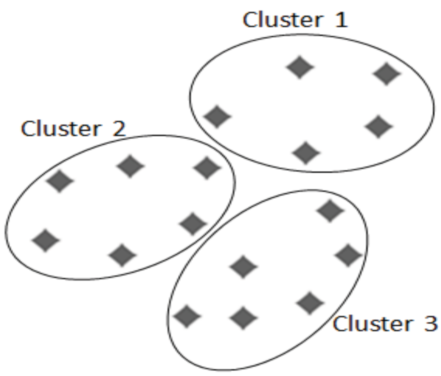

Fig. 3. The application of K-means clustering.

Fig. 1 illustrates the result of applying k-mean clustering on the input graph of (17) nodes, where $k = 3$ according to Eq. (4).

B. Finding Optimal Path in Each Cluster

The optimal path in each cluster will be found after all of the nodes in the TSP are clustered. This paper uses firefly algorithm to find the shortest path in the clusters. FA starts by generating an initial population for the fireflies randomly, where each firefly represents one solution of TSP.

The firefly that has less bright will move towards the brightest firefly based on the brightness of fireflies, after that update the light intensity and find the current best solution. This will continue looping until the maximum iteration is reached. Fig. 4 illustrates the result of applying firefly algorithm to find the minimum tour in each cluster.

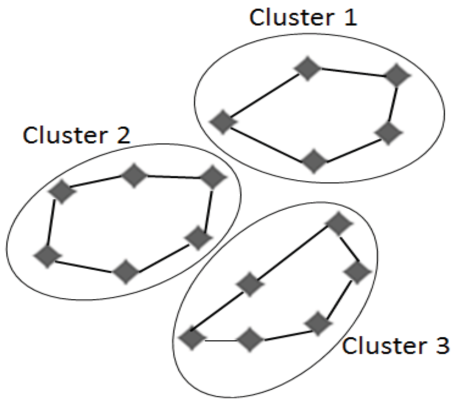

Fig. 4. Finding the optimal path in each cluster.

C. Connect the Clusters

After finding the optimal path in each cluster, the clusters will be connected with each other. At this point, we have k-clusters each contains local minimal tour. The goal is to reconnect these clusters by connecting the endpoints of each tour to produce one global minimal tour.

To connect the tour of the k-clusters we apply the following procedure:

1. Calculate the distances between the centroids of each cluster.
2. Select the two clusters with the minimum distance between their centroids.
3. Combine the selected clusters in one larger cluster.
4. Concatenate the tours of these two clusters.
5. Repeat step 2 until the global minimum tour is generated.

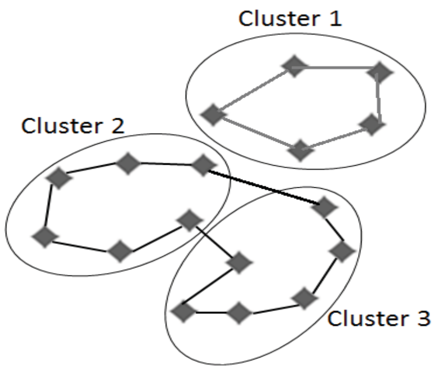

Fig. 5. Cluster reconnection.

Fig. 5 illustrates the concatenation of cluster 2 and cluster 3 as they have a minimum distance between their centroids. Then Fig. 6 shows the final concatenation that leads to the global minimum tour.

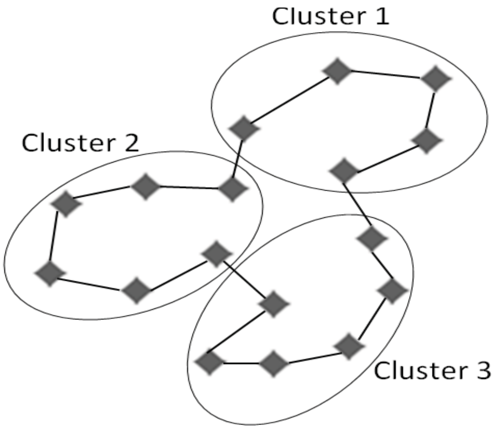

Fig. 6. Final path of Traveling Salesman Problem.

V. Experimental Results

In this paper, we combine K-mean clustering with Firefly to generate a solution to the travelling salesman problem using java eclipse on a platform with the specifications of Intel CORE i3 and 4GB RAM.

To evaluate the proposed methods we tested the algorithm on variety of TSP datasets (eil51, eil76, pr76 and ulysses16) from TSPLIB [11]. These datasets in two-dimensional Euclidean distance. The experiment was repeated 20 times to produce the average and minimum tour length on each dataset. The experimental results were compared against other proposed algorithm that also use clustering approach.

Table 1 shows the resulting tour length upon applying our proposed approach (k-mean clustering and firefly). It also compares these results to the results of applying other algorithm from the literature on the same data set [7, 8, 9].

Results showed that combining k-mean clustering with firefly produces either better or competitive tour in most of the cases. The proposed algorithm scores better results for the data sets (eil76, pr76 and ulysses16). On the other hand, Ant colony algorithm seems to produces better results for the data set (eil51), however, the difference is very small.

TABLE I. Comparison Between Clustering Ant Colony, Firefly and Clustering Firefly Algorithms

Datasets	Optimal Path	Algorithm	Minimum	Average
eil51	426	*Ant Colony using Clustering*	484	–
		Genetic Algorithm using Clustering	484	-
		Firefly Algorithm	435.60	451.37
		Firefly using Clustering	**485.31**	**531.95**
eil76	538	*Ant Colony using Clustering*	624	–
		Genetic Algorithm using Clustering	624	
		Firefly Algorithm	–	–
		Firefly using Clustering	**613.02**	**633.15**
pr76	108159	*Ant Colony using Clustering*	125464	–
		Genetic Algorithm using Clustering	125243	-
		Firefly Algorithm	–	–
		Firefly using Clustering	**112386.74**	**119931.63**
ulysses16	6859	*Ant Colony using Clustering*	7400	7808
		Genetic Algorithm using Clustering	7399	7400
		Firefly Algorithm	7399	7408
		Firefly using Clustering	**7084**	**7326**

In average, firefly algorithm using k-means clustering technique can minimize the length of TSP tour in comparison with clustering ant colony algorithm and firefly algorithm

VI. Conclusion

This paper has proposed Firefly algorithm with clustering technique to solve TSP. The proposed method involves three major steps: First, the nodes are divided into sup problems using k-means clustering method. Then, the firefly algorithm is applied in each cluster to find the optimal path. Finally, the local best paths of all clusters are connected to form one global path that traverse all nodes. The experimental results prove the effectiveness of the firefly algorithm with k-means clustering in providing a competitive solution to the TSP.

References

[1] Rego, César et al. "Traveling salesman problem heuristics: Leading methods, implementations and latest advances." European Journal of Operational Research, vol. 211, pp. 427-441, 2011.

[2] Matai, Rajesh et al. "Traveling Salesman Problem : An Overview of Applications , Formulations , and Solution Approaches," 2012.

[3] J. Kennedy, "Particle Swarm Optimization." Swarm Intelligence vol. 1, pp. 33–57, 2007.

[4] X. S. Yang, "Nature-Inspired Metaheuristic Algorithms." 2008.

[5] X. S. Yang, "Firefly Algorithms for Multimodal Optimization." SAGA, 2009.

[6] V. Faber, "Clustering and the Continuous k-meansAlgorithm, Los Alamos Science, vol. 22, pp. 138-144, 1994.

[7] S. Gupta, and P. Poonam,"Solving travelling salesman problem using genetic algorithm," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3(6), pp. 376-380, 2013.

[8] L. Tan, Y. Tan, G. Yun and C. Zhang, "An improved genetic algorithm based on k-means clustering for solving traveling salesman problem,"Computer Science, Technology and Application: Proceedings of the 2016 International Conference on Computer Science, Technology and Application (CSTA2016), pp. 334-343, 2017.

[9] T. Phienthrakul, "Clustering Evolutionary Computation for Solving Travelling Salesman Problems," International Journal of Advanced Computer Science and Information Technology (IJACSIT), vol. 3(3), pp. 243-262, 2014.

[10] Jaradat, A., Banihamad, "Community Structure Detection Using Firefly Algorithm," International Journal of Applied Metaheuristic Computing, vol. 9(4), pp. 52-70, 2018.

[11] TSPLIB, http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/