Fuzzy Clustering in Grouping Traditional Market Distribution and Genetic Algorithm Application in Routing of Packed Cooking Oil Distribution

Teja Primawati Utami, Syamsul Ma’arif, and Yandra Arkeman
Department of Agroindustrial Technology
Bogor Agricultural University
E-mail: tejaprimawati@yahoo.com, syamsul4958@gmail.com, yandra_ipb@yahoo.com

Keywords: Fuzzy clustering, transportation salesperson problem, genetic algorithm, packaging cooking oil

Abstract
This paper presents the modelling of intelligent routing of transportation of packaging cooking oil from the distribution center to traditional market in the cluster in Indonesia, especially in Jakarta. Indonesia is the nation who has many islands. Every island has different population of people. Everyday many people go to traditional market to buy main consumption products like palm oil etc. Price of palm oil is very sensitive to increase when it lack at the market, so sustainability of present palm oil at the market is very important. Focus of this research is to demonstrate how to optimize routing distribution from distribution centre to markets in the cluster. Optimum route hopefully can guarantee the availability of product and stock in the market to maintaining the price. The clustering is created by fuzzy clustering and the routing is created by Transportation Salesperson Problem (TSP) with Genetic Algorithm (GA) method. Genetic algorithm is a method for solving optimization problem that based on evolutionary theory in biology.

Introduction
The movement of finished product to customers is market distribution. In market distribution, the end customer represents the final destination (Bowersox, 2002). The ability to deliver goods as customer ordered is service. It will be call logistic which are integrated production and distribution. The logistics components of a corporation consist of: (1) a number of manufacturing plants, (2) zero, one, or more distribution echelons with distribution centres, (3) the customers, (4) the suppliers of components and raw materials, (5) recycling centres for used products and returned packaging containers, and finally (6) the transportation channels that link all of the above components (Goetschalckx, et al., 2002). Indonesia is the nation who has many islands. Every island has different population of people. Every region has many traditional markets to serve daily people consumption. These traditional markets need distribution centre to ensure availability of consumption product in these traditional markets. How to optimum determine distribution centre of each region can define by clustering. Hamzah (2001) show that clustering process by fuzzy (fuzzy clustering) gives the result better that defined it by firm directly approaches. In this paper we focus on Jakarta region.

Jakarta is capital city of Indonesia, which have 5 regions are North Jakarta, South Jakarta, West Jakarta, East Jakarta and Central Jakarta (Figure 1). Traditional market in Jakarta is coordinated by PD Pasar Jaya. PD Pasar Jaya have 153 traditional market (figure 2). Every traditional market uncontrollable availability of goods and disparity of goods prices. So this paper will purpose to make distribution centre to solve this problem.
Distribution centre will be created by fuzzy clustering. Partition clustering essentially deals with the task of partitioning a set of entities into a number of homogeneous clusters, with respect to a suitable similarity measure. Due to the fuzzy nature of many practical problems, a number of fuzzy clustering methods have been developed following the general fuzzy set theory strategies outlined by Zadeh (1965). The main difference between the traditional hard clustering and fuzzy clustering can be stated as follows: in hard clustering an entity belongs only to one cluster, while in fuzzy clustering entities are allowed to belong to many clusters with different degrees of membership.

Clustering has been around for many decades and located itself in a unique position as a fundamental conceptual and algorithmic landmark of data analysis. Almost since the very inception of fuzzy sets, the role and potential of these information granules in revealing and describing structure in data was fully acknowledged and appreciated (Mika Sato, 2006).

In the recent years clustering has undergone a substantial metamorphosis. From being an exclusively data driven pursuit, it has transformed itself into a vehicle whose data centricity has been substantially augmented by the incorporation of domain knowledge thus giving rise to the next generation of knowledge-oriented and collaborative clustering. Related to these, the fuzzy clustering is used.

After distribution centre are defined there is a need of Travelling Salesperson Problem (TSP) to distribute the product to the markets. In the TSP, the goal is to find the shortest distance between $N$ traveling points. The number of possible route for an $N$ city tour requires $N!$ additions. An exhaustive search through all possible paths is acceptable only when $N$ is small. As $N$ increases, the number of possible path grows geometrically. A 20-city tour involves $2.43 \times 10^{18}$ additions. Even with 1 billion additions performed in 1 second, this would take over 1852 years. Adding one more city would cause the number of additions to increase by a factor of 21. Obviously, exhaustive search becomes impractical.

So, to make it more quickly and simply, genetic algorithm is necessary to used for saving the time. Genetic Algorithm (GA) is a method for solving optimization problem that based on evolutionary theory in biology. This algorithm work with a population of candidate solutions named as chromosome that initially generated randomly from the area of the solution space of objective function. By using a mechanism of genetic operator i.e. crossover and mutation the population is evolutes controlled by fitness function that directed to convergence condition (Widyastuti and Hamzah, 2007).

This paper presents the application of GA approach in this cluster market of routing transportation problem called TSP. Although GA probably will not lead to the best solution, it can find a near optimal solution in a much less time (within several minutes).

**LITERATURE REVIEW**

**Fuzzy C-Means (FCM)**

The fuzzy c-means (FCM) algorithm (Bezdek, 1981) is one of the most widely used methods in fuzzy clustering. Data clustering is the process of dividing data elements into classes or clusters so that items in the same class are as similar as possible, and items in different classes are as dissimilar as possible. Depending on the nature of the data and the purpose for which clustering is being used, different measures of similarity may be used to place items into classes, where the similarity measure controls how the clusters are formed. Some examples of measures that can be used as in clustering include distance, connectivity, and intensity.
There are two clustering namely hard and soft. In hard clustering, data is divided into distinct clusters, where each data element belongs to exactly one cluster. In fuzzy clustering (also referred to as soft clustering), data elements can belong to more than one cluster, and associated with each element is a set of membership levels. These indicate the strength of the association between that data element and a particular cluster. Fuzzy clustering is a process of assigning these membership levels, and then using them to assign data elements to one or more clusters.

The FCM algorithm attempts to partition a finite collection of \( n \) elements \( X = \{x_1, \ldots, x_n\} \) into a collection of \( c \) fuzzy clusters with respect to some given criterion. Given a finite set of data, the algorithm returns a list of \( c \) cluster centre \( C = \{c_1, \ldots, c_c\} \) and a partition matrix \( U = u_{i,j} \in [0, 1], \ i = 1, \ldots, n, \ j = 1, \ldots, c \), where each element \( u_{ij} \) tells the degree to which element \( x_i \) belongs to cluster \( c_j \). Like the \( k \)-means algorithm, the FCM aims to minimize an objective function. The standard function is:

\[
    u_{k}(x) = \frac{1}{\sum_{j} \left( \frac{d(x, c_{j})}{d(x, c_{k})} \right)^{2/(m-1)}}.
\]

which differs from the \( k \)-means objective function by the addition of the membership values \( u_{ij} \) and the fuzzifier \( m \). The fuzzifier \( m \) determines the level of cluster fuzziness. A large \( m \) results in smaller memberships \( u_{ij} \) and hence, fuzzier clusters. In the limit \( m = 1 \), the memberships \( u_{ij} \) converge to 0 or 1, which implies a crisp partitioning. In the absence of experimentation or domain knowledge, \( m \) is commonly set to 2. The basic FCM Algorithm, given \( n \) data points \( (x_1, \ldots, x_n) \) to be clustered, a number of \( c \) clusters with \((c_1, \ldots, c_c)\) the center of the clusters, and \( m \) the level of cluster fuzziness.

In fuzzy clustering, each point has a degree of belonging to clusters, as in fuzzy logic, rather than belonging completely to just one cluster. Thus, points on the edge of a cluster, may be in the cluster to a lesser degree than points in the center of cluster. An overview and comparison of different fuzzy clustering algorithms is available.

Any point \( x \) has a set of coefficients giving the degree of being in the \( k \)th cluster \( w_k(x) \). With fuzzy \( c \)-means, the centroid of a cluster is the mean of all points, weighted by their degree of belonging to the cluster:

\[
    c_k = \frac{\sum_{x} w_k(x) x}{\sum_{x} w_k(x)}.
\]

The degree of belonging, \( w_k(x) \), is related inversely to the distance from \( x \) to the cluster center as calculated on the previous pass. It also depends on a parameter \( m \) that controls how much weight is given to the closest center. The fuzzy \( c \)-means algorithm is very similar to the \( k \)-means algorithm.

Choose a number of clusters. Assign randomly to each point coefficients for being in the clusters. Repeat until the algorithm has converged (that is, the coefficients’ change between two iterations is no more than \( \epsilon \), the given sensitivity threshold). Compute the centroid for each cluster, using the formula above. For each point, compute its coefficients of being in the clusters, using the formula above. The algorithm minimizes intra-cluster variance as well, but has the same problems as \( k \)-means; the minimum is a local minimum, and the results depend on the initial choice of weights.

The expectation-maximization algorithm is a more statistically formalized method which includes some of these ideas: partial membership in classes. Fuzzy \( c \)-means has been a very important tool for image processing in clustering objects in an image. In the 70's,
mathematicians introduced the spatial term into the FCM algorithm to improve the accuracy of clustering under noise.

**Cluster Analysis**

Cluster analysis or clustering is the task of assigning a set of objects into groups (called clusters) so that the objects in the same cluster are more similar (in some sense or another) to each other than to those in other clusters.

Clustering is a main task of explorative data mining, and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with low distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It will often be necessary to modify preprocessing and parameters until the result achieves the desired properties.

Besides the term clustering, there are a number of terms with similar meanings, including automatic classification, numerical taxonomy, botryology (from Greek βότρυς "grape") and typological analysis. The subtle differences are often in the usage of the results: while in data mining, the resulting groups are the matter of interest, in automatic classification primarily their discriminative power is of interest. This often leads to misunderstandings between researchers coming from the fields of data mining and machine learning, since they use the same terms and often the same algorithms, but have different goals.

**Center of Cluster**

In centroid-based clustering, clusters are represented by a central vector, which may not necessarily be a member of the data set. When the number of clusters is fixed to k, k-means clustering gives a formal definition as an optimization problem: find the k cluster centers and assign the objects to the nearest cluster center, such that the squared distances from the cluster are minimized.

The optimization problem itself is known to be NP-hard, and thus the common approach is to search only for approximate solutions. A particularly well known approximative method is Lloyd’s algorithm, often actually referred to as "k-means algorithm". It does however only find a local optimum, and is commonly run multiple times with different random initializations. Variations of k-means often include such optimizations as choosing the best of multiple runs, but also restricting the centroids to members of the data set (k-medoids), choosing medians (k-medians clustering), choosing the initial centers less randomly (K-means++) or allowing a fuzzy cluster assignment (Fuzzy c-means).

Most k-means-type algorithms require the number of clusters k to be specified in advance, which is considered to be one of the biggest drawbacks of these algorithms. Furthermore, the algorithms prefer clusters of approximately similar size, as they will
always assign an object to the nearest centroid. This often leads to incorrectly cut borders in between of clusters (which is not surprising, as the algorithm optimized cluster centers, not cluster borders).

K-means has a number of interesting theoretical properties. On one hand, it partitions the data space into a structure known as Voronoi diagram. On the other hand, it is conceptually close to nearest neighbor classification and as such popular in machine learning. Third, it can be seen as a variation of model based classification, and Lloyd's algorithm as a variation of the Expectation-maximization algorithm for this model discussed below.

**Traveling Salesperson Problem**

The idea of the travelling salesman problem (TSP) is to find a tour of a given number of cities, visiting each city exactly once and returning to the starting city where the length of this tour is minimized. The first instance of the travelling salesman problem was from Euler in 1759 whose problem was to move a knight to every position on a chess board exactly once (Michalewicz, 1994).

The Travelling Salesperson Problem (TSP) is one of the issues combinatorial optimization, if there are a number of cities (or place) and the cost of travel from one city to other cities. Description of the problem is how to find the cheaper route of visit all the cities, each the city is only visited once, and must back to the original departure city. The combination of all existing route is the factorial number of cities. Travel cost can be a distance, time, fuel, convenience, and so forth.

**Genetic Algorithm**

Genetic algorithms are search techniques and optimization which is inspired by the principles of genetics and natural selection (Darwin's theory of evolution). This algorithm is used to obtain the exact solution for the optimization problem of a single variable or multi variable.

GA is a general purpose guided random search that based on the natural selection principles of biological evolution to improve the potential solutions. GA includes random elements which help to prevent the search begin trapped in local minimum. These properties overcome some of the short comings of conventional optimization approaches in ill-structured problems (Can and Rad, 2002).

Being inherently parallel, GA is performed over a population of solution candidates. The manipulation process uses genetic operators to produce a new population of individuals (offspring) by manipulation the solution candidates. The algorithms start working by evaluating thousands of scenarios automatically until they find an optimal answer. The genetic algorithms bias the selection of chromosomes so that those with the better fitness functions tend to reproduce more often than those with worse evaluations.

Given an optimisation problem, GA first encodes the parameters into solution candidates. In the initial phase, the population consists of randomly enervated heterogeneous solution candidates. After all chromosomes go through evaluation process, an initial population will improve as parents are replaced by better and better children. The best individual in the final population can be a highly evolved solution to the problem.

According to Briant and Arthur (2000), the genetic algorithm process generally consists of the following steps i.e.: encoding, evaluation, crossover, mutation, and decoding.
MATERIALS AND METHODS

Fuzzy clustering is one method which can capture the uncertainty situation of real data and it is well known that fuzzy clustering can obtain a robust result as compared with conventional hard clustering (Sato, 2006). Following the emphasis on the general problem of data analysis, which is a solution able to analyze a huge amount of complex data, the merit of fuzzy clustering is then presented.

After cluster was constructed, next step is to design routing from centre of cluster to the members. The members and the cluster are traditional market in Jakarta, Indonesia. Routing is designed by Transportation Salesperson Problem and Genetic Algorithm is used to make optimization. The methodology framework is shown on Figure 3.

RESULTS AND DISCUSSION

There is a great interest in clustering techniques due to the vast amount of data generated in every field including business, health sciences, engineering and aerospace. It is essential to extract useful information from the data. Clustering techniques are widely used in pattern recognition and related applications. This research monograph presents the clusters for traditional market in Jakarta, which these have each distribution centre.

Identify Parameter for Grouping

Clustering of traditional market in Jakarta is constructed by 4 parameters combining. These are latitude position, longitude position, density of traders at the markets and accessibility of 153 transitional markets.

Clustering to Define Centre of Distribution

Centre of distribution of traditional market in Jakarta is defined by fuzzy clustering. We use MATLAB to create the clustering. Fuzzy clustering with c-means is used for data analysis. The algorithm of fuzzy c-means (FCM) are below:

1. Input data to be in the cluster is a matrix of n x m (n = number of data sample, m attribute for each data). X_{ij} = sample data to i (i = 1,2,...,n), attribute to-j (j=1,2,...,m).
   Number of cluster (c) = 15
   Square (w) = 2
   Maximum iteration (maxIter) = 100
   Error (\(\delta\)) = 10^{-5}
   First objective function (P_0) = 0
   First iteration (t) = 1

2. Random number (\(\mu_{ik}\)) generated, \(i = 1,2,...,n; k = 1,2,...,c\); with sequence below.
   \[Q_f = \sum_k \mu_{ik}J\]
   \(j = 1,2,...,m\)
   which are,
   \[\mu_{ik} = \frac{Q_{ik}}{Q_f}\]

3. Center of cluster to-k ; with k = 1,2,...,c; and j = 1,2,...,m
   \[V_{kj} = \frac{\sum_{i=1}^{n}(\mu_{ij})}{\sum_{i=1}^{n}(\mu_{ij})}\]
4. Objective function at iteration to-t, \( P_t \):
\[
P_t = \sum_{i=1}^{n} \sum_{k=1}^{c} \left( \frac{\sum_{j=1}^{n} (x_{ij} - V_{kj})^2}{\sum_{j=1}^{n} (x_{ij} - V_{kj})^2} \right)^{\frac{1}{\mu}}
\]

5. Partition matrix change

\[
\mu_{ik} = \frac{\left( \sum_{j=1}^{n} (x_{ij} - V_{kj})^2 \right)^{-\frac{1}{\mu-1}}}{\sum_{k=1}^{c} \left( \sum_{j=1}^{n} (x_{ij} - V_{kj})^2 \right)^{-\frac{1}{\mu-1}}}
\]
i = 1,2,...,n ; and k = 1,2,...,c

6. Finish iteration

If: \( (|P_t - P_{t-1}| \leq E) \) or \( (t > M) \) so iteration is stoping.
If not \( t = t+1 \), looping go to 3.

Clustering of numerical data forms the basis of many classification and system modelling algorithms. The purpose of clustering is to identify natural groupings of data from a large data set to produce a concise representation of a system's behaviour.

Fuzzy Logic Toolbox tools allow to find clusters in input-output training data. It can use the cluster information to generate a Sugeno-type fuzzy inference system that best models the data behaviour using a minimum number of rules. The rules partition themselves according to the fuzzy qualities associated with each of the data clusters. The command-line function is using, \texttt{genfis2} to automatically accomplish this type of FIS generation.

Quasi-random two-dimensional data is used to illustrate how FCM clustering works.

To load the data set and plot it, type the following commands:

\[
\text{load sheet1.dat}
\]
\[
\text{plot(sheet1(:,1), sheet1(:,2),'o')}
\]

Next, invoke the command-line function \texttt{fcm} to find two clusters in this data set until the objective function is no longer decreasing much at all.

\[
[\text{center,U,objFcn}] = \text{fcm(sheet1)}
\]

Here, the variable centre contains the coordinates of the fifteen cluster centres, \( U \) contains the membership grades for each of the data points, and \( \text{objFcn} \) contains a history of the objective function across the iterations.

The \texttt{fcm} function is an iteration loop built on top of the following routines:

- \texttt{initfcm} — initializes the problem
- \texttt{distfcm} — performs Euclidean distance calculation
- \texttt{stepfcm} — performs one iteration of clustering

To view the progress of the clustering, plot the objective function by typing the following commands:

\[
\text{figure}
\]
\[
\text{plot(objFcn)}
\]
\[
\text{title('Objective Function Values')}
\]
\[
\text{xlabel('Iteration Count')}
\]
\[
\text{ylabel('Objective Function Value')}
\]

Figure of convergency is presented by fig. 4.

Finally, plot the fifteen cluster centres found by the \texttt{fcm} function using the following code:
\[
\text{maxU} = \text{max}(U);
\]
\[
\text{index1} = \text{find}(U(1, :) == \text{maxU});
\]
\[
\text{index2} = \text{find}(U(2, :) == \text{maxU});
\]
\begin{verbatim}
figure
    line(fcmdata(index1, 1), fcmdata(index1, 2), 'linestyle',
         'none', 'marker', 'o', 'color', 'g');
    line(fcmdata(index2, 1), fcmdata(index2, 2), 'linestyle',
         'none', 'marker', 'x', 'color', 'r');
    hold on
    plot(center(1, 1), center(1, 2), 'ko', 'markersize', 15, 'LineWidth', 2)
    plot(center(2, 1), center(2, 2), 'kx', 'markersize', 15, 'LineWidth', 2)
\end{verbatim}

Coordinate geographic centre of each cluster is presented by table 2.

**Mapping Distribution Centre by Fuzzy Clustering**

The 15 cluster traditional market in Jakarta was defined. The centres of cluster are presented with different colour. One of them will be presented by fig. 5.

**Routing with Transportation Salesperson Problem-Genetic Algorithms.**

The TSP is a standard problem in optimization. The objective in this paper is to minimize the travelling distance of \( N \) cities in a 10 km square radius from (0,0). Figure 3 for cluster 1 which colour is yellow shows a 8-city tour starting from green dot (Kramat Jati) colour which is the centre to 7-others cities in the 10 km square radius, where the yellow dots are indicates the city needed to be travelled.

Fig. 7 describes the flow of the optimization of TSP by GA. GA first encodes the travelling cities into chromosome. The population size is 1. After the chromosome goes through evaluation process, a fitness value is assigned to the chromosome. The child is then compared with the parent. If it is fitter than the parent, it will replace the parent, or it will not be used. Then the parent will reproduce a child through neighbourhood mutation (which will be discussed in part v in this section). The process repeats until it reaches the maximum number of generations. The chromosome in the final population is a highly evolved solution to the problem.

**Coding**

In GA, the parameters to be optimized are encoded into chromosomes (Figure 8) and each chromosome is a solution candidate. The encoding scheme depends on the nature of parameters to be optimized. In this problem, each city going to be visited is represented by an integer. The chromosome \( S \) is a sequence of integers, can be formed by encoding the list of cities in the order they are visited. The length of chromosome equals to \( N \).

**Initialization**

In this problem, we set the population size equals to 1 and the initial population is randomly generated.

**Evaluation**

In the evaluation module, each chromosome is coded with the integer of the cities to be travelled and the travelling time is calculated. The fitness value, calculated according to the fitness function, which is defined by the designer, is assigned to the chromosome.
Reproduction and Generation Selection
The reproduction module selects the alleles to be mutated. Then a new child chromosome is produced. The new chromosome is compared with parent chromosome. Elitism is used in the generation selection. If the new chromosome fitter than the parent, then it replaces the parent, else it will not be used. This avoids the loss of potential candidates by copying the best member into the succeeding generation.

Neighbourhood Mutations
Conventional crossover and mutation are the most commonly used operations in GA to obtain offspring. However, simple crossover and mutation may lead to violation of the constraint of TSP, as the city to be travelled may be missed or duplicated. As shown in Figure 5, the crossover operation will not work. Let’s say, we have a 2nd crossover point. Every number in parent 1 before the crossover point is copied into the same position in child 1. Then, every number after the crossover point in parent 2 is put into child 1. The opposite is done for child 2.

After the crossover operation, in Child 1, the city 1 is visited twice and city 8 is missed. The reproduction should preserve all the cities required in the chromosomes from the parents to the children. A different approach has therefore been adopted to the reproduction of chromosomes. Ho and Yeung (2000), A neighbourhood is defined for the best chromosome in a generation and the chromosome only evolves to one of its neighbours.

The choice of chromosomes for the initial generation plays a vital role in the convergence toward the optimal solution. In order to smooth out this effect, 20 tests have been carried out for each traffic condition with each neighbourhood definition. In each test, the chromosomes of the initial generation are selected randomly from the set of possible sequences. The average of the minimum crossover the 20 tests is then calculated. All the simulation runs are performed on MATHLAB. Figure 7 summarizes the average travelling distance (of 20 tests) of 8 cities over 100 generations for different neighbourhood mutations.

The following pseudo-code that is created for solve the above problems with the TSP using genetic algorithms:

function Fitness (Kromosom[i])→integer
    {calculate the fitness value of each chromosome}

Declaration
    Jum : integer
    j : integer
    Chromosome[][] : array of integer of integer
Distance function (input A, B : integer) → integer
    {generate the distance between two cities A and B }

Algorithm
    Jum ← Jarak(A,Kromosom[i][1])
    for j ← 2 to 4 do
        Jum → Jum + Distance (chromosome[i][j-1], chromosome[i][j])
    endfor
    Jum ← sum + Jarak(Kromosom[i][4],A)
        → Jum
Crossover procedure (input populasi: integer, pc:real)
    {parent selection on the cross over}
**Declaration**

k : integer
R[] : array of integer
function random (input a-b : integer) → integer
\( \text{generates random numbers from a number to b} \)

**Algorithm**

\( k = 0 \)
While \( k \leq \text{populasi} \) do
\( R[k] \leftarrow \text{random}(0-1) \)
if \( R[k] < \rho_c \) then
pilih Kromosom[k][]sebagai induk
endif
\( k \leftarrow k+1 \)
endwhile

function of Number mutations (input JumGen, JumlahKromosom: integer, \( \rho_m \): real) → integer
\( \text{/count the number of themutations} \)

**Declaration**

TotalGen : integer
JumMutasi : integer

**Algorithm**

\( \text{TotalGen} \leftarrow \text{JumGen} \ast \text{JumlahKromosom} \)
\( \rho_m \leftarrow 0.2 \)
\( \text{JumMutasi} \leftarrow 0.2 \ast \text{TotalGen} \)
\( \rightarrow \text{JumMutasi} \)

**CONCLUSION**

Existing traditional market is very important to help Jakarta’s people life. They go to traditional market everyday to buy many things for basic need consumption. Availability of goods and stability of price are important to consider. This paper gives a solution by presenting the distribution centres to facilitate all 153 traditional markets in Jakarta. It should be distribute to 15 clusters. Each cluster has one centre, it could be distribution centre. MATLAB is used to calculate and solve the problem by fuzzy clustering. The iteration to convergence were 27 iteration. Every distribution centre is nearly optimum to distribute the goods to all traditional markets in the cluster.

Genetic algorithms appear to find good solutions for the travelling salesman problem, however it depends very much on the way the problem is encoded and which crossover and mutation methods are used. It seems that the methods that use heuristic information or encode the edges of the tour (such as the matrix representation and crossover) perform the best and give good indications for future work in this area.

Overall, it seems that genetic algorithms have proved suitable for solving the travelling salesperson problem. It seems that the biggest problem with the genetic algorithms devised for the travelling salesperson problem is difficulty to maintain structure from the parent chromosomes and still end up with a legal tour in the child chromosomes.
Perhaps a better crossover or mutation routine that retains structure from the parent chromosomes would give a better solution than we have already found for some travelling salesman problems.

**LITERATURE CITED**


### Tables

Tabel 1. history of the objective function across the iterations

<table>
<thead>
<tr>
<th>ITERATION</th>
<th>FCN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration count = 1</td>
<td>obj. fcn = 0.041209</td>
</tr>
<tr>
<td>Iteration count = 2</td>
<td>obj. fcn = 0.030768</td>
</tr>
<tr>
<td>Iteration count = 3</td>
<td>obj. fcn = 0.030263</td>
</tr>
<tr>
<td>Iteration count = 4</td>
<td>obj. fcn = 0.029399</td>
</tr>
<tr>
<td>Iteration count = 5</td>
<td>obj. fcn = 0.027972</td>
</tr>
<tr>
<td>Iteration count = 6</td>
<td>obj. fcn = 0.026048</td>
</tr>
<tr>
<td>Iteration count = 7</td>
<td>obj. fcn = 0.024263</td>
</tr>
<tr>
<td>Iteration count = 8</td>
<td>obj. fcn = 0.022998</td>
</tr>
<tr>
<td>Iteration count = 9</td>
<td>obj. fcn = 0.021964</td>
</tr>
<tr>
<td>Iteration count = 10</td>
<td>obj. fcn = 0.021105</td>
</tr>
<tr>
<td>Iteration count = 11</td>
<td>obj. fcn = 0.020507</td>
</tr>
<tr>
<td>Iteration count = 12</td>
<td>obj. fcn = 0.020139</td>
</tr>
<tr>
<td>Iteration count = 13</td>
<td>obj. fcn = 0.019853</td>
</tr>
<tr>
<td>Iteration count = 14</td>
<td>obj. fcn = 0.019580</td>
</tr>
<tr>
<td>Iteration count = 15</td>
<td>obj. fcn = 0.019307</td>
</tr>
<tr>
<td>Iteration count = 16</td>
<td>obj. fcn = 0.019042</td>
</tr>
<tr>
<td>Iteration count = 17</td>
<td>obj. fcn = 0.019042</td>
</tr>
<tr>
<td>Iteration count = 18</td>
<td>obj. fcn = 0.018506</td>
</tr>
<tr>
<td>Iteration count = 19</td>
<td>obj. fcn = 0.019042</td>
</tr>
<tr>
<td>Iteration count = 20</td>
<td>obj. fcn = 0.018366</td>
</tr>
<tr>
<td>Iteration count = 21</td>
<td>obj. fcn = 0.018337</td>
</tr>
<tr>
<td>Iteration count = 22</td>
<td>obj. fcn = 0.018313</td>
</tr>
<tr>
<td>Iteration count = 23</td>
<td>obj. fcn = 0.018292</td>
</tr>
<tr>
<td>Iteration count = 24</td>
<td>obj. fcn = 0.018275</td>
</tr>
<tr>
<td>Iteration count = 25</td>
<td>obj. fcn = 0.018262</td>
</tr>
<tr>
<td>Iteration count = 26</td>
<td>obj. fcn = 0.018251</td>
</tr>
<tr>
<td>Iteration count = 27</td>
<td>obj. fcn = 0.018242</td>
</tr>
</tbody>
</table>

Table 2. Coordinate geographic centre of each cluster

<table>
<thead>
<tr>
<th>No.</th>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.166.857</td>
<td>1.065.258</td>
</tr>
<tr>
<td>2</td>
<td>6.117.264</td>
<td>1.065.397</td>
</tr>
<tr>
<td>3</td>
<td>6.097.971</td>
<td>1.065.185</td>
</tr>
<tr>
<td>4</td>
<td>6.113.407</td>
<td>1.064.485</td>
</tr>
<tr>
<td>5</td>
<td>6.132.045</td>
<td>1.065.233</td>
</tr>
<tr>
<td>6</td>
<td>608.422</td>
<td>1.064.826</td>
</tr>
<tr>
<td>7</td>
<td>6.198.298</td>
<td>1.065.389</td>
</tr>
<tr>
<td>8</td>
<td>6.177.212</td>
<td>1.065.032</td>
</tr>
<tr>
<td>9</td>
<td>6.112.826</td>
<td>106.504</td>
</tr>
<tr>
<td>10</td>
<td>614.756</td>
<td>1.064.774</td>
</tr>
<tr>
<td>11</td>
<td>6.090.052</td>
<td>1.064.985</td>
</tr>
<tr>
<td>12</td>
<td>6.111.806</td>
<td>1.064.832</td>
</tr>
<tr>
<td>13</td>
<td>6.075.793</td>
<td>1.065.701</td>
</tr>
<tr>
<td>14</td>
<td>6.151.379</td>
<td>1.065.046</td>
</tr>
<tr>
<td>15</td>
<td>6.085.015</td>
<td>1.064.342</td>
</tr>
</tbody>
</table>

680
Figures

Fig. 1. Region of Jakarta Map

Fig. 2. Traditional markets in Jakarta
Fig. 3. The Methodology Framework

Fig. 4: Convergency Iteration
Fig. 5. Cluster 1, 2, and 3.

Fig. 6. Cluster 3.
Fig. 7. Cluster one as a sample for routing TSP

Fig. 8. Mechanisms of the proposed algorithms
Fig. 9. Crossover of chromosome

Fig. 10. The average travelling distance (of 20 tests) of 8 cities over 100 generations.

Fig. 11. Sample optimum routing of 8 cities from distribution centre (1) to seven other cities.