Improved K-Means Algorithm for Capacitated Clustering Problem

S. GEETHA[1]
G. POONTHALIR[2]
P. T. VANATHI[3]

PSG College of Technology
Tamil Nadu
India
[1]`geet_shan@yahoo.com`
[2]`thalirkathir@rediffmail.com`

Abstract. The Capacitated Clustering Problem (CCP) partitions a set of n items (eg. customer orders) into k disjoint clusters with known capacity. During clustering the items with shortest assigning paths from centroids are grouped together. The summation of grouped items should not exceed the capacity of cluster. All clusters have uniform capacity. The CCP is NP-Complete and Combinatorial optimization problem. Combinatorial optimization problem can be viewed as searching for the best item in a set of discrete items, which can be solved using search algorithm or meta heuristic. However, generic search algorithms have not guaranteed to find an optimal solution. Many heuristic algorithms are formulated to solve CCP. This work involves the usage of the best known clustering algorithm k-means with modification, that use priority as a measure which directs the search for better optimization. The iterative procedure along with priority is used for assigning the items to the clusters. This work is developed using MATLAB 7.0.1 and tested with more than 15 problem instances of capacitated vehicle routing problem (CVRP). The computational results are competitive when compared with the optimal solution provided for the problems.

Keywords: Combinatorial optimization problem, Capacitated Clustering Problem, Centroids, K-means algorithm.

(Received July 17, 2009 / Accepted October 26, 2009)

1 Introduction

In real life, there is a need for moving goods/services from the service providers to various geographically dispersed points of service requester. The final cost of delivery of service depends on transportation and routing cost. The cost of delivery can be considerably reduced by using software management without compromising the services provided to the requesters. The requesters are grouped based on their needs/demands with optimal number of cluster and minimum cost of each service delivery. The provider has a lot of constraints, in delivering their service. This constraint includes capacity of cluster, delivery cost, and number of clusters.

The optimal consolidation of customer's orders into vehicle shipment is an important problem in logistics. This arises in a variety of applications like grouping order into load that fills the vehicle. The vehicle is then assigned to deliver the customer orders to each group from a single service provider.

A provider can be a post office, departmental store, etc. that initiates the service. An example of clustering with single service provider is shown in the figure 1 with three clusters of service requester and vehicles used for shipment.

Clustering is an unsupervised classification of patterns into groups [5]. Clustering is a difficult combinatorial problem. Clustering algorithm can be hierarchical or partitioned. Hierarchical algorithm find successive clusters using previously established clusters, whereas partitioned algorithm determine all cluster at once.

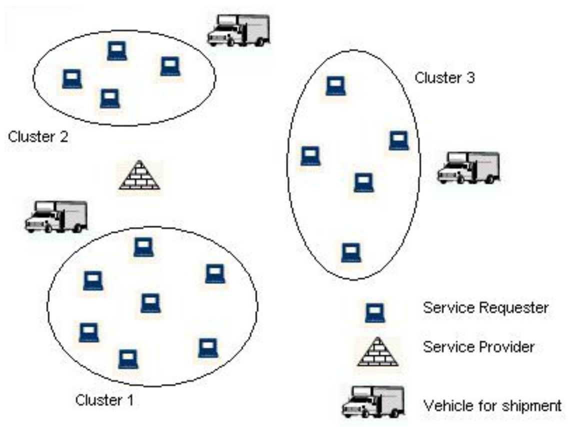

Figure 1: A sample of Clustering

Another important property is whether the clustering uses symmetric or asymmetric distance. An important step in clustering is to select a distance measure that determines the similarity between items. This influences the shape of the cluster. The Euclidean distance is a commonly used distance measure. The service requesters are clustered depending on their demands and Euclidean distance. This forms the basis for determining optimal routing of transportation problem. The service providers are limited with the capacity of goods/ service to be transported. This capacity limit is taken for the formation of clusters. These problems are often formulated as CCP [14]. The CCP is a NP-Complete and Combinatorial Optimization Problem. The exact optimization integer programming algorithm is inefficient for larger problem [4]. The CCP is a special case of facility location problem and closely related to generalized assignment problem [10]. Facility location problem is a branch of operations research, concerning itself with mathematical modeling and to provide solution for the problems like placement of facilities to minimize transportation costs, avoid placing hazardous materials near housing, outperform competitors' facilities, etc. A simple facility location problem is the Fermat-Weber problem, in which a single facility is to be placed, with the only optimization criterion being the minimization of the sum of distances from a given set of point sites. More complex problems considered in this discipline include the placement of multiple facilities, constraints on the locations of facilities, and more complex optimization criteria. When deciding where to place a facility that serves geographically scattered client sites - whether the facility is a delivery center, a distribution center, a transportation hub, a fleet dispatch location, etc - a typical objective is to minimize the sum of the distances from the facility's location to the client sites.

Definition: The CCP is defined as grouping 'n' items into k clusters to minimize the route cost/distance with the specified capacity constraint.

This problem is closely related with Capacitated Centered Clustering Problem (CCCP) [11], Capacitated p-median problem (CPMP) [2]. The CCP also provides solution to vehicle routing problem (VRP) [1][13], bin packing [6], waste collection [7] etc. which involves clustering.

2 Literature Survey

Several assignment and heuristic algorithm have been developed in past decade to solve CCP. Mulvey and Beck [9] proposed heuristics and used randomly generated seeds as initial solution for solving the CCP. Koskosidis and Powell [8] extended the work of [9] by proposing an iterative algorithm that was shown more effective than other heuristic algorithms. It avoids the specification of seed customers required by other algorithms. They developed special algorithm for seed selection rather than random generation of seeds. The iterative heuristics use self-correcting scheme in three phases which they describes as greedy assignment, seed relocation and local exchange. Thangiah and Gubbi [15] used genetic algorithm to find good cluster of customers for VRP with a 'cluster first and route second'. Hybrid Simulated Annealing and tabu search to solve CCP was proposed by Osman and Christofieds [12]. They developed a simple constructive heuristic with R-interchange generation mechanism, hybrid Simulated Annealing, and Tabu search. How-Ming Shieh and May [14] used Genetic Algorithm for solving CCP. Binary coded strings are used to represent the chromosomes, which eliminates the occurrence of infeasible solution. The chromosome is divided into two parts, one represents the customers and other represents the seeds of the clusters. An adaptive penalty function is used to handle the capacity constraint that enhances the convergence and solution quality. França et al.[3] developed a new adaptive tabu search approach to solve the CCP. They used two neighborhood generation mechanisms of the local search heuristic: pairwise interchange and insertion. Zalik has introducted a k-means algorithm that performs correct clustering without pre-assigning the exact number of clusters [16]. In this paper, the clusters are formed using modified k-means algorithm for solving CCP which is simple and competitive when compared with other methods.

3 Problem Definition

The CCP is considered to have n requesters, whose demands are known and are distributed in (x,y) coordinates. The n requesters are grouped to form k clusters. Each cluster has $n_1, n_2, ..., n_k$ number of requesters with the condition that

$$\sum_{j=1}^{k} n_j = n$$

where n is the total number of requesters The problem is given with a set of

Requesters : $r_1, r_2, r_3, ..., r_n$
Coordinates : (x_1, y_1), (x_2, y_2), (x_3, y_3), ... ,(x_n, y_n)
Demands : $d_1, d_2, d_3, ..., d_n$
Capacity : C
where $r_i \in R$ are the set of requesters who are distributed in the Euclidean plane (x_i, y_i), the demand (d_i) and capacity (C) of cluster are positive integers. A Property of Euclidean plane is that the distances are symmetric.

Let X be the binary matrix, such that

$$x_{ij} = \begin{cases} 1 & \text{i}f\ requester\ i\ is\ assigned\ to\ cluster\ j, \\ 0 & \text{o}therwise \end{cases}$$

Consider an example of 10 customers need to be allocated to 3 vehicles based on the capacity constraint that the customer's demand should be less than or equal to the vehicle capacity.

Let c1,c2...c10 be the customers distributed in the (x,y) plane. The binary matrix is 10 X 3, where rows represents the customers and column represents the clusters. Matrix x_{ij} is represented as 0 01, 010, 100, 001, 100, 010, 001, 100, 010, 100.

The objective is to find X which minimizes

$$\sum_{j=1}^{k} \sum_{i=1}^{n} cost_{ij} * x_{ij} \qquad (1)$$

Subject to

$$\sum_{j=1}^{k} x_{ij} = 1, i = 1, 2, ..., n \qquad (2)$$

$$\sum_{i=1}^{n} d_i x_{ij} <= C, j = 1, 2, ..., k \qquad (3)$$

Where $cost_{ij}$ represents the cost of the closeness of requester i to the cluster j (i.e. it can be the cost or time of travel between i and j). The objective function (1) strives to minimize the total assignment cost of requester to the cluster. The constraint (2) ensures that each requester i is assigned to only one cluster j. The constraint (3) is to restrict that the total demand of the requester in the cluster should not exceed the cluster capacity C.

4 Proposed Work

In this work, the CCP is solved using improved k-means algorithm which includes capacity as one of the constraints for clustering the items along with the Euclidean distance for checking the closeness of the items within cluster.

4.1 K-means Clustering Algorithm

The k-means algorithm assigns each point to the cluster whose center (are also called as centroid) is nearest. The center is the average of all the points in the cluster i.e. the co-ordinates are the arithmetic mean of each dimension separately over all points in the cluster.

$Algorithm$: The steps in k-means is given as follows

1. Choose the number of clusters, k.
2. Randomly generate k clusters and determine the cluster centers, or directly generate k random points as cluster centers.
3. Assign each point to the nearest cluster center.
4. Recompute the new cluster centers.
5. Repeat the two previous step until some convergence criterion is met or the assignment has not changed.

4.2 Improved k-means clustering algorithm

The improved k-means algorithm includes a priority measure to select the requesters for a cluster. The requesters are assigned to the nearest cluster based on maximum demand and minimum distance so the requester having larger demand are assigned to the cluster first and the requester with smaller demand can be easily packed in to other clusters. If requesters are assigned based on distance alone, the number of clusters formed may not be optimal since requesters with smaller demand may be assigned to the cluster before the requester with larger demand which may lead to the formation of additional cluster. The Euclidean distance $(cost_{ij})$ measure calculation (4) is used in calculating the distance between the requester and centroid as (4).

$$cost_{ij} = \sqrt{(x_i - x_j)^2 (y_i - y_j)^2}, \quad (4)$$

where $i = 1, 2, ..., n$
$j = 1, 2, ..., k$
x and y refers the coordinate in the Euclidean plane and i refers the requester and j refers the cluster. The $cost_{ij}$ is calculated for all i to every j.

5 Algorithm of Proposed Work

The major steps involved in the formation of the algorithm are described in the following section.

Calculate the number of clusters

It is calculated based on the demand (d_i) of the requester and capacity of cluster (C) as

$$k = \left\lceil \sum_{i=1}^{n} d_i / c \right\rceil \quad (5)$$

Select initial centroids

The initial k centroids are selected by arranging the requester based on their demand in their non-increasing order $d_1 > d_2 > d_3 > d_n$. Let it be the list D. Then the first k requester becomes k centroids.

Assign the requester

The Euclidean distances between each requester to all the k centroids are calculated. Group all the requester r_i to the closest centroid j. To find the appropriate centroid j for ri, calculate a priority value as,

$$Prioirty\ P_i = cost_{ij}/d_i \quad (6)$$

This priority determines the r_i which has the highest priority of having the centroid j.

The selected r_i is assigned based on the constraint (3). If the constraint (3) is not satisfied the selected r_i will be assigned to the next nearest centroid based on (6) and (3)

Centroid Calculation

The centroid (X_j, Y_j) for each cluster is calculated based on the members of the cluster.
Let $(x_1, y_1), (x_2, y_2), ...(x_j, y_j)$ be the co-ordinates of the members of cluster j.

$$x_j = \sum_{m=1}^{j} x_m / n_j$$

$$y_j = \sum_{m=1}^{j} y_m / n_j$$

$$c_j = (x_j, y_j) \quad (7)$$

where c_j represent the jth centroid and n_j represent the number of requester in cluster j

Convergence Criteria

The iterative procedure is repeated until there is no change in cluster formed.

```
**Improved K means algorithm**
**Input :**
    Co-ordinates $(x_i, y_i)$
    Demands $d_i$
    Requester $r_i$
**Output :**
    k clusters
**Procedure :**
Calculate k using (5)
Select first k centroids from list D
Initialize the binary matrix X with zeros
while not converged
    for each requester $r_i \in R$,
        while $r_i$ is not assigned
            Calculate the Euclidean distance measure using (4) to each of the k clusters and arrange it in sorted order
            Assign nearest centroid of $r_i$ as m
            Group all unassigned requesters as G with m as their nearest centroid
            Calculate the priority value for $r_i \in G$ using (6)
            Assign $r_i \in G$ to their nearest centroid based on the priority value without violating the constraint (3).
            Update $x_{ij}$.
            if $r_i$ is not assigned then
                choose the next nearest centroid
            end if
        end while
    end for
    Calculate the new centroid from the formed clusters using (7).
end while
```

6 Computational Results

The three algorithms have been implemented in MATLAB 7.0.1 with Pentium4.0, 2.3GHz. The first algorithm is the general k-means algorithm as given in section 4.1 for CCP. The second algorithm is the modified k-means without priority for CCP and the third algorithm is the improved k-means with priority. The Capacitated Vehicle Routing Problem (CVRP) Bench mark dataset of Augerat,et al set A, Augerat,et al set B, Augerat,et al set P, Christofides and Eilon are used for testing. In CVRP the customers are need to be clustered in order to have minimum traveling cost between them. The customers have demands and vehicles servicing these customers also have a uniform capacity.

The number of clusters depends on the number of vehicles and each cluster is to be serviced by a vehicle. Based on demands and the location of customers on the Euclidean plane, the clusters are formed. The formed clusters efficiency are measured based on the tightness ratio which is calculated as,

$$Tightnessratio = TotalDemand/TotalCapacity$$

The tightness of the constraints is more when the ratio is high. The algorithm is tested with 15 instances of the benchmark data sets. In the benchmark data set each customer has a node number and a demand. Every customer has x and y co-ordinates and are randomly distributed in the Euclidean plane. The Table 1 lists the characteristics of the problem sets. The columns specifies the problem instance, number of customers, cluster capacity, number of clusters. The number of iterations taken by k-means algorithm without priority and with priority and tightness ratio of optimal and obtained value for k-means without priority and with priority are tabulated in table 2. The problem instances of large number of customers, high cluster capacity, customers with high demands are tested and the results are tabulated in Table 1. The data set is available at http://www.branchandcut.org.

The general k-means algorithm is not converging with the optimal number of vehicles, since there is no specification is imposed in selecting the initial centroids also the customers are not considered in sorted order.The convergence is not guaranteed, because the customers with smallest demands are considered while clustering even before the customer with largest demand in some cases.

The k-means algorithm without priority considers the customers with largest demand as their initial centroids. Customers are also considered in the order of their decreasing demands. While assigning the customers it checks for the nearest centroid, if constraint overrides, it considers the next nearest centroid.

The k-means algorithm with priority is similar to k-means algorithm without priority, except that it includes the priority measure.

Now to measure the effectiveness of the proposed work, the mean and standard deviation of the clusters formed are compared with the optimal solution in Table 3 and 4. Table 3 projects the mean and standard deviation of optimal solution and table 4 projects the mean and standard deviation of obtained solution with and without priority. The mean is calculated for each cluster and the overall mean is calculated as mean of all cluster means. The mean of each cluster is calculated based on the demand of customers in that cluster, which shows how effectively the vehicle is packed.

Table 1: Characteristics of Problem Instances

Problem Instance	No. Cust	Clust Capacity	No. clust
A-n33-k5	33	100	5
A-n45-k7	45	100	7
A-n55-k9	55	100	9
A-n60-k9	60	100	9
A-n80-k10	80	100	10
B-n45-k5	45	100	5
B-n63-k10	63	100	10
B-n78-k10	78	100	10
E-n22-k4	22	6000	4
E-n33-k4	33	8000	4
E-n101-k8	101	200	8
E-n101-k14	101	112	14
P-n16-k8	16	35	8
P-n45-k5	45	150	5
P-n101-k4	101	400	4

Table 2: Number of Iterations and tightness ratio

Problem Instance	No. Iterate		Tightness ratio		
	without priority	with priority	opt	without priority	with priority
A-n33-k5	4	3	0.89	0.89	0.89
A-n45-k7	16	7	0.91	0.91	0.91
A-n55-k9	11	11	0.93	0.93	0.93
A-n60-k9	7	6	0.92	0.92	0.92
A-n80-k10	14	10	0.94	0.94	0.94
B-n45-k5	5	6	0.97	0.97	0.97
B-n63-k10	19	3	0.92	0.92	0.92
B-n78-k10	17	14	0.93	0.93	0.93
E-n22-k4	7	6	0.94	0.94	0.94
E-n33-k4	4	4	0.92	0.92	0.92
E-n101-k8	14	10	0.91	0.91	0.9
E-n101-k14	15	10	0.93	0.93	0.93
P-n16-k8	2	2	0.88	0.88	0.88
P-n45-k5	4	6	0.92	0.92	0.92
P-n101-k4	18	6	0.91	0.91	0.91

The mean of cluster means is calculated to find the average demand of each customer. The standard deviation shows the load balancing of all clusters. The lesser the deviation the balancing of load is high, otherwise balancing is low.

7 Discussion

The results shown in Table 1 projects that the number of clusters calculated using (5) has shown to be optimal with respect to tightness ratio. The number of iteration is minimum when compared to k-means algorithm without priority.

The result in Table 3 and 4 shows the amount of deviation of the clusters formed with benchmark data set. The rows 1, 2, 5, 7, 10, 11, 12, 13, 14 and last column

Table 3: Mean and Std. Dev. of optimal solution

Problem Instance	Optimal Solution Mean	Optimal Solution std Dev
A-n33-k5	14.1548	2.2875
A-n45-k7	14.7102	3.1984
A-n55-k9	16.2399	3.8361
A-n60-k9	14.2952	2.6088
A-n80-k10	12.9422	2.9211
B-n45-k5	11.203	1.3297
B-n63-k10	16.279	4.5701
B-n78-k10	12.8532	2.1588
E-n22-k4	1103.3	233.9
E-n33-k4	1151.3	576.28
E-n101-k8	-	-
E-n101-k14	-	-
P-n16-k8	19.646	8.6852
P-n45-k5	15.856	1.3428
P-n101-k4	14.652	0.9987

Table 4: Mean and Std. Dev. of obtained solution

Problem Instance	Improved k-mean without Priority		Improved k-mean with Priority	
	Mean	Std dev	Mean	Std dev
A-n33-k5	14.3003	3.5142	13.9476	1.4147
A-n45-k7	14.9649	3.9483	14.943	2.296
A-n55-k9	16.2449	4.1736	16.1164	3.9957
A-n60-k9	15.5310	6.8897	15.7907	7.1249
A-n80-k10	12.5494	3.255	12.2671	2.3624
B-n45-k5	11.9042	3.5592	12.09	4.1811
B-n63-k10	15.9383	5.1243	15.494	2.9056
B-n78-k10	12.9251	3.4963	12.713	3.6101
E-n22-k4	1231.7	561.11	1202.9	542.07
E-n33-k4	946.181	375.45	1038.3	452.82
E-n101-k8	14.6946	2.2031	14.8619	3.0687
E-n101-k14	15.5070	4.5751	15.2018	3.9287
P-n16-k8	18.9583	7.6209	18.958	7.6209
P-n45-k5	15.8094	1.2433	15.739	0.7807
P-n101-k4	15.1251	4.0417	14.937	3.4507

of table 4 indicate the deviations are less when compared to bench mark clusters. For few instances the results are not challenging, this depends on distribution of customers and their demands. If the customers are concentrated in a specific interval of location rather than evenly distributed, the proposed work has some deviation. Symmetric distribution of customers i.e., evenly distributed customers with varying demands produce challenging results. This variation in results purely depends on the symmetric and asymmetric distribution of customer's location and their demands. For example, the problem instance a-n33-k5 has its customers distributed evenly in the Euclidean plane, the results are proven to be promising. In the case of a-n60-k9 the customers are concentrated on a particular location in Euclidean plane, which has drastic impact on the result. The graphs in this paper shows the scatter of customers in Euclidean plane of x and y location. The sample iteration for the problem instance is projected for p-n45-k5. The initial distributions of the requesters are shown in the figure 2. The five centroids are also identified which is plotted with marker star and depot is shown with polygon.

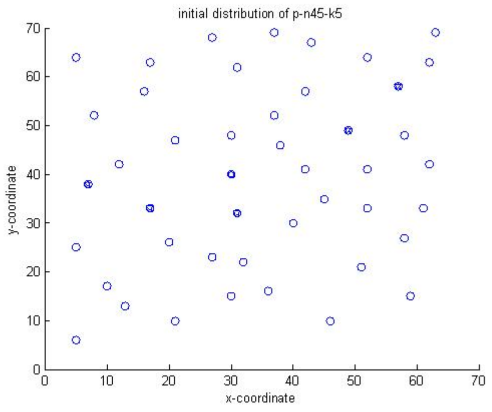

Figure 2: Initial distribution of requesters

Figure 3 shows the initial assignment of requesters to the centroids with different colors of + marker. This shows there is a need for tuning, because there are some requesters who are not grouped to near by centroid.

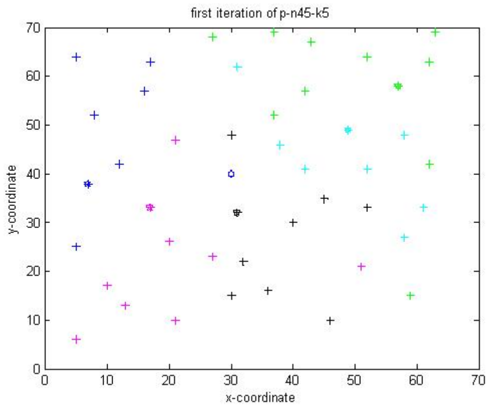

Figure 3: After first iteration

The figure 4 shows 3rd iteration of improved k-means in which only one requester is not clustered properly and all others have been clustered properly.

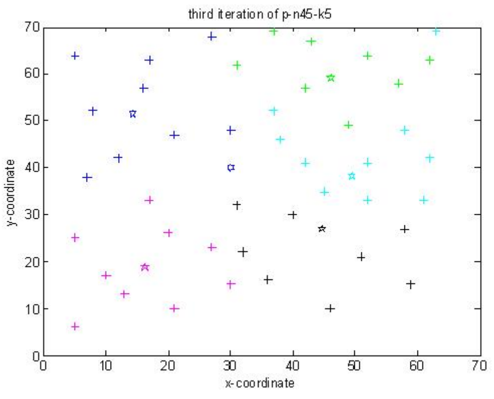

Figure 4: After third iteration

For the next two iteration there is not much change in their groups. Figure 5 shows the last iteration in which all requesters are grouped into proper clusters. The

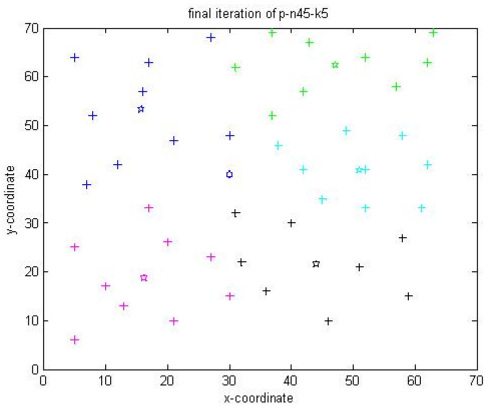

Figure 5: After final iteration

clustering formed by improved k-means algorithm for few problem instances are shown in the figures 6, 7, and 8.

8 Conclusion and Future Work

This study applies k-means algorithm to solve CCP. In the computational study the bench mark data instances are solved to evaluate the tightness measure. Experimental results have shown that the priority applied to handle the capacity constraint can guide the search direction. This study enhances the solution quality of

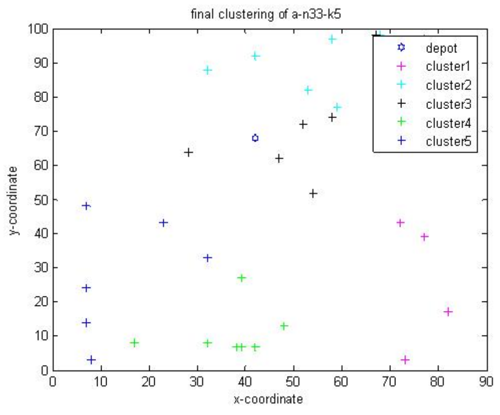

Figure 6: a-n33-k5

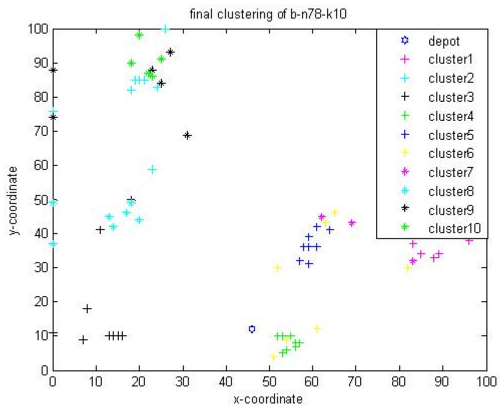

Figure 7: b-n78-k10

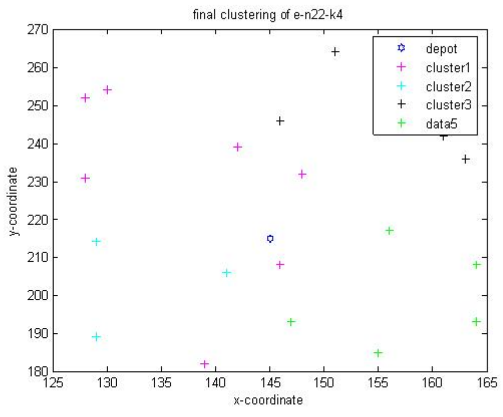

Figure 8: e-n22-k4

CCP with k-mean algorithm. The results obtained are competitive with benchmark results.

Further the CCP application can be modified to CVRP by including traveling salesman problem with in each cluster, facility location problem by predicting the location of facilities like hospital, post office etc. which can be considered as centroids of each clusters. This work can also be extended to computer networks by assigning terminals to concentrators. This work can be applied for finding the location of cluster heads in wireless networks. In multiprocessor environment, the task assigned to the processor can also be performed by modifying the CCP.

References

[1] Baldacci, R., Hadjiconstantinou, E., and Mingozzi, A. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations Research*, 52(5):723–738, 2004.

[2] Chaves, A. A., de Assis Correa, F., and Lorena, L. A. N. Clustering search heuristic for the capacitated p-median problem. *Advances in Software Computing Series(Springer)*, pages 136–143, 2007.

[3] Franca, P. M., Sosa, N. M., and Pureza, V. An adaptive tabu search algorithm for the capacitated clustering problem. *International Transactions in Operational Research*, 6:665–678, 2006.

[4] Garey, M. R. and Johnson, D. S. Computers and intractability: A guide to the theory of np-completeness. *Freeman*, 1979.

[5] Jain, A. K., Murthy, M. N., and Flynn, P. J. Data clustering :a review. *ACM Computing Surveys*, 31(3):1–12, 1999.

[6] Khuri, S., Shütz, M., and Heikötter, J. Evolutionary heuristics for the bin packing problem. *Proceedings of the ICANNGA, Springer-Verlag, Vienna*, pages 285–288, 1994.

[7] Kim, B., Kim, S., and Sahoo, S. Waste collection vehicle routing problem with time windows. *Computers and Operations Research*, 33(12):3624–3642, 2006.

[8] Koskosidis, I. and Powell, W. B. Clustering algorithms for consolidation of customer orders into vehicle shipments. *Transportation Research Part B*, 26:365–379, 1992.

[9] Mulvey, J. and Beck, M. P. Solving capacitated clustering problems. *European Journal of Operational Research*, 18:339–348, 1984.

[10] Nauss, R. M. Solving the generalized assignment problem: An optimizing and heuristic approach. *Informs Journal on Computing*, 15(3):249–266, 2003.

[11] Negreiros, M. and Palhano, A. The capacitated centered clustering problem. *Computers and Operations Research*, 33:1639–1663, 2006.

[12] Osman, I. and Christofides, N. Capacitated clustering problem by hybrid simulated annealing and tabu search. *International Transactions in Operational Research*, 1:317–336, 1994.

[13] Ralphs, T. K. Parallel branch and cut for capacitated vehicle routing. *Parallel Computing*, 29(5):607–629, 2003.

[14] Shieh, H. M. and May, M. D. Solving the capacitated clustering problem with genetic algorithms. *Journal of the Chinese Institute of Industrial Engineers*, 18(3):1–12, 2001.

[15] Thangiah, S. R. and Gubbi, A. V. Effect of genetic sectoring on vehicle routing problems with time windows. *IEEE International Conference on Developing and Managine Intelligent System Projects*, pages 146–153, 1993.

[16] Zalik, K. R. An efficient k-means clustering algorithm. *Pattern Recognition Letters*, 29:1385–1391, 2008.