



---

## Clustering Evolutionary Computation for Solving Travelling Salesman Problems

---

---

### Authors

**Tanasanee Phienthrakul**

Department of Computer Engineering, Faculty of Engineering,  
Mahidol University

*tanasaee.phienthrakul@mahidol.ac.th*  
NakornPathom, 73170, THAILAND

---

### Abstract

*This paper proposes the methods for solving the traveling salesman problems using clustering techniques and evolutionary methods. Gaussian mixer model and K-means clustering are two clustering techniques that are considered in this paper. The traveling salesman problems are clustered in order to group the nearest nodes in the problems. Then, the evolutionary methods are applied to each cluster. The results of genetic algorithm and ant colony optimization are compared. In the last steps, a cluster connection method is proposed to find the optimal path between any two clusters. These methods are implemented and tested on the benchmark datasets. The results are compared in terms of the average minimum tour length and the average computational time. These results show that the clustering techniques are able to improve the efficiency of evolutionary methods on traveling salesman problems. Moreover, the proposed methods can be applied to other problems.*

---

### Key Words

*Evolutionary Computation, Gaussian Mixer Models, K-means Clustering, Traveling Salesman Problems*

---

### I. INTRODUCTION

Traveling salesman problem (TSP) is one of the most widely discussed problems in the combinatorial optimization researches. This problem has been addressed extensively by mathematicians and computer scientists [1]. The applications of TSP are appeared in planning, scheduling, and searching in scientific and engineering fields, such as vehicle routing, manufacturing, computer operations, integrated circuit testing, cytological testing, and examination timetabling. In the simplest TSP, an optimal path of the shortest length is found on an undirected graph that represents cities or nodes to be visited [1]. The salesman in TSP will

start at a node, visit all other nodes successively only one time each, and finally return to the starting node [1]. This problem is hardly solvable through finding the exact solution directly. Thus, many heuristics and approximation algorithms have been proposed to produce the useable solutions for TSP, such as neural networks [2], [3], simulated annealing [4], genetic algorithm [5],[6], and ant colony optimization [7]-[9] .

Both genetic algorithm and ant colony optimization are the most popular evolutionary algorithms with the diverse ranges of optimization applications. New techniques in genetic algorithm have been developed to solve TSP. In a research of Yang et al. [10], the generalized chromosome in genetic algorithm was proposed. They showed the method to create chromosomes for solving TSP. Liu and Zeng [11] applied reinforcement learning to genetic algorithm. They used this method to solve TSP. Albayrak and Allahverdi [12] solved TSP by genetic algorithm using a new mutation operator. These methods can produce the good solutions of TSP, but they still use a lot of computational time in the large TSP.

To improve the computational time, the concept of clustering is considered. If the nodes in the large TSP are clustered, the TSP problem should be smaller. The computational time of evolutionary computation for solving the smaller TSPs might be reduced. The similar concept can be found in the clustered traveling salesman problem (CTSP) researches [13]. The CTSP consists of determining a least cost Hamiltonian tour or cycle on graph in which the vertices of any clusters are contiguous [13]. There are some researches on CTSP, such as in [14], Anily, Bramel, and Hertz adjusted Christofides' algorithm to get the shortest Hamiltonian path in each cluster and present a 5/3-approximation algorithm for the ordered CTSP. Guttmann-Beck et al., [15] developed the approximation algorithms that guarantee the performance of the ordered CTSP. Ding, Cheng, and He [16] used genetic algorithm to find the shortest Hamiltonian cycle for each cluster and developed genetic algorithm for ordered CTSP. However, the genetic algorithm technique for ordering the CTSP used a lot of computational time. Moreover, this technique did not always generate the good result.

In this paper, clustering algorithms are used to improve the efficiency of evolutionary computation algorithms in TSP. K-means clustering and Gaussian mixer models are compared. Genetic algorithm and ant colony optimization are used as 2 evolutionary computation algorithms in this research. The results are compared in the average minimum tour length and the average computational time. Moreover, this paper proposes a technique for clustered TSP connecting. The basic concepts of clustering and evolutionary computation are briefly reviewed in section II. Section III presents the clustering evolutionary computation algorithm that is the main idea of this paper. The experimentation results are showed in section IV and the conclusion is presented in section V.

## **II. BACKGROUND**

Clustering techniques and evolutionary computation are applied to solve the TSPs. K-means and Gaussian mixer model will be compared in the clustering step. Genetic algorithm and ant colony optimization are used as the standard methods for solving the TSPs. The basic concepts of these algorithms are briefly reviewed in this section.

## A. Clustering Techniques

A cluster is a set of objects in which each object is closer or more similar to every other object in the cluster than to any object not in the cluster. Clustering is a main task of explorative data mining, and a common technique for statistical data analysis. Clustering techniques are used in many fields, including machine learning, pattern recognition, image analysis, and information retrieval [17]. The aim of clustering is finding the useful groups of objects (clusters), where usefulness is defined by goals of the data analysis. For two dimensional data, there are many kinds of clustering techniques, such as prototype-based clustering or centre-based clustering, graph-based clustering, and density-based clustering [17]. In TSPs, the minimum tour length is the main objective of the path finding. Set of nodes or cities that are closer should be grouped to the same cluster. The centre of the cluster can be used as the prototype. Thus, K-means clustering and Gaussian mixture model which are prototype-based clustering will be considered.

1. *K-means Clustering*: Each object in K-means clustering will be assigned to precisely one of a set of clusters [18]. This method of clustering is started by deciding how many clusters, denoted by K. The value of K is generally a small integer. Then, K nodes are selected to be the initial centroids. These nodes can be selected in any way, but this method may be better if K nodes are chosen fairly far apart. There are many ways in which K clusters might potentially be formed. A method to measure the quality of a set of clusters is the sum of squares of the distances between each point and the centroid of its cluster. This value should be as small as possible.

All nodes will be assigned to a cluster by selecting a centroid that gives the minimum distance between node and centroid. Then, there are K clusters based on the original K centroids but the centroids may not be the true centroids of the clusters. Hence, the centroids of each cluster must be recalculated. Each node will be reassigned to the cluster with the nearest centroid. These processes are repeated until the centroids are no longer move. The K-means clustering algorithm and example are showed in Figure I.

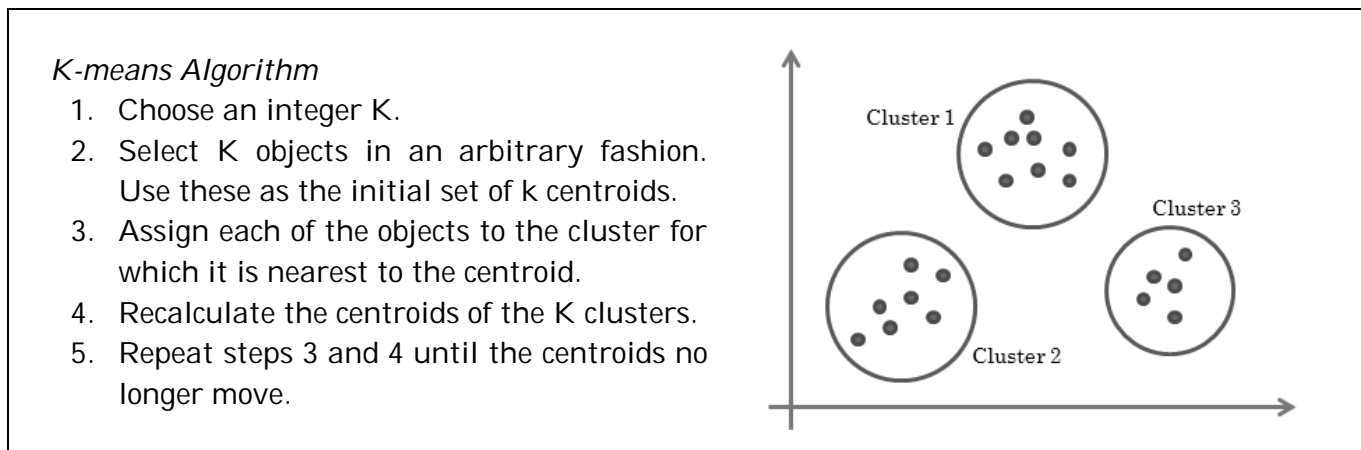


FIGURE I: K-MEANS CLUSTERING ALGORITHM AND EXAMPLE.

2. *Gaussian Mixer Models*: A Gaussian mixer model is a collection of K Gaussian distribution. Each distribution represents a cluster of data points. Gaussian mixer model uses the expectation and maximization (EM) algorithm [19] to fit the Gaussian distributions to the data. The

algorithm is started by defining the number of cluster (K) and chooses the parameter of K Gaussian distributions:

$$\lambda = (\mu_1, \mu_2, \dots, \mu_K, \sigma_1, \sigma_2, \dots, \sigma_K), \quad (1)$$

when each cluster is the normal distribution with  $N(\mu_i, \sigma_i^2)$ .

The EM algorithm calculates the probability that each point belongs to each distribution and then uses these probabilities to compute a new estimate for the parameters. This iteration continues until the estimates of the parameters either do not change or change very little. The EM algorithm for Gaussian mixer model and the example are showed in Figure II.

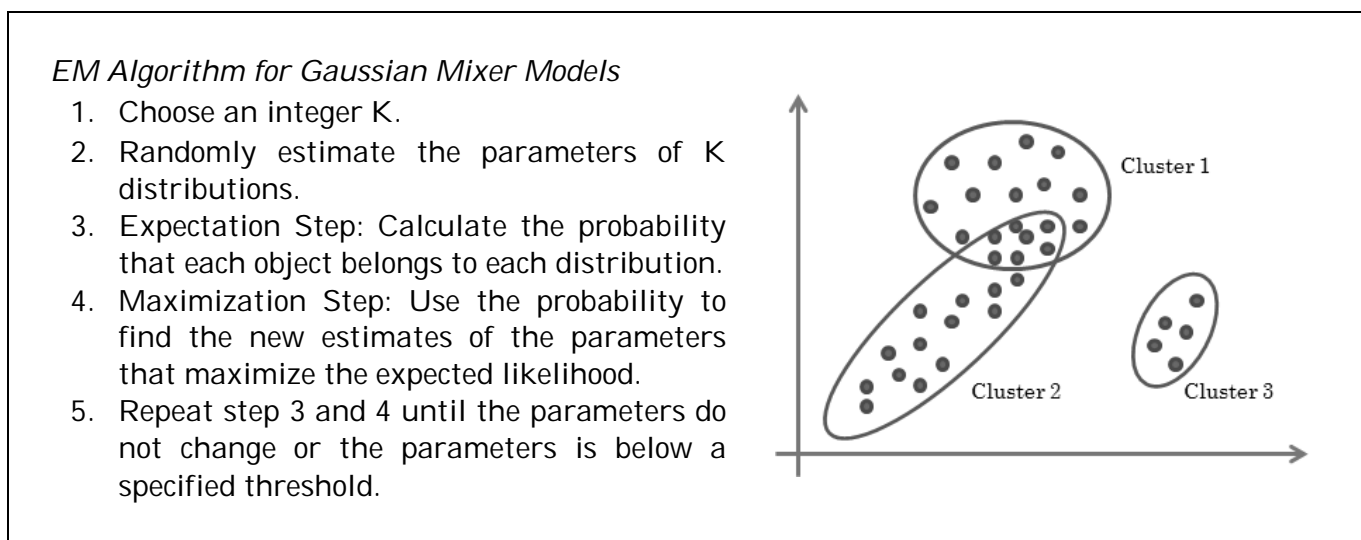


FIGURE II: EM ALGORITHM FOR GAUSSIAN MIXER MODEL AND EXAMPLE.

## B. Evolutionary Computation

Evolutionary computation uses iterative progress, such as growth or development in a population [20]. The population is selected in a guided random search using parallel processing to achieve the desired end. Such processes are often inspired by biological mechanisms of evolution. Evolutionary computation techniques mostly involve meta-heuristic optimization algorithms such as evolutionary algorithms (comprising of genetic algorithms, evolutionary programming, evolution strategy, genetic programming, learning classifier systems, and differential evolution) and swarm intelligence (comprising of ant colony optimization and particle swarm optimization) [20]. Moreover, there are the lesser extent algorithms, such as self-organization such as self-organizing maps, growing neural gas, artificial life (digital organism), cultural algorithms, harmony search algorithm, and artificial immune systems [20]. This paper focuses on genetic algorithm and ant colony optimization that are popular algorithms in evolutionary algorithms and swarm intelligence, respectively.

1. *Genetic Algorithm*: This algorithm was developed by John Holland and his collaborators in the 1960s and 1970s [21]. Genetic algorithm is a model or abstraction of biological evolution based on Charles Darwin's theory of natural selection [21]. Genetic algorithm involves the encoding of solutions as arrays of bits or character strings (chromosomes). The fitness of these chromosomes is evaluated. The creation of a new population is performed by the genetic operators, i.e., crossover, mutation, and fitness proportionate selection. These new chromosomes are selected based on their fitness. The old population is replaced by the new one. These steps are repeated for a number of generations. At the end, the best chromosome is decoded to obtain a solution. Figure III shows the cycle of genetic algorithm.

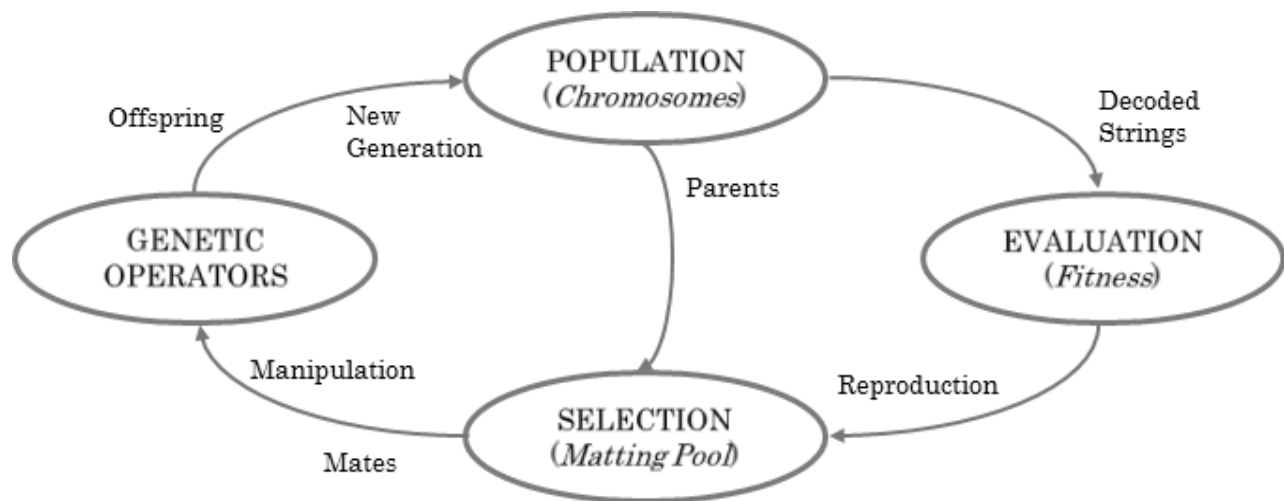


FIGURE III: THE CYCLE OF GENETIC ALGORITHM.

The genetic operators are the essential components of genetic algorithms as a problem-solving strategy. Crossover operator generates 2 offspring chromosomes from parents' chromosomes which bring about diversity characteristic of new child. There are many techniques of crossover operator operators such as single point, two point, and order crossover. Mutation operator is used to prevent falling all solutions in population into a local optimum of solved problem. Mutation changes randomly the new offspring. Selection operation is a selection of chromosomes to be the chromosome in the next generation. The best chromosome should survive and become the original breed for the next generation. Chromosomes with good fitness are probably selected. There are many methods how to select the best chromosomes, for example, Roulette wheel selection, rank selection, and tournament selection. Elitism copies the best members of the parent population to the next generation. Elitism has proved to increase speed of performance of genetic algorithm because the best found solution is preserved to the child generation [22].

2. *Ant Colony Optimization*: The first ant colony optimization system was introduced in 1992, and was called ant system [23]. The natural metaphor of ant algorithms are based the ant colonies [7]. Real ants are capable to find the shortest path from a food source to their nest by exploiting pheromone information. While walking, ants deposit pheromone on the ground and follow, in probability, pheromone previously deposited by other ants. These pheromone can be evaporated depend on the time. Hence, the shorter path will have more accumulated pheromone. Figure IV presents how real ants find a shortest path.

In Figure IV-A, ants arrive at a decision point. Some ants choose the upper path and some ants go to the lower path. These choices are randomly selected. Since ants move at approximately constant speed, the ants which choose the lower path reach to the opposite site faster than those which choose the longer path. Pheromone accumulates at a higher rate on the shorter path. The number of dashed lines in Figure IV-D is approximately proportion of the pheromone that deposited by ants. Since the lower path is shorter than the upper one, more ants will visit it. Therefore, the pheromone accumulates faster. After a short transitory period, the difference in the total of pheromone on the two paths is sufficiently large. This pheromone is influent to the decision of the news coming ants in the system. Very soon, all ants will use the shorter path [7]. This concept can be implemented as an optimization algorithm and it can be implemented for traveling salesman problems.

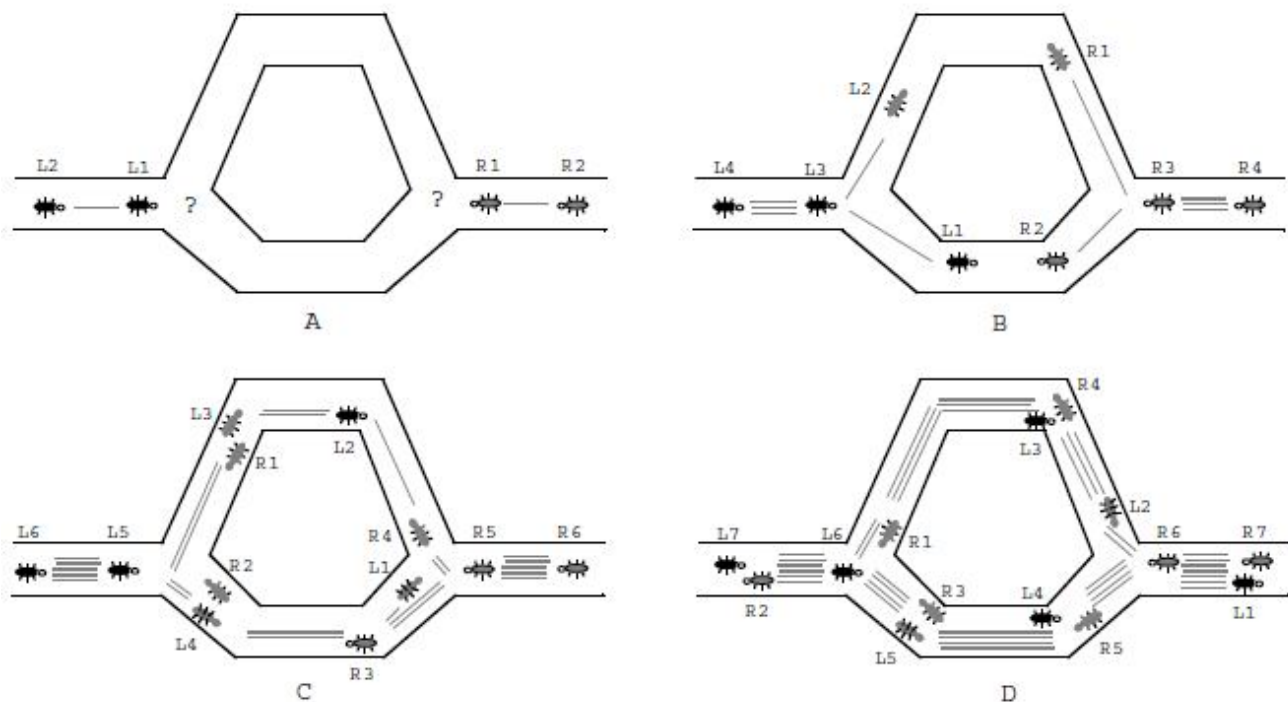


FIGURE IV: PATH FINDING OF REAL ANT.

Informally, the ant colony system works by  $m$  ants. These ants are initially positioned on  $n$  cities. The positions are chosen according to some initialization rule (e.g., randomly). Each ant builds a tour or a feasible solution to the TSP by repeatedly applying a random greedy rule or the state transition rule. While constructing a tour, an ant also modifies the amount of pheromone on the visited edges by applying the local updating rule. The amount of pheromone on edges is modified again by applying the global updating rule when all ants terminate their tour. In building the tours, ants are guided by both heuristic information and by pheromone information. An edge with a high amount of pheromone is a very desirable choice.



### III. CLUSTERING EVOLUTIONARY COMPUTATION

This paper proposes to use clustering techniques, to improve the performance of the evolutionary computation algorithm, i.e., genetic algorithm and ant colony optimization, on the traveling salesman problems. This section presents the steps of solving the TSP, which can be divided into three main steps, i.e., node clustering, optimal path finding, and cluster connecting. The flow of clustering evolutionary computation algorithm is showed in Figure V.

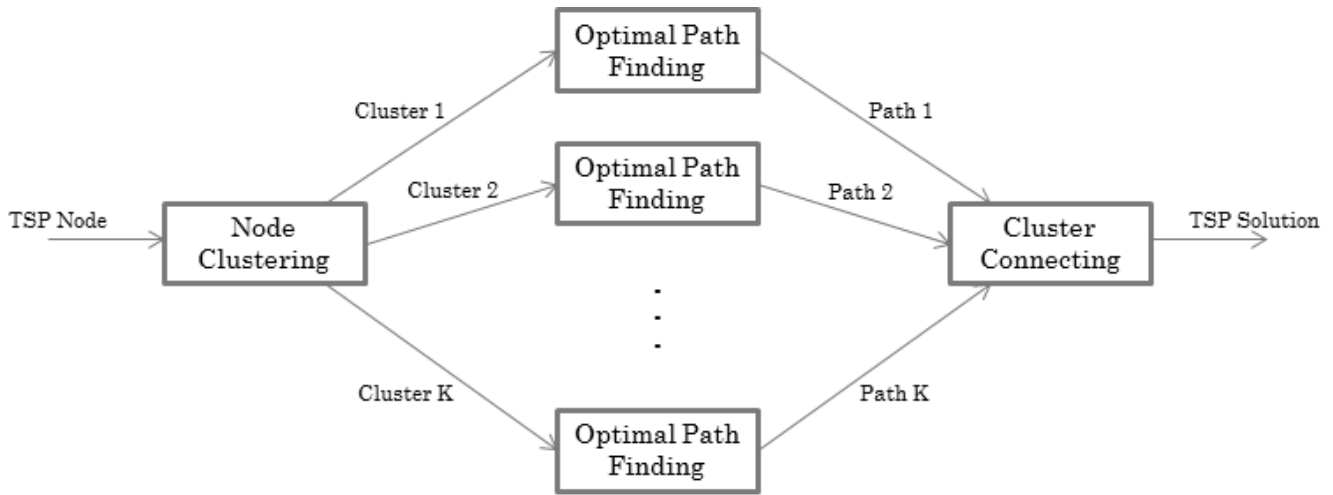


FIGURE V: CLUSTERING EVOLUTIONARY COMPUTATION STEPS.

#### A. Node Clustering

In the first step, nodes in the TSP are grouped by a clustering technique. By the concept, any clustering algorithms can be used in this step. However, in this paper, K-means clustering and Gaussian mixer models will be considered because they are the distance-based clustering algorithms, which the centroids of each cluster can be easily calculated. For a TSP with  $N$  nodes, the number of clusters ( $K$ ) is defined as (2).

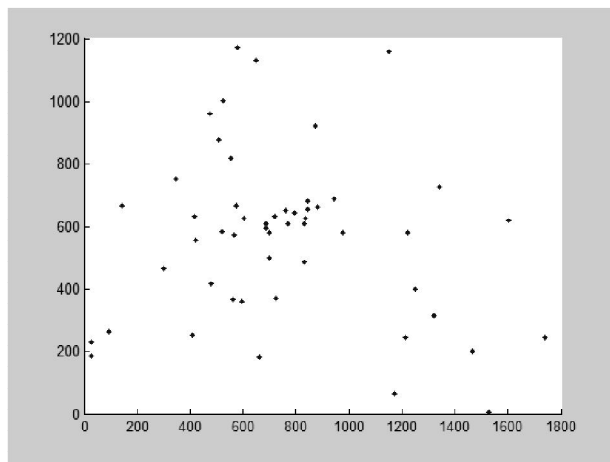
$$K = \left\lceil \sqrt{N/2} \right\rceil \quad (2)$$

K-means clustering is one of the simplest unsupervised learning algorithms. This algorithm performs an iterative alternating fitting process to form a specific number of clusters. In the first step, a set of  $K$  points are selected to be a first guess of the means of the clusters. These points are called cluster seeds or centroids. Each node is assigned to the nearest seed to form a set of temporary clusters. Then, the seeds are replaced by the cluster means and the points are reassigned. This process continues until no further changes occur in the clusters. Hence, by the K-means clustering, the shape of cluster looks like the sphere.

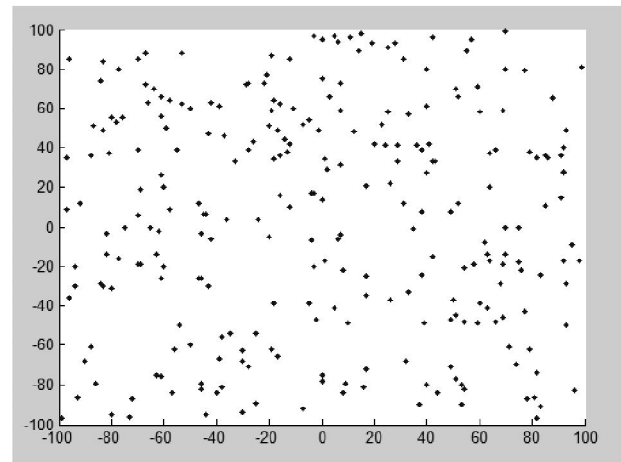
In the Gaussian mixture models, the shape of cluster can be sphere or ellipse. The probability density functions of nodes are used to create the clusters. Usually a mixture model with full

covariance matrices will need fewer components to model a given density, but each component has more adjustable parameters. The EM algorithm calculates the probability that each point belongs to each distribution and then uses these probabilities to compute a new estimate for the parameters. These iteration processes are continued until the parameters are not changed.

In the Figure VI, two traveling salesman problems, Berlin52 and Gil 262 [24], are visualized. The outputs of K-means clustering and Gaussian mixer model are showed in Figure VII and Figure VIII. These outputs show that there are the differences between the shapes of clusters in K-means and Gaussian mixer models when the number of clusters is predefined. The shapes of clusters in Gaussian mixer model are flexible and some clusters are ellipse. In the Gaussian mixer model, the sizes of clusters are different because they are clustered by the density of nodes, while K-means clustering will consider only the distance between node and centroid.

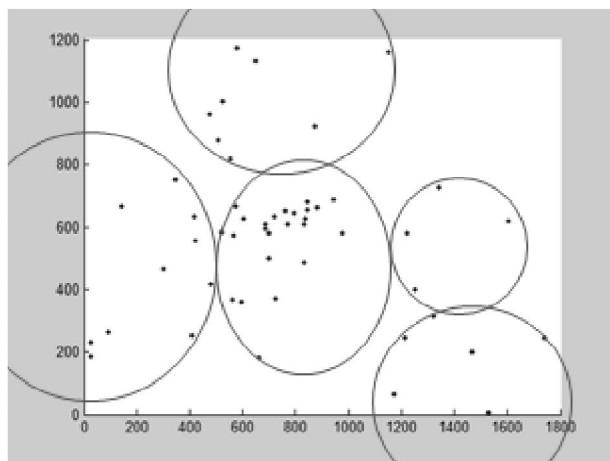


(a) Berlin 52

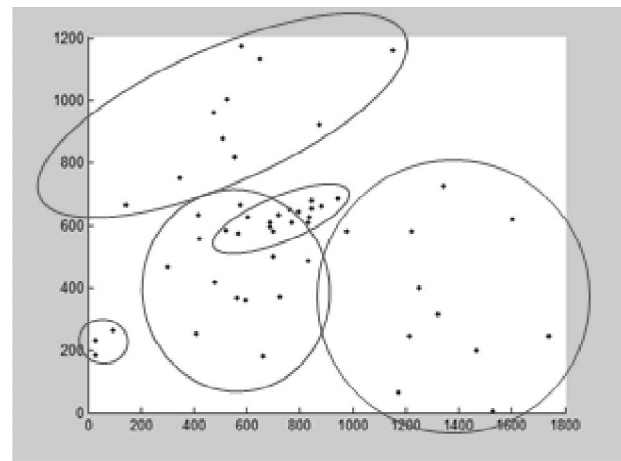


(b) Gil262

FIGURE VI: TWO TRAVELING SALESMAN PROBLEMS.



(a) K-means Clustering



(b) Gaussian Mixer Model

FIGURE VII: THE CLUSTERS OF BERLIN52.



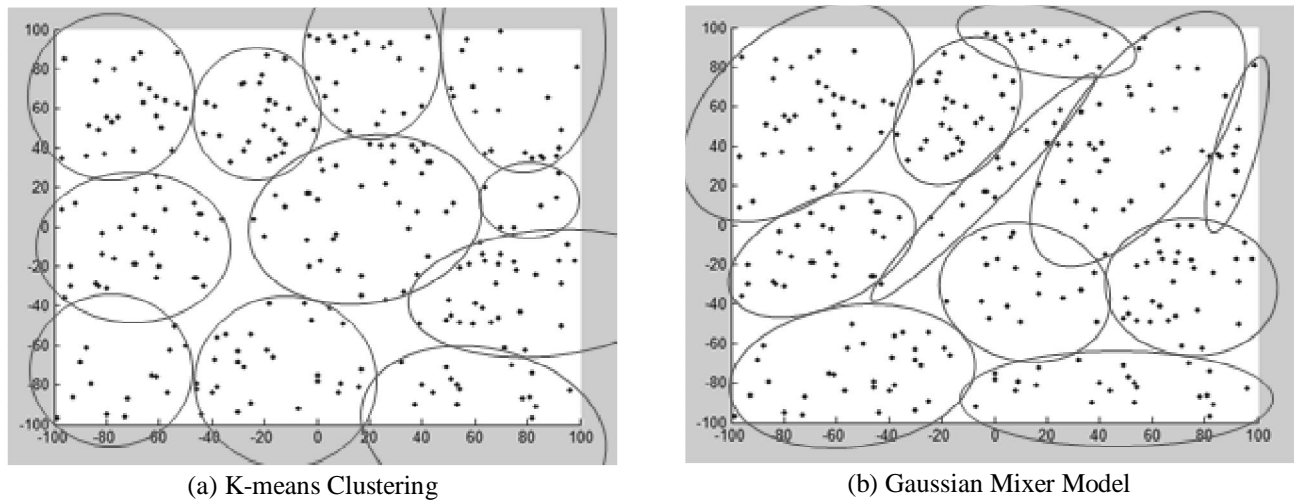


FIGURE VIII: THE CLUSTERS OF GIL262.

## B. Optimal Path Finding

After the nodes in the traveling salesman problems were clustered, the shortest path in each cluster will be searched. There are many techniques that can be used for this step. However, for the large TSP, the deterministic methods may take a lot of time. The evolutionary computation is a good choice. Genetic algorithm and ant colony optimization are the most popular evolutionary algorithms, which are widely used in many research areas. The genetic algorithm is simple and effective, while the ant colony optimization yields the good results on TSP. In this paper, the genetic algorithm is implemented according to the evolutionary theories. Permutation encoding is used to represent a chromosome that is a solution of TSP. The values in chromosomes are the order of cities which salesman will visit them. Figure IX (a) shows some examples of chromosomes in a population.

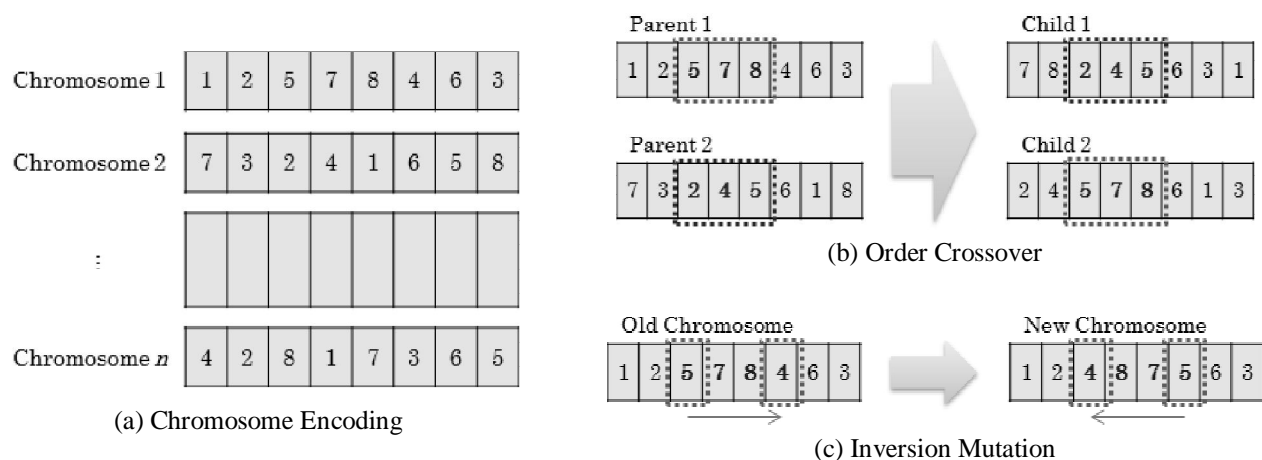


FIGURE IX: CHROMOSOMES AND GENETIC OPERATORS.

Only the most suitable chromosomes in a population are survived. These chromosomes are used to generate offspring, thus their biological characteristics are transmitted to new generations. The population of solutions are created by the genetic operators, i.e., order crossover and inversion mutation, that are shown in Figure IX (b) and Figure IX (c), respectively. Solutions are selected according to their fitness value to form new offspring. These processes are repeated until a fixed number of generations.

An alternative method, ant colony optimization, is implemented to solve the TSP in each cluster. Each ant constructs a TSP solution in an iterative way; it adds new cities to a partial solution by exploiting both information gained from past experience and a greedy heuristic. Ant colony optimization will be compared to the clustering genetic algorithm. Figure X shows an example of the optimal path finding in each cluster. Eleven nodes (A-K) are clustered to 3 groups and three optimal paths are founded for these three clusters. The centroid of each cluster will be used for the cluster connecting step.

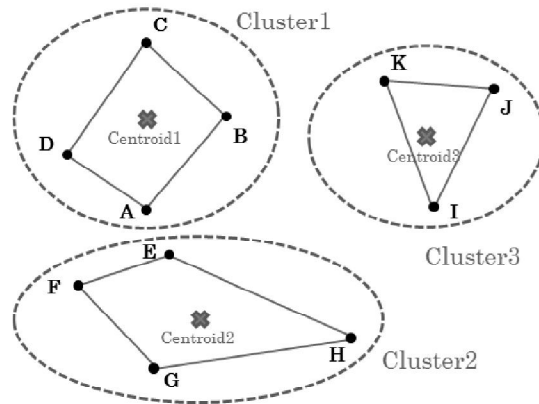


FIGURE X: THE OPTIMAL PATH FINDING IN EACH CLUSTER.

### C. Cluster Connecting

To connect all  $K$  paths from  $K$  clusters, a method for selecting the clusters and nodes is proposed. This method can be divided into 3 steps, i.e., the cluster selection, the next cluster selection, and the last node connection.

1. *The Cluster Selection:* In this step, the Euclidean distances of any two clusters are compared. All pairs of centroids will be used to calculate the distances. The shortest distance will be selected. If there are  $K$  clusters, the number of distance calculation will be (3).

$$\text{Number of Distance Calculation} = \frac{K!}{(K-2)! 2!} \quad (3)$$

Figure XI shows an example of the distance calculation on 3 pairs of centroids. The shortest distance is the distance between Cluster1 and Cluster2. Hence, these clusters will be used for choosing the nearest nodes.

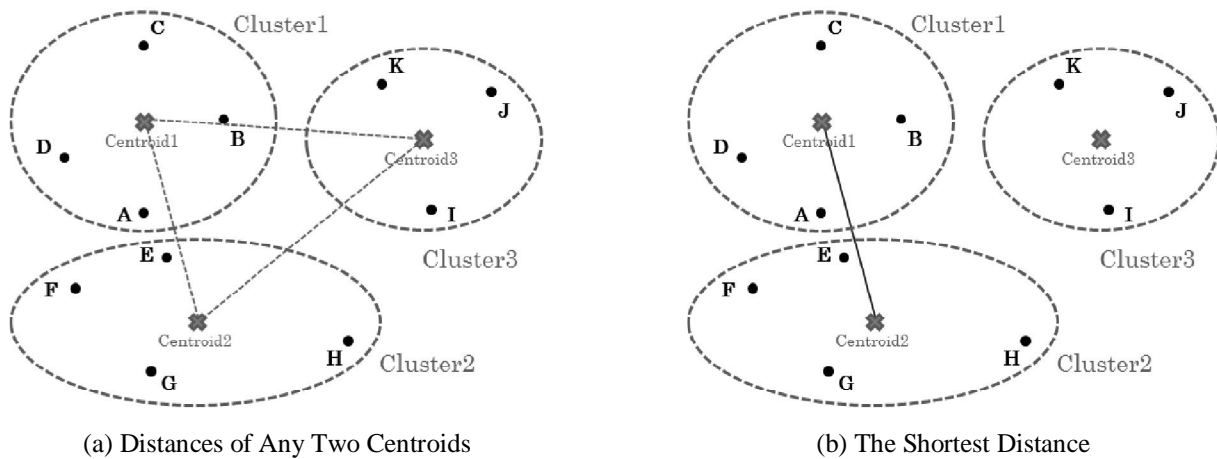


FIGURE XI: EXAMPLES OF THE DISTANCE CALCULATION AND THE SHORTEST DISTANCE.

Then, the distances between Centroid1 and all nodes in Cluster2 are calculated. The nearest node from Cluster2 will be selected, which it is node E in Figure XII (a). In the same way, the distances between Centroid2 and all nodes in Cluster1 are calculated. The nearest node from Cluster1 is selected. A path between these two nodes will give the minimum distance for two clusters connection. From the example, node A in Cluster1 will be connected to node E in Cluster2.

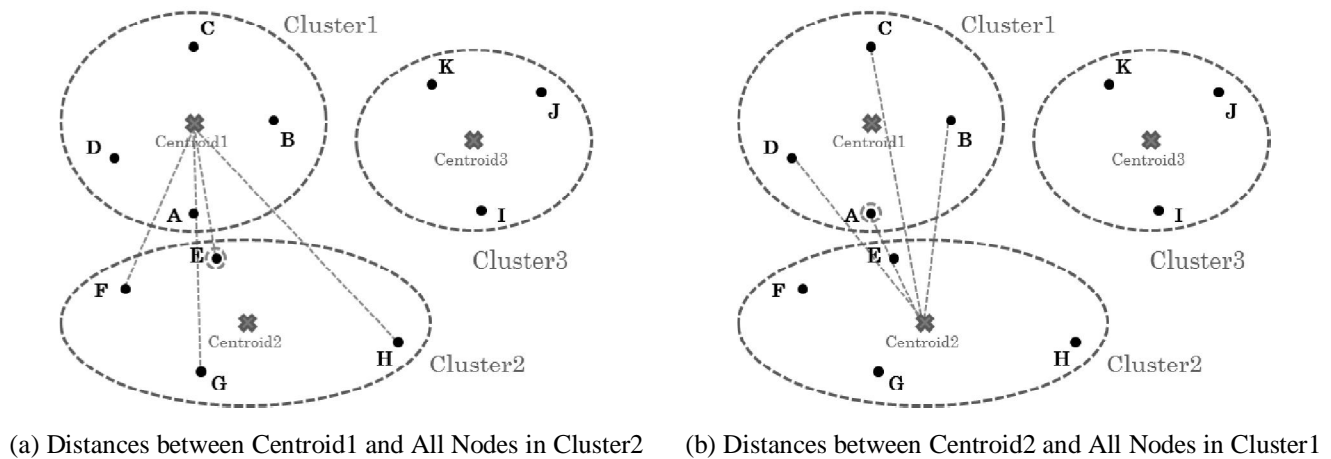


FIGURE XII: THE NEAREST NODES SELECTION.

For node A and node E, the distance to the next left node and the distance to the next right node will be compared. The nearest one will be selected. For example in Figure XIII (a), the distance between A and D is shorter than the distance between A and B. Therefore, node A will be connected to node D, and node D will be connected to another nodes according to the order in the optimal path from the previous step. For cluster2, node F will be connected to node E, and then node F will be connected to another nodes. Then, paths from Cluster1 and Cluster2 are connected as showed in Figure XIII (b).

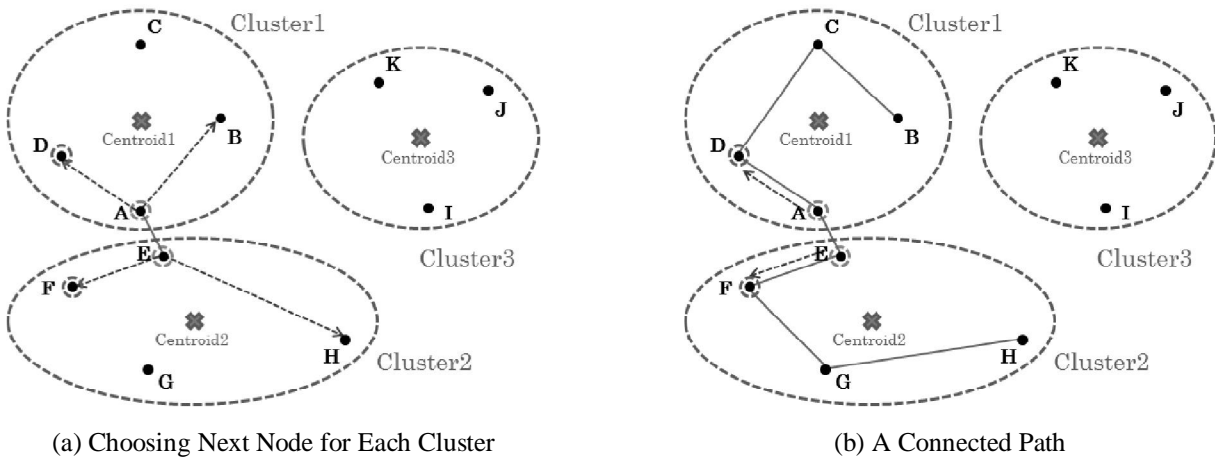


FIGURE XIII: EXAMPLES OF TWO PATHS CONNECTING.

2. *The Next Cluster Selection:* From the previous step, there are two end-nodes that are node B and node H from Cluster1 and Cluster2, respectively. These nodes will be connected to other clusters by choosing the shortest distance between nodes and centroids. The next centroids will be selected by comparing the distances between two end-nodes and all remained centroids. This step must be repeated until the last centroid is selected as showed in Figure XIV (a).

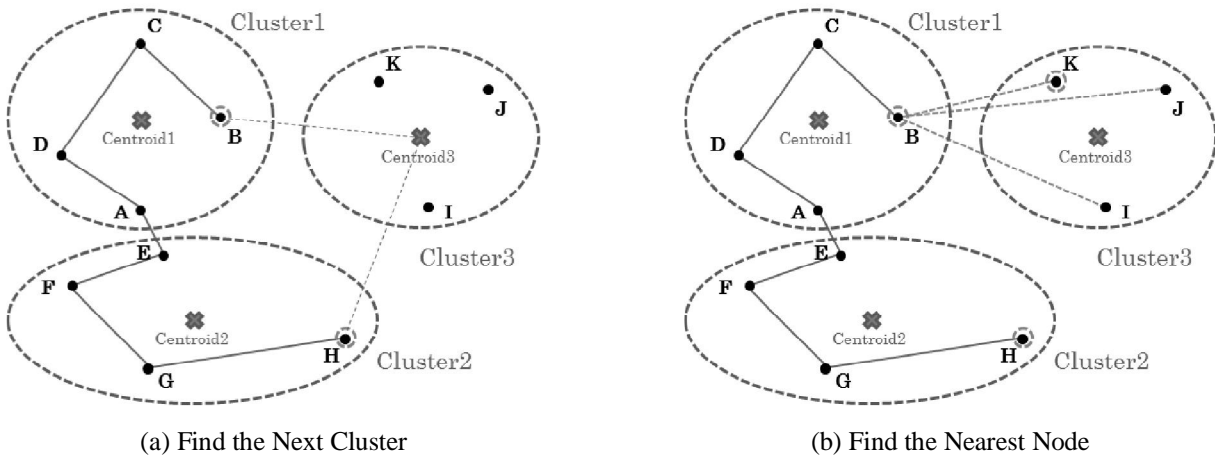


FIGURE XIV: THE NEXT CLUSTER SELECTION.

If there are  $K$  clusters, this step will be repeated  $K - 2$  times. The number of distance calculation for all  $K - 2$  repetitions will be (4).

$$\begin{aligned}
 \text{Number of Distance Calculation} &= 2(K - 2) + 2(K - 3) + \dots + 2 \\
 &= 2[(K - 2) + (K - 3) + \dots + 1] \\
 &= 2\sum_{i=1}^{K-2}(i) = (K - 2)(K - 1)
 \end{aligned} \tag{4}$$

From Figure XIV, only Cluster3 have not been selected, thus the distance from node B and node H to Centroid3 will be compared. The distance between node B and Centroid3 is shorter than that of node H. Then, node B will be connected to the nearest node in Cluster3. The distances between node B and all nodes in Cluster3 are compared. Node K in Cluster3 gives the shortest distance. Therefore, node B will be connected to node K as showed in Figure XIV (b). From node K, the distances to the neighbour nodes will be compared. As showed in Figure XV (a), a shorter distance is selected, and the solution path will go along this direction.

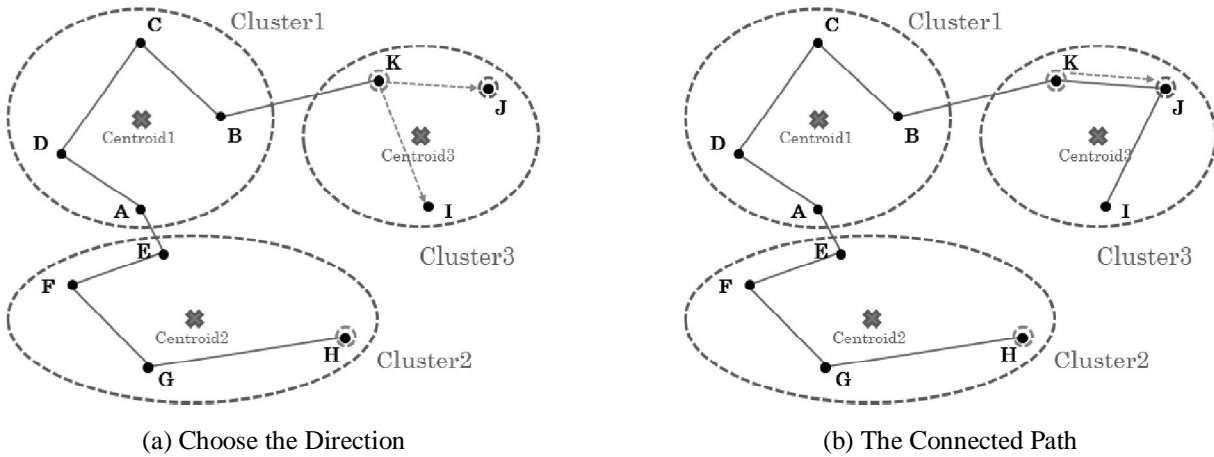


FIGURE XV: THE NEXT CLUSTER SELECTION.

**3. The Last Node Connection:** At the last cluster, when there are no remaining clusters, two end-pointed will be linked. A complete TSP solution will be appeared as showed in Figure XVI. This method may not guarantee the shortest path, because the greedy methods are applied in many steps. However, when the clustering techniques were applied, the TSP sub-problems should be easier to be solved in a specific time.

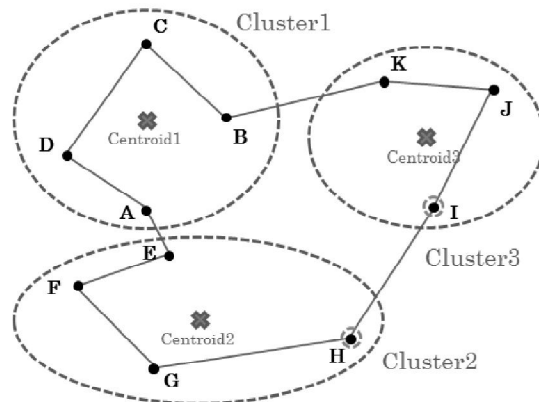


FIGURE XVI: A COMPLETE TSP SOLUTION.

#### IV. EXPERIMENTATION

In order to evaluate the proposed methods, 10 TSP datasets from TSPLIB [24] are tested. These 10 datasets are symmetric TSPs in the two-dimensional Euclidean distance. The number of cities and the optimal tour length of these datasets are shown in Table I.

TABLE I: 10 DATASETS FOR TESTING

No.	Dataset	#Cities	Optimal Length	No.	Dataset	#Cities	Optimal Length
1	eil51	51	426	6	kroB200	200	29,437
2	berlin52	52	7,542	7	gil262	262	2,378
3	eil76	76	538	8	lin318	318	42,029
4	pr76	76	10,8159	9	pcb442	442	50,778
5	kroE100	100	22,068	10	rat783	783	8,806

Each experiment will be repeated ten times on each datasets to find the average minimum tour length and the average computational time. The results will be compared between simple genetic algorithm and the clustering genetic algorithms that use K-means clustering and Gaussian mixer models. Furthermore, these results will be compared to the ant colony optimization.

For the parameter settings, the number of cluster is defined as  $\lceil \sqrt{N/2} \rceil$  on both K-means clustering and Gaussian mixer model, where  $N$  is the number of cities in a TSP. For each cluster that is a TSP sub-problem, the parameters of genetic algorithm and ant colony optimization are defined as Table II, where  $n$  is the number of nodes in a cluster.

TABLE II: THE PARAMETER SETTINGS FOR GENETIC ALGORITHM AND ANT COLONY OPTIMIZATION

Genetic Algorithm's Parameters	Value	Ant Colony Optimization's Parameters	Value
Length of Chromosomes	$n$	Number of Ants	$n$
Number of Chromosomes	$3n$	Maximum Time	10
Number of Generations	$5n$	Initial Pheromone Value	1
Crossover Rate	0.9	Influence of Pheromone on Direction ( $\alpha$ )	3
Mutation Rate	0.8	Influence of Adjacent Node Distance ( $\beta$ )	7
Selection Method	Tournament	Pheromone Decreasing Factor ( $\rho$ )	0.03332
Elitism Rate	0.2	Pheromone Increasing Factor ( $\nu$ )	0.03332



The experimental results of simple genetic algorithm and clustering genetic algorithms are compared in Table III. The average minimum tour length and the average computational time are illustrated. The results show that most of the average minimum tour lengths of the clustering genetic algorithms, except only eil51, are better than that of the simple genetic algorithm. For eil51, the cities in this problem are randomly distributed and the number of cities is not too much. Therefore, the genetic algorithm can be used to solve that problem and yields a better result in a time limitation. For pcb442 and rat783, these problems are large and their computational times are more than 10,000 seconds. Therefore, the average minimum tour length under the pre-condition cannot be illustrated since they are time out.

TABLE III: THE RESULTS OF CLUSTERING GENETIC ALGORITHM

No.	Dataset	Average Minimum Tour Length			Average Computational Time (Sec.)		
		Genetic Algorithm	Clustering		Genetic Algorithm	Clustering	
			K-means GA	Gaussian GA		K-means GA	Gaussian GA
1	eil51	<b>477</b>	484	484	3.58	<b>3.36</b>	3.85
2	berlin52	8,586	<b>8,416*</b>	8,439*	<b>3.67</b>	3.81	4.35
3	eil76	641	<b>624*</b>	<b>624*</b>	12.93	<b>3.55</b>	4.14
4	pr76	129,456	125,243**	<b>124,857**</b>	12.93	<b>3.76</b>	4.25
5	kroE100	30,119	<b>25,918***</b>	26090***	32.56	<b>4.04</b>	4.46
6	kroB200	53,838	34,879***	<b>34,816***</b>	380.15	<b>5.69</b>	5.97
7	gil262	4,569	2,801***	<b>2,781***</b>	1,027.73	<b>7.33</b>	8.13
8	lin318	88,128	51,746***	<b>51,665***</b>	2,097.93	<b>11.03</b>	12.81
9	pcb442	-	<b>63,851</b>	63,970	>10,000.00	<b>17.03</b>	18.24
10	rat783	-	<b>14,370</b>	14,421	>10,000.00	41.04	<b>39.86</b>

\* Significantly difference from the genetic algorithm at 90% confident interval.

\*\* Significantly difference from the genetic algorithm at 99% confident interval.

\*\*\* Significantly difference from the genetic algorithm at 99.9% confident interval.

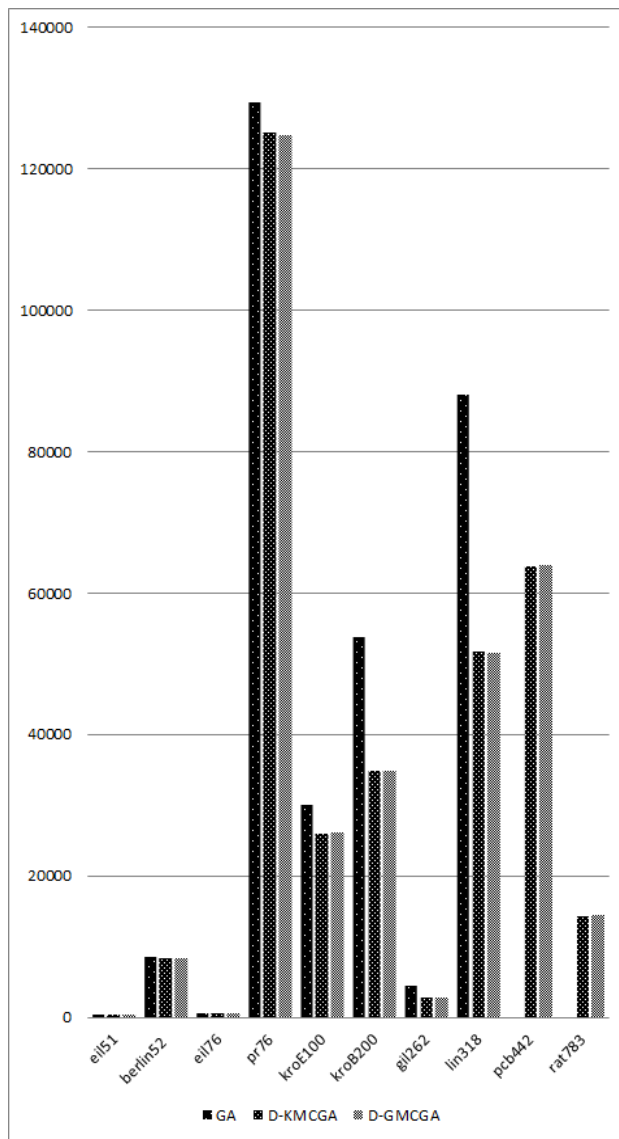
Also, the results in Table III show that when the sizes of problems are increased, the results of the cluster genetic algorithms are more significantly differed from the simple genetic algorithm. On kroE100, kroB200, gil262, and lin318, the average minimum tour lengths of the clustering genetic algorithms are significantly better than those on the genetic algorithm at 99.9% confident interval. Moreover, in obviously, the average computational times of the proposed methods are shorter than those of the simple genetic algorithm when the problems are large.

When the clustering techniques are compared, the results of K-means clustering is better than the results of Gaussian mixer models on 4 datasets, draw on 2 datasets, and Gaussian mixer models are better than K-means clustering on 4 datasets. For the computational time, the average times of K-means clustering is slightly better than the average times of Gaussian mixer models. Hence, when the concept of clustering genetic algorithm is used, both K-means and Gaussian mixer can be applied in the clustering step. If the data are randomly distributed, K-means clustering will be more suitable than Gaussian mixer model because it can reduce the computational time and give the better results. In the opposite, Gaussian mixer model is better than K-means when there are patterns in the distribution of data or the cities are dense in some areas of the input spaces.

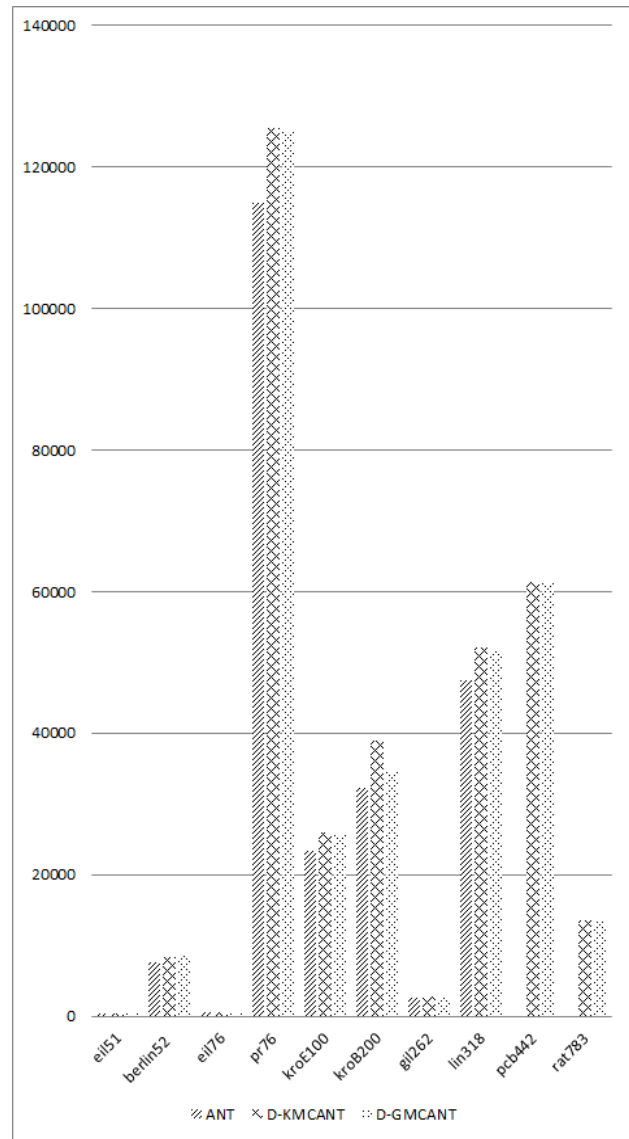
When the ant colony optimization is tested, the average minimum tour lengths are lower than those of the genetic algorithm and clustering genetic algorithms. In Table IV, the results show that ant colony optimization yields the best solutions. However, the computational times of clustering ant colony optimization methods are lower for all datasets. The differences in the average minimum tour lengths are compared by the charts in Figure XVII. Since the ant colony optimization yields the good results on the average minimum tour lengths, the results may be difficult to improve by the clustering ant colony, while the clustering techniques can improve the efficiency of genetic algorithm as showed in Figure XVII (a).

TABLE IV: THE RESULTS OF CLUSTERING ANT COLONY OPTIMIZATION

No.	Dataset	Average Minimum Tour Length			Average Computational Time (Sec.)		
		Ant Colony	Clustering		Ant Colony	Clustering	
			K-means Ant Colony	Gaussian Ant Colony		K-means Ant Colony	Gaussian Ant Colony
1	eil51	<b>442</b>	484	484	4.90	<b>3.17</b>	3.7
2	berlin52	<b>7,608</b>	8,424	8,515	5.25	<b>3.57</b>	3.95
3	eil76	<b>559</b>	624	623	22.35	<b>3.36</b>	3.97
4	pr76	<b>115,057</b>	125,464	125,128	20.11	<b>3.61</b>	4.13
5	kroE100	<b>23,406</b>	25,970	25,838	57.72	<b>3.72</b>	4.31
6	kroB200	<b>32,276</b>	38,960	34,538	1,044.34	<b>5.13</b>	5.84
7	gil262	<b>2,647</b>	2,752	2,747	3,113.64	<b>7.38</b>	7.89
8	lin318	<b>47,514</b>	52,093	51,641	6,992.79	<b>11.00</b>	11.64
9	pcb442	-	<b>61,418</b>	61,439	>10,000.00	<b>17.63</b>	18.53
10	rat783	-	<b>13,539</b>	13,546	>10,000.00	46.97	<b>46.91</b>



(a) Genetic Algorithm and Clustering GA



(b) Ant Colony Optimization and Clustering Ant

FIGURE XVII: THE AVERAGE MINIMUM TOUR LENGTH.

For the average computational time, both K-means and Gaussian mixer models are able to reduce the executable times of the ant colony optimization. Hence, the average minimum tour length may be improved if the numbers of ants or the maximum times are increased. Chart in Figure XVIII illustrates that the average computational times of the simple ant colony optimization and the simple genetic algorithm are higher than those of the other methods. For ant colony optimization and genetic algorithm, they cannot be compared to each other due to the difference in their algorithms and parameters. The proposed methods, clustering evolutionary computation, can improve the average computational time of the simple algorithms. When K-means clustering or Gaussian mixer model were applied, the average computational time can be reduced, especially in the large TSP.

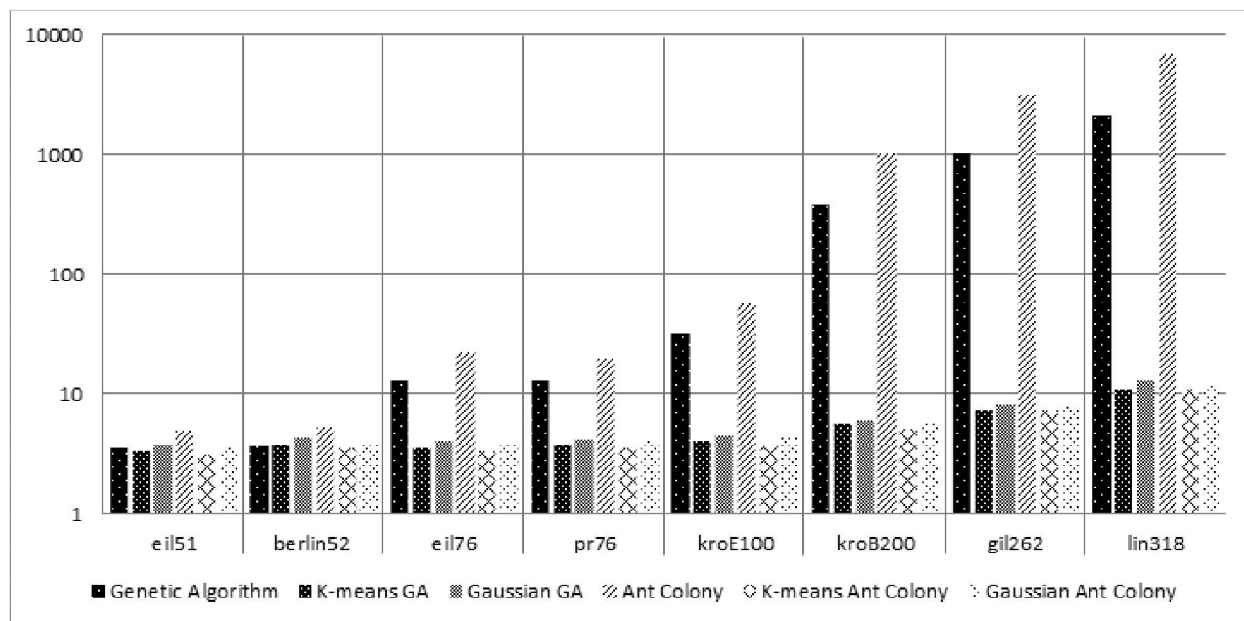


FIGURE XVIII: THE AVERAGE COMPUTATIONAL TIME.

## V. CONCLUSION

Clustering evolutionary computation is proposed to improve the performance of evolution computation on TSP. The proposed methods have three main steps, i.e., node clustering, optimal path finding, and cluster connecting. In the first step, nodes in the TSPs are divided into the smaller problems using K-means clustering or Gaussian mixer model. Then, the genetic algorithm and the ant colony optimization are used to find the optimal path. In the final step, a simple method for choosing clusters and nodes is presented to connect all clusters in the TSP. Path of all clusters are connected by considering the centroids of clusters and the marginal nodes.

The experimental results show that the clustering techniques, both K-means clustering and Gaussian mixer model, yields the better solutions when the optimal paths are found by the genetic algorithm. The computational times of the proposed methods are shorter than the basic evolutionary computation algorithm. This method is suitable for the large scale of TSP. The proposed methods can be applied to the other optimization techniques such as simulated annealing, particle swarm intelligence, or estimation of distribution algorithm (EDA). Moreover, this method is not specific for the TSPs; it can be applied to the other optimization or combinatorial problems.

## REFERENCES

- [1] Al-Dulaimi, B.F., & Ali, H.A. (2008). Enhanced traveling salesman problem solving by genetic algorithm technique (TSPGA). *Proceedings of World Academy of Science: Engineering & Technology*, 40, 296-302.
- [2] Leung, K.S., Jin H.D., & Xu, Z.B. (2004). An expanding self-organizing neural network for the traveling salesman problem. *Neurocomputing*, 62, 267-292.

- [3] Masutti, T.A.S., & de Castro, L.N. (2009). A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem. *Information Sciences*, 179(10), 1454–1468.
- [4] Lo, C.C., & Hsu, C.C. (1998). Annealing framework with learning memory. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 28(5), 648-661.
- [5] Jayalakshmi, G.A., & Sathiamoorthy, S. (2001). A hybrid genetic algorithm – A new approach to solve traveling salesman problem. *International Journal of Computational Engineering Science*, 2(2), 339–355.
- [6] Tsai, H.K., Yang, J.M., Tsai, Y.F., & Kao, C.Y. (2003). Heterogeneous election genetic algorithms for traveling salesman problems. *Engineering Optimization*, 35(3), 297–311.
- [7] Dorigo, M., & Gambardella, L.M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1), 53-66.
- [8] Liu, J.L. (2005). Rank-based ant colony optimization applied to dynamic traveling salesman problems. *Engineering Optimization*, 37(8), 831–847.
- [9] Puris, A., Bello, R., & Herrera, F. (2010). Analysis of the efficacy of a two-stage methodology for ant colony optimization: Case of study with TSP and QAP. *Expert Systems with Applications*, 37(7), 5443–5453.
- [10] Yang, J., Wu, C., Lee, H.P., & Liang, Y. (2008). Solving traveling salesman problems using generalized chromosome genetic algorithm. *Progress in Natural Science*, 18(7), 887-892.
- [11] Liu, F., & Zeng, G. (2009). Study of genetic algorithm with reinforcement learning to solve the TSP. *Expert Systems with Applications*, 36(3), 6995–7001.
- [12] Albayrak M., & Allahverdi, N. (2011). Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms. *Expert Systems with Applications*, 38(3), 1313-1320.
- [13] Laporte, G., & Palekar U. (2002). Some applications of the clustered travelling salesman problem. *The Journal of the Operational Research Society*, 53(9), 972–976.
- [14] Anily, S., Bramel, J., & Hertz, A. (1999). A 5/3-approximation algorithm for the clustered traveling salesman tour and path problems. *Operations Research Letters*, 24(1), 29-35.
- [15] Guttmann-Beck, N., Hassin, R., Khuller, S., & Raghavachari B. (1998). Approximation algorithms with bounded performance guarantees for the clustered traveling salesman problem. *Lecture Notes in Computer Science*, 1530, 6-17.
- [16] Ding, C., Cheng Y., & He, M. (2007). Two-level genetic algorithm for clustered traveling salesman problem with application in large-scale TSPs. *Tsinghua Science & Technology*, 12(4), 459-465.
- [17] Tan, P., Steinbach, M., & Kumar, V. (2006). *Introduction to data mining*. Boston: Pearson Addison Wesley.
- [18] Bramer, M. (2007). *Principles of data mining*. London: Springer-Verlag.
- [19] Dempster, A.P., Laird, N.M., & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1), 1-38.

- [20] Das, S., Abraham, A., & Konar, A. (2009). *Metaheuristic Clustering*. Berlin: Springer.
- [21] Goldberg, D.E. (1989). *Genetic algorithms in search, optimization, and machine learning*. New York: Addison Wesley.
- [22] Jadaan, O.A., Rajamani, L. & Rao, C.R. (2005). Improved selection operator for GA. *Journal of Theoretical and Applied Information Technology (JATIT)*, 2005-2008, 269-277.
- [23] IRIDIA University Libre de Bruxelles. *Publications of ACO algorithms* [Online] 06 March 2012. Available from: URL: <http://www.aco-metaheuristic.org/publications.html> [Accessed 15 December 2012].
- [24] TSPLIB. University of Heidelberg. Available from: <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/> [Accessed: October 2012].