ABSTRACT
The Traveling Salesman Problem (TSP) is a well-known nondeterministic problem aiming to find the shortest route that visits each city once and finally returns back to the starting city. Ant Colony Optimization (ACO) technique gives a good solution to TSP, However it takes a lot of computational time.

In this paper, a novel algorithm as proposed to solve TSP: Adaptive Affinity Propagation (AAP) was used to optimize the performance of Ant Colony Optimization. The basic idea of the new proposed approach is to group cities into many clusters using AAP and then find the optimal path for each cluster separately using ACO. Thus, the computational time decreases. Experimental results show that the proposed algorithm has preferable performance compared to ACO in term of computational time and optimal path length.

General Terms
Data Clustering, Data Mining.

Keywords
Travelling Salesman Problem, Ant Colony Optimization, Adaptive Affinity Propagation.

1. INTRODUCTION
The Travelling Salesman Problem can simply be stated as: if a travelling salesman wishes to visit exactly once each of a list of cities and then return to the home city, what is the least costly route the travelling salesman can take? The TSP has many practical applications including: regular distribution of goods or resources, finding of the shortest of costumer servicing route, planning bus lines etc. The TSP is a well-known NP problem which means that there is no exact algorithm to solve it in polynomial time. Consequently, the development of approximation algorithms to solve TSP has received considerable attention through the last years.

Ant colony optimization (ACO) [1, 2] is one of the most attractive approximation algorithms that has been used for this purpose. Dorigo [3] proposed the first ACO at the early of 1990’s. In principal, it was inspired by the foraging behavior of real ant colony when ants try to find the shortest path between their nest and a food source. The ants deposits a pheromone trail on the ground while walking back and forth between their nest and a food source. At any time, an ant tends to follow a path on which are the pheromone trails left by its predecessors. Ultimately, the ants will be able to find the shortest path from the food source to the nest, and the whole colony will follow this path to transport the food back to their nest.

Adaptive Affinity Propagation is new approach for clustering data with large number of instances. It iteratively transmits messages between data samples, until a high-quality set of exemplars appears. Adaptive Affinity Propagation was proposed to overcome the limitations of affinity propagation algorithm. The operation of these two approaches will be reviewed in details later.

The rest of paper is organized as follows: In section II, the basic concepts of the Travel Salesman Problem, Ant Colony Algorithm and Adaptive Affinity Propagation algorithms are reviewed. The proposed approach is described in Section III. Section IV shows the simulation experiments and results of the new proposed algorithm. Section V concludes the paper with some directions for future work.

2. RELATED WORK
Many researches recently were conducted Travel Salesman Problem. In this section, some of algorithms that have been proposed to solve TSP are introduced [4, 5]. In [6], Gaussian mixture model and k-means clustering are used as clustering approaches to enhance the efficiency of evolutionary algorithms in TSP and to reduce the computational time. Then, the genetic and ant colony algorithms are used to find the shortest path of each cluster. As a final step, all clusters paths are connected together to obtain the optimal path of TSP. Chao, et, al[7] developed a two level Genetic algorithm for Clustered Traveling Salesman Problem with Application in Large-Scale TSPs. TLGA depends on finding the shortest Hamiltonian cycle for each cluster using a genetic algorithm (GA) and then connecting all the intra-cluster paths in a certain sequence to form a whole tour using the proposed algorithm.

2.1 Affinity Propagation Algorithm
Affinity Propagation [8] is a simple message-passing clustering algorithm developed by Frey and Dueck (2007). In Affinity Propagation, all data points send messages recursively to all other points to announce their willingness to be an exemplar [9]. These messages are stored in two matrices: the availability matrix which indicates how appropriate that candidate would be as an exemplar, and the responsibility matrix that measures how fit any point to represent other points. The iterations are performed until either the cluster boundaries remain unchanged over a number of iterations, or after some predetermined number of iterations. The exemplars are extracted from the final matrices as those whose 'responsibility + availability' for themselves is positive. The AP algorithm steps are shown below [10, 11].
1. Initialize the availabilities to zero \( a(i,k) = 0 \).
2. Update the responsibilities using the following equation:
   \[
   R(i,k) = S(i,k) - \max\{A(i,j) + S(i,j)\}
   \]
   Where \( S(i,k) \) is the similarity of data point \( i \) and exemplar \( k \).
3. Update the availabilities using following equation:
   \[
   A(i,k) = \min\{0, R(k,k) + \sum \max\{0, R(j,k)\}\}, \text{ where } j \in \{1,2,\ldots,n\} \text{ but } j \neq i \text{ and } j \neq k; \text{ And Update self-availability by following equation:}
   \]
   \[
   A(k,k) = \sum \max\{0, R(j,k)\}.
   \]
4. Compute \( \text{sum} = A(i,k) + R(i,k) \) for data point \( i \) and find the value of \( k \) that maximize the sum to identify the exemplars.
5. If Exemplars do not change for fixed number of iterations go to step (6) else go to Step (1)
6. Assign the data points to Exemplars on the basis of maximum similarity to find clusters.

### 2.2 Adaptive Affinity Propagation

Adaptive Affinity Propagation [12], is a new proposed approach to overcome the limitation of AP. In AP, it is difficult to determine the best value of ‘preference’ parameter to obtain the optimal clustering form. Also, oscillations cannot be eliminated automatically if occur [13]. To deal with these limitation, AP includes the following techniques: adaptive adjusting of the damping factor to eliminate oscillations when they happen, the adaptive escaping from oscillations when adaptive damping method fails by decreasing the preferences value, and adaptive scanning of preferences to search space of the number of clusters to form the optimal clustering structure. The value of "Preferences" (p) plays a key role in determining the clustering performance and the number of formed clusters. Adaptive Affinity Propagation use the following adaptive technique to determine the optimal preferences value (p) for a given number of clusters:

\[
P_{t+1} = \alpha P_t + \beta
\]

Where \( t \) is the running time of Affinity propagation, and \( \alpha > 1 \) mean number of clusters in the \( t \)th running result, \( \beta \) is searching step and \( P_{t+i}, P_t \) are negative values [14].

### 2.3 Ant Colony Optimization

Ant Colony Optimization [1, 3] is a recent optimization heuristic. It was inspired by the way that ants living in a collaborative colonies looking for food sources through different paths. Finding the shortest path to be followed by others depends on pheromone trail which used as indirect communication to construct the best path as shown in Figure 1 [15].

![Fig 1: Ants use pheromone as indirect communication to build best tour.](image)

Each ant is randomly put on a city and choose the following city to be visited using a Probabilistic decision rule as expressed in the following equation:

\[
p_{ij}(t) = \left\{ \begin{array}{ll}
\frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{w \in j_k(t)} [\tau_{iw}(t)]^\alpha [\eta_{iw}]^\beta}, & \text{if } j \in j_k(i) \\
0, & \text{otherwise}
\end{array} \right.
\]

Where \( \tau_{ij} \) is the intensity of trail between edge \((i,j)\) and \( \eta_{ij} = 1/d_{ij} \) is heuristic visibility, \( j_k^{(i)} \) is a set of cities which unvisited yet. \( \alpha \) and \( \beta \) weights to control the pheromone trail and heuristic visibility.

After ant completes its tour, it updates the local pheromone trail with equation below.

\[
\tau_{ij}(t + 1) = (1 - \rho)\tau_{ij} + \Delta \tau_{ij}(t)
\]

where \( \tau_{ij} \) is the amount of pheromone on a given edge \( i,j \), \( \rho \) is the rate of pheromone evaporation, \( \Delta \tau_{ij} \) is the amount of pheromone deposited, typically given by

\[
\Delta \tau_{ij}(t) = \left\{ \begin{array}{ll}
\frac{Q}{L_K}, & \text{if } (i,j) \in \text{tour done by ant } k \\
0, & \text{otherwise}
\end{array} \right.
\]

where \( L_K \) is the cost of the \( K \)th ant’s tour (typically length) [16].
3. THE PROPOSED ALGORITHM

In this paper, a new algorithm is proposed, that uses Adaptive Affinity Propagation clustering to optimize the performance of Ant Colony Algorithm, to solve TSP. Figure 2 shows the full operation of the proposed algorithm. In the first stage, data points are grouped into many clusters using Adaptive Affinity Propagation algorithm, then the optimal path for each cluster is found by applying Ant Colony algorithm for each cluster. Finally, all intra-cluster paths are connected to build a whole tour.

3.1 Data Clustering

The objective of the first stage of the proposed algorithm is to group data points in the TSP dataset into many clusters using Adaptive Affinity Propagation clustering Algorithm. This helps to minimize the computational time of Optimization algorithm as well as to reduce the number of iterations to obtain the optimal solution.

3.2 Finding the Optimal Path for Each Cluster

After grouping data points into clusters, the shortest path in each cluster is calculated. There are many techniques to calculated the shortest path of TSP [17]. However, in large data sets, the computational time will be high. Thus, data clustering stage helps to improve computational time of calculating the optimal path through the whole dataset.

In this paper, ACO was employed to calculate the shortest path in each cluster as ACO yields good results on TSP [18, 19]. Figure 3 shows an example of finding optimal path of each cluster.

3.3 Clusters Connecting

In this step, after computing the shortest path of each cluster separately \(p_1, p_2, \ldots, p_n\), the complete path is generated by aggregating cluster paths together to compute the final tour. Calculating the shortest path for each cluster separately facilitates obtaining the whole optimal solution in a very short time.

4. EXPERIMENTAL EVALUATION

In this section, the performance of the proposed algorithm is examined and evaluated through conducting many simulations. The proposed algorithm has been implemented and simulated using MATLAB Programming Environment. Performance of the new proposed algorithm is evaluated through carrying out many MATLAB simulation experiments. The results of the proposed algorithm are compared to those obtained from simple ACO algorithm and other optimization algorithms.

The real datasets used for testing [20] are shown below in Table 1.

<table>
<thead>
<tr>
<th>No.</th>
<th>Dataset</th>
<th># of Cities</th>
<th>Optimal Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Arabic24</td>
<td>24</td>
<td>229.5</td>
</tr>
<tr>
<td>2</td>
<td>Dj38</td>
<td>38</td>
<td>6656</td>
</tr>
<tr>
<td>3</td>
<td>Berlin52</td>
<td>52</td>
<td>7,542</td>
</tr>
<tr>
<td>4</td>
<td>St70</td>
<td>70</td>
<td>675</td>
</tr>
</tbody>
</table>

4.1 Experimental Results for Arabic24 Dataset

Arabic24 dataset is set of indices of 24 Arabic capital cities; this dataset is symmetric TSPs in the two-dimensional Euclidean distance. The first step of proposed work is to apply simple Ant Colony Optimization to Arabic24 dataset to measure to shortest path and time required to find it. Figure 4 represents the optimal path to visit all cities and return to first city.

Fig 1: Main Steps of the proposed algorithm to Solve TSP

Fig 2: Finding the optimal path for each cluster using ACO.
To compare these results with Ant Colony Optimization based on Adaptive Affinity Propagation, the data set was loaded in Adaptive Affinity Propagation algorithm to group data points into clusters, then measure the shortest path of each cluster using Ant Colony Optimization algorithm, with consideration that shortest path of each cluster mean minimum length of all cities and return to first city. The total optimal route is the summation of all clusters’ shortest path.

With new algorithm, dataset was grouped to two clusters using Adaptive Affinity Propagation, ACO then was applied to these two clusters to find shortest path, Figures 5 and 6 represent the shortest path of each cluster of dj38, The total shortest path= 7111.1223 compared to simple ACO algorithm=6662.3479.

The proposed technique groups the Arabic24 dataset to two clusters, the first cluster get shortest path with 131.6734 and takes few seconds to calculate it, while the second cluster get shortest path is73.7046 and also takes few seconds.
For $dj38$, the proposed algorithm groups the data set to six clusters with one path for each cluster.

Table 2 shows that the clustering Ant Colony Optimization algorithm is better than the simple Ant Colony Optimization algorithm, since it gives minimum tour length and less computational time compared to results obtained using simple ACO. For example, on $dj38$, the minimum tour length of the proposed algorithm is significantly better than that of simple ACO algorithm, and mostly, calculated in not more than half
of its computational time. The differences in the minimum path lengths and time are compared in Figures 9 and 10.

![The minimum path length](Image)

**Fig 9:** The minimum path length for different datasets using ACO and the proposed algorithm.

![The minimum computational time](Image)

**Fig 10:** The computational time for different datasets using ACO and the proposed algorithm.

5. CONCLUSION

The Ant Colony Optimization technique gives a good solution of TSP. However, it takes a lot of computational time. In this proposed approach, the computational time is reduced via using Adaptive Affinity Propagation (AAP) to optimize the performance of simple Ant Colony Optimization for solving TSP.

The proposed algorithm improves the computational time and the length of optimal route over the traditional ACO approaches. This idea might be useful for other problems solved by the ant colony algorithm.

6. REFERENCES


