Clustering Process to Solve Euclidean TSP

Abdulah Fajar
*Informatics Department, Faculty of Engineering
Universitas Widyatama Bandung Indonesia
Faculty of Information
and Communication Technology
Universiti Teknikal Malaysia Melaka
#*abd.fajar@gmail.com,

Nur Azman Abu#, Nanna Suryana Herman#,
Shahrin Shahib#
Faculty of Information
and Communication Technology
Universiti Teknikal Malaysia Melaka
#{ nura, nsuryana,shahrinsahib}@utem.edu.my

***Abstract* — Human is able to cluster and filter object efficiently. Clustering problem has been approached from diverse domains of knowledge like graph theory, statistics, artificial neural network and so on. There has been growing interest in studying combinatorial optimization problems by clustering approach, with a special emphasis on the Euclidean Traveling Salesman Problem. Classical ETSP appears as a fundamental problem in various problem such as transportation, manufacturing and logistics application. This study will focus on tour construction. Most of methods focus on tour improvement and using nearest neighborhood for tour construction. This paper will use clustering process to decompose ETSP into smaller sub problem. Clustering process hierarchically arrange adjacency and vertices to form clusters. A threshold of edge weight is applied to split one clusters to several sub clusters. Using this approach the running time can be cut into half compared to TSPLib standard time. The main objective is to develop best clustering process to ETSP and produce a near optimal solution within 10% of best known solution in TSPLib.**

Keywords: Hierarchical Clustering, Euclidean TSP, Tour Construction, Adjacency

I. Introduction

Human has a natural ability to cluster multiple objects quickly. This capacity assist human in making routine decision. However making computers act like human beings in solving the problem is very difficult and demanding the attention of computer scientists and engineers around the world until now [1]

Clustering problem has been approached from diverse domains of knowledge like graph theory, statistics, artificial neural network and so on. There has been growing interest in studying combinatorial optimization problems by clustering strategy, with a special emphasis on the Euclidean Traveling Salesman Problem (ETSP) [1-4]. Classical ETSP appears as a fundamental problem in various problem such as transportation, manufacturing and logistics application, this problem has caught much attention of mathematicians and computer scientists.

The rest of this paper will describe about motivation of this study regarding Euclidean TSP in section II, objective this study will be described in section III. Related work will be described in section IV and section V describes hierarchical clustering approach. Section VI and VII describes hierarchical clustering process and clustering process to Euclidean TSP. The impact of clustering process will discuss in section VIII. Final section (i.e. section IX) will describe conclusion.

II. Motivation

Euclidean Traveling salesman problem (ETSP) is a combinatorial global optimization task of finding the shortest tour of n vertices given the weight of edges. In a more formal way the goal is to find the least weight Hamiltonian cycle in a complete graph G. The TSP problem is a '*NP-hard*' problem. An *NP-hard* problem is extremely unlikely to have a polynomial algorithm to solve it optimally although it is not proved that the worst case exponential solution running times are unavoidable. Good approximation algorithms can produce solutions that are only a few percent longer than an optimal solution and the time of solving the problem is a low-order polynomial function of the number of vertices. Some approximating algorithms produce tours whose lengths are close to that of the shortest tour, but the time complexity is substantially higher than linear.

III. Objective of Study

The primary objective of this study is to produce a better hierarchical clustering strategy that fit into Euclidean TSP solution method. The current research in hierarchical clustering to solve ETSP mostly producing their best algorithm with the running time near $O(n^2)$. For practical industry purposes, an algorithm with running time $O(n^3)$ is still acceptable/viable and average final ETSP solution is maximum 10% above the best known solution . The other supporting objectives are to develop efficient algorithm which produce non edge crossing in planar graph; clusters generation for large size cluster; produce Hamilton path within and inter cluster ; inter cluster connection as initial tour to form global tour.

IV. Related Work

Haxhimusa *et. al* [1] formulated a pyramid algorithm for E-TSP motivated by the failure to identify an existing algorithm that could provide a good fit to the subjects data. More recently, hierarchical (pyramid) algorithms have been used to model mental mechanisms involved in other types of visual problems. The main aspects of the models are (multi resolution) pyramid architecture and a coarse to fine process

978-1-4244-5539-3/10/$26.00 ©2010 IEEE

of successive tour approximations. This algorithm was motivated by the properties of the human visual system has main idea to use agglomerative (bottom up) processes to reduce the size of the input by clustering the nodes into small group and top-down refinement to find an approximate solution, as same as Graham do in [5]. The size of the input (number of vertices in the graph) is reduced so that an optimal solution can be found by the combinatorial search. A pyramid is used to reduce the size of the input in the bottom-up processes.

In bottom-up clustering, vertices are close neighbors are put into the same cluster using greedy approach. These clustered vertices are considered as a single vertex at the reduced resolution. Graph pyramid strategy use MST using Bouruvka algorithm. MST is used as natural lower bound for TSP solution. In the case TSP with triangle inequality which in the case for the Euclidean TSP, MST can be used as to prove the upper bound.

Graph pyramid solution gives good approximation compared to TSPLIB. Solution error for this solution tends to linger around 12%. Running time performance is quadratic for this solution especially whenever number of vertices is more than 600.

V. Hierarchical Clustering Approach

A clustering algorithm is expected to discover the natural grouping that exists in a set of pattern. In hierarchical clustering, the data are not partitioned into a particular cluster in a single step. Hierarchical clustering may be represented by a two dimensional diagram known as dendrogram, as shown in Figure 1.

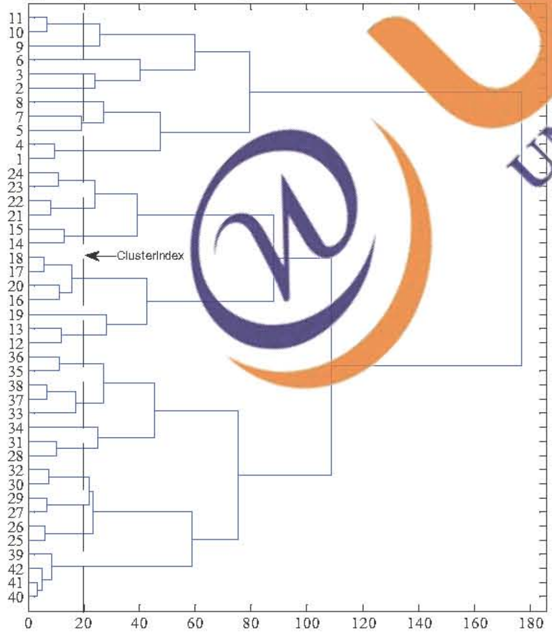

Figure 1. Dendgram of Complete Linkage Method for Dantzig42.tsp

Figure 1 shows dendogram which illustrates the fusions or divisions made at each successive stage of analysis [6].

Among the most popular hierarchical clustering algorithms, BIRCH [7] can typically find a good clustering in a single scan of the data and improve the quality further in a few additional scans. It is also the first clustering algorithm that handles noise effectively. CURE [8] represents each cluster by a certain number of points selected from a well-scattered sample and then shrinking them toward the cluster centroid by a specified fraction. It uses a combination of random sampling and partition clustering to handle large databases. ROCK [9] is a robust clustering algorithm for Boolean and categorical data. It introduces two new concepts: a point's neighbors and links, and to measure the similarity/proximity between a pair of data points.

Clustering approach divide TSP problem to sub graph or sub clusters and reduce problem size. There are many clustering approach available but it should be choose a suitable one. Hierarchical is the simplest algorithm that can be chosen as clustering approach.

An agglomerative hierarchical clustering procedure produces a series of partitions of the data, C_n, C_{n-1}, ... , C_1. The first C_n consists of n single object 'clusters', the last C_1, consists of single group containing all n cases. At each particular stage the method joins together the two clusters which are closest together (most similar). Differences between methods arise because of the different ways of defining distance (or similarity) between clusters.

Number of clusters that is generated in hierarchical clustering is automatically. In this work, it is defined a threshold R [2], where

$$R = \sqrt{\frac{\text{Area of Cluster } A}{\text{The number of Vertices } n}}$$

This threshold will act as guidance to split one cluster to several and result basic clusters. Basic cluster have small size number of nodes. This threshold also prevent clusters overlapped one to others.

Let $G=(V,E,W)$ be a graph, $G_r=(V_r,E_r,W_r)$ and $G_s=(V_s,E_s,W_s)$ be sub graph of graph G, set n_r and n_s are number of vertices V_r and V_s . Here is several technique of linkage between the clusters used in this study:

A. Nearest linkage

The distance between two clusters is the minimum distance between two clusters. This method can cause "chaining" clusters. Cluster's shape potentially overlapped to another clusters.

$$d(G_r, G_s) = \min\{W(V_r, V_s)\}$$

Figure 2. Single Linkage Distance

B. Complete linkage

This method is opposite of single linkage and tends to produce compact clusters. Outliers are given more weight with this method. It is generally a good choice if the clusters are far apart in feature space, but not good if the data are noisy.

$$d(G_r, G_s) = \max\{W(V_r, V_s)\}$$

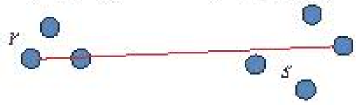

Figure 3. Complete Linkage Distance

C. Average linkage

The distance between two clusters is the mean distance between all possible pairs of nodes in the two clusters.

$$d(G_r, G_s) = \frac{1}{n_r n_s}\sum_{i=1}^{n_r}\sum_{j=1}^{n_s} dist(V_{r_i}, V_{s_j})$$

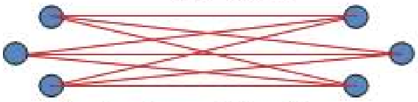

Figure 4. Average Linkage Distance

D. Median Linkage

Used only for Euclidean distance. The distance between two clusters is the Euclidean distance between their *weighted* centroids.

$$d(G_r, G_s) = \|\tilde{V}_r - \tilde{V}_s\|_2$$

Where $\tilde{V}_r$ is the weighted centroid of r. If r was created from clusters p and q, then $\tilde{V}_r$ is defined recursively as:

$$\tilde{V}_r = \frac{1}{2}(\tilde{V}_p + \tilde{V}_q)$$

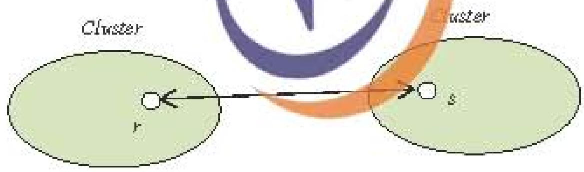

Figure 5. Median Linkage Distance

E. Centroid linkage

Used only for Euclidean distance. The distance between two clusters is the Euclidean distance between their centroids, as calculated by arithmetic mean

$$d(G_r, G_s) = \|\tilde{V}_r - \tilde{V}_s\|_2$$

Where $\tilde{x}$ is the centroid of r, by arithmetic mean:

$$\tilde{V}_r = \frac{1}{n_r}\sum_{i=1}^{n_r} V_{r_i}$$

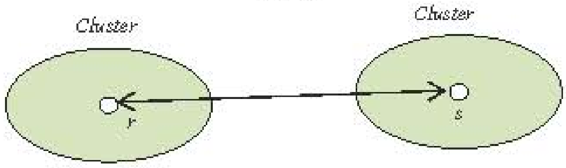

Figure 6. Centroid Linkage Distance

F. Ward's linkage (Sum Square Method)

The sum square (Ward's) method is appropriate for the clustering of points in Euclidean space. Ward's method says that the distance between two clusters, r and s, is how much the sum of squares will increase when merge them.

$$d(G_r, G_s) = \frac{n_r + n_s}{n_r n_s}\|\tilde{V}_r - \tilde{V}_s\|^2$$

Where $\tilde{V}_r$ center of cluster r and n_r is is number of points in cluster r.

VI. Hierarchical Clustering Process

In this study, clustering approach decompose problem into smaller by split a cluster into several sub cluster. This algorithm will use Johnson's algorithm and split one cluster to another using R threshold.

Let $G(V,E,W)$ graph for dantzig42.tsp sample. The clustering process start with n clusters $C_1, C_2, \ldots, C_n$ where each clusters consists of one vertex only. The second step is to find the closest $d(C_i,C_{i+1}) < R$ (most similar using linkage distance and less than R) pair of clusters and merge them into a single cluster, so that now number of clusters becomes one less than the amount previously. Then, next steps, is to compute distances (similarities) between the new cluster and each of the old clusters. Final step is to repeat previous step until distance between clusters not exceed than R value. The result using complete linkage for dantzig42.tsp is shown in Figure 1.

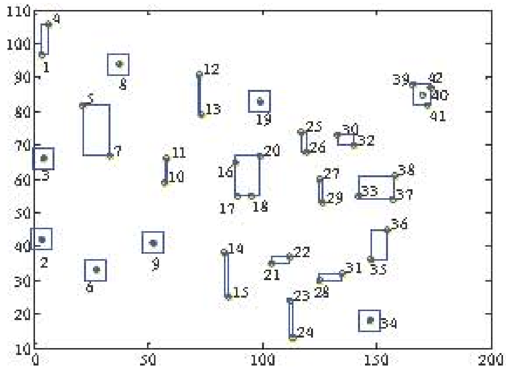

Figure 7. Clusters of dantzig42.tsp

VII. CLUSTERING APPROACH TO EUCLIDEAN TSP

The first step coming from clustering process to generate small compact clusters as shown in Figure 7. Each cluster shall be connected to adjacent clusters. In each clusters, a middle point shall be computed. Based on global hamilton cycle shall be formed as global cluster tour. Possibility of inter cluster connection is shown in Figure 8. Each cluster shall be formed Hamilton path based on two end points of adjacent clusters. The best Hamilton path of each cluster shall be integrated based on global cluster tour to result global tour. Global tour shall be improved and refined to achieve shortest final tour as shown in Figure 9. Tour improvement tour can be applied by opt-2 exchange move.

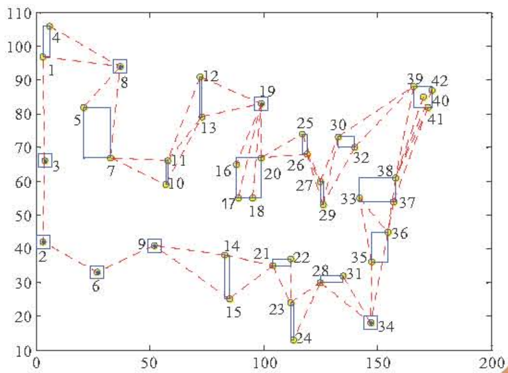

Figure 8. Interconnection between Cluster

VIII. THE CLUSTERING PROCESS IMPACT

The selection method of linkage in hierarchical clustering is very important. In previous study, nearest linkage is used to clusters the problem, and at current progress another linkage is used such as complete, centroid, median, ward and average. Cluster index is used with a role in determining the average number of points in the cluster. Small cluster produces a better path than the cluster with larger size. The difficulty encountered is the link between the paths of each cluster to a problem with large size.

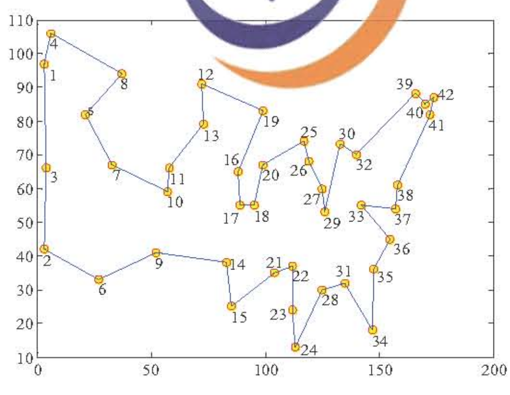

Figure 9. Inter Cluster and Final tour of Dantzig42.tsp

TABLE I. NUMBER OF NODE PER CLUSTER OF RANDOM SAMPLES (N=16 TO N=1024)

Method	Average	Max	Min	Stddev
Nearest	6	8	3	0.96
Complete	2	2	2	0.08
Average	2	3	2	0.12
Median	3	3	2	0.15
Centroid	3	3	2	0.15
Ward	2	2	2	0.08

Random examples with n = 16 – 1024 [2] gave results between 2 to 3 points per cluster unless nearest linkage. While examples of problems TSP Lib give similar results with random sample size is still in the basic clusters of 2 to 4 nodes per cluster. Nearest linkage gives a different result is 6 nodes per cluster. Table 1 shows the results of clustering (nodes per cluster) with a uniform pattern for all the linkage method (except nearest linkage).

TABLE II. TOUR DISTANCE OF RANDOM SAMPLES (N=100)

Method	Average	Max	Min	Stddev
Nearest	832.43	887.85	774.02	29.25
Complete	807.26	862.41	745.37	28.44
Average	[illegible]	873.14	740.33	28.34
Median	809.08	878.30	756.60	29.42
Centroid	806.81	876.30	740.33	29.68
Ward	808.06	869.82	745.37	30.32

Statistically as shown in table II, tour length comparison between the previous methods (nearest linkage) with the other methods yield about 3% shorter. Table 1 and 2 showed that there was a significant correlation that the number of nodes per cluster can affect the length of the tour solution.

TABLE III. RUNNING TIME OF RANDOM SAMPLES (N=100)

Method	Average	Max	Min	Stddev
Nearest	3.61	18.85	0.83	3.60
Complete	0.77	0.88	0.67	0.05
Average	0.77	0.91	0.62	0.08
Median	0.81	1.08	0.60	0.11
Centroid	0.82	1.07	0.63	0.11
Ward	0.79	0.88	0.71	0.05

Table III show that the running time is consumed by a method that is currently used is very short compared with the previous method (nearest linkage method). Consumption level of the running time of current methods is between 21% and 23% only compared to the consumption of the running time of previous methods. In other words there are efficiency and suitability of the method in terms of running time

This study also compares the method to TSP Lib best known solution and standard time. Simulation of this study use Intel Quad Core 2.6 GHz with 2 GB Memory for simulation and run Concorde software from TSP Lib. This simulation is written using Matlab 12 as development tool software.

TABLE IV. TOUR DISTANCE GAP COMPARED TO TSP LIB BEST KNOWN SOLUTION FOR N=29 TO N=1000 NODES

Method	Average	Max	Min	Stddev
Nearest	7.08%	14.71%	-0.01%	3.30%
Complete	6.66%	14.34%	-0.01%	3.11%
Average	6.33%	16.84%	0.98%	3.51%
Median	6.68%	18.64%	0.98%	3.49%
Centroid	6.43%	16.32%	0.98%	3.65%
Ward	6.48%	14.57%	-0.01%	3.20%

All methods also are shown that tour length is not have enough difference one to another (see table IV) but is contradicted to table V. Complete and Ward linkage method give best improvement compared to other method. Both methods consume only 59% and 56% running time compared to TSP Lib Standard time. The maximum running time that is consumed by two methods is about 8 times compared to TSP Lib Standard Time on berlin52.tsp. In that sample (berlin52.tsp) standard time only 0.05 second while Complete and Ward Linkage Method consume 0.40 and 0.42 second. Other methods consume running time longer than TSP standard time. The fastest running time is shown by complete linkage method almost 1% compared to Concorde running time on rat575.tsp and the gap about 7%. Overall result for tour distance less than 7% on average and this result answer the objectives.

TABLE V. RUNNING TIME COMSUMING COMPARED TO CONCORDE FROM TSP LIB FOR N=29 TO N=1000 NODES

Method	Average	Max	Min	Stddev
Nearest	1378.21%	29449.30%	5.27%	4499.28%
Complete	58.47%	797.07%	0.91%	135.04%
Average	116.82%	982.11%	1.03%	238.47%
Median	152.12%	2692.16%	1.21%	410.24%
Centroid	167.00%	3073.47%	1.15%	459.87%
Ward	56.07%	839.34%	1.09%	139.10%

IX. CONCLUSION

Clustering technique should be able to produce near optimal solution. Clustering technique that used should consider number cluster limitation. Smaller cluster size tends to result in more optimal to Hamilton path for each cluster.

The number of nodes of each cluster influence in running time. The six methods presented in this study provide results that are divided into 2 different results. Complete and Ward linkage methods produce smaller running time with 6% longer tour distance compared to TSP Lib. A conclusion obtained from this study is the role of the clustering process can accelerate the running time by selecting the appropriate linkage method.

The next challenge is to minimize the final average result by finding the appropriate method for connecting between the clusters. The method used to connect between the clusters in this study is the cheapest insertion method. This insertion method has a worst case result up to 2 times than the best results, so other methods need to be the worst case result can be smaller.

ACKNOWLEDGMENT

This research is funded by Science and Innovation Fund from Ministry of Sciences and Innovation Technology Malaysia.

REFERENCES

[1] Y. Haxhimusa, W. G. Kropatsch, Z. Pizlo, and A. Ion, "Approximative graph pyramid solution of the E-TSP," *Image and Vision Computing*, 2008.

[2] N. A. Abu, S. Sahib, and N. Suryana, "A Novel Natural Approach to Euclidean TSP," in *The 3th International Conference on Mathematics and Statistics (ICoMS-3)*, Bogor, Indonesia, 2008.

[3] J. N. MacGregor, T. C. Ormerod, and E. P. Chronicle, "A model of human performance on the traveling salesperson problem," *Memory and Cognition*, vol. 28, pp. 1183-1190, 20-01-2000 2000.

[4] D. Vickers, M. Butavicius, M. Lee, and A. Medvedev, "Human performance on visually presented traveling salesman problems," *Psychological Research*, vol. 65, pp. 34-65, 2001.

[5] S. M. Graham, A. Joshi, and Z. Pizlo, "The traveling salesman problem: A hierarchical model," *Journal of Memory & Cognition*, vol. 27, pp. 1191-1204, 2000.

[6] S. Das, A. Abraham, and A. Konar, "An Overview Metaheuristics Clustering," in *Metaheuristic Clustering*. vol. 178, P. J. Kacprzyk, Ed. Berlin: Springer, 2009, pp. 1-62.

[7] Zhang T, Ramakrishnan R, and L. M., "BIRCH: an efficient data clustering method for very large databases," in *ACM SIGMOD International Conference on Management of Data*, Montreal, Canada, 1996.

[8] S. Guha, R. Rastogi, and K. Shim, "CURE: An Efficient Clustering Algorithm for Large Database," in *ACM SIGMOD International Conference on Management of Data*, Montreal, Canada, 1998.

[9] S. Guha, "ROCK: A Robust Clustering Algorithm for Categorical Attributes," *Journal of Information System*, vol. 25, pp. 245-366, 2000.