AEMDS-2013

A Hybrid Approach for Optimization of Capacitated Centered Clustering Problem

Vaishali[1] and Deepak Garg[2]

Computer Science and Engineering Department, Thapar University, Patiala, India.
e-mail: [1]wadhwavaishali@gmail.com; [2]deep108@yahoo.com

Abstract. Capacitated Centered Clustering problem (CPMP) is an important variation of p-median problems. The p-median problems are known to be NP-hard. A hybrid genetic algorithm is used to solve the problem efficiently. The analysis shows that the performance of the proposed algorithm is better than the traditional local search heuristic.

Keywords. Capacitated Centered Clustering, k-means clustering, genetic algorithm.

1. Introduction

Many facility location planning situations in the public and private sectors are concerned with the total or average distance between the facilities and demand points [6]. For example, in public sector locating a network of service providers such as the total distance that customer has to traverse to reach their closest facility is minimized. One classical problem in this area is the p-median problem, in which the objective is to find the location of p new facilities so as to minimize the total weighted distance between demand points and the facilities to which they are assigned. The p-median problem is widely used in both public and private sector location decisions. The uncapacitated version of p-median assumes that each candidate facility to a median can serve any number of demand points. The capacitated p-median problem, on the other hand considers that each candidate facility has a fixed capacity on the number of demand points it can serve [2].

ReVelle and Swain [1] provided an integer programming formulation for p-median problem, which is given in Equation (1) below:

$$\text{minimize } Z = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i d_{ij} Y(ij) \tag{1}$$

Subject to:

$$\sum_{i=1}^{n} Y_{ij} = 1, \forall\, i \tag{2}$$

$$Y_{ij} Y_{jj},\ \forall i, j \text{ pairs} \tag{3}$$

$$\sum_{i=1}^{n} Y_j j = p \tag{4}$$

In this formulation, $Y_{ij} = 1$ if point i is assigned to facility located at j, 0 otherwise; w_i is demand at point i; d_{ij} is the distance between point i and j and p is the number of facilities to be located. The objective function minimizes the total weighted distance between p new facilities and the existing facilities such that the demands of all the customers are satisfied. Constraint (2) specifies that each point is assigned to exactly one facility and constraint (4) specifies that all the facilities must be allocated to at least one point.

The capacitated facility location problem assumes that there is a limited capacity on the demand that can be served. This assumption restricts that a demand may not be assigned to its closest facility. So if Q_j is the capacity of a facility at candidate site j and x_j is a decision variable such that $x_j = 1$ if facility is located at site j and 0 otherwise, then the objective function of (1) has an additional constraint, $\sum_{j\in J} h_i y_{ij} - Q_j x_j \leq 0, \forall i \in I$ which restricts the assignment of points to only open facilities with a limited capacity or matter will need to create these components, incorporating the applicable criteria that follow.

Solving exactly the CFLP is a difficult task. As discussed in above section, for variable number of facilities, this problem is NP-hard. So to solve them

efficiently we need to look for some heuristics for getting some good solutions in acceptable time. Clustering is one such heuristic which plays an important role in optimizing location decisions. The capacitated clustering problem (CCP) is the problem in which a given set of weighted objects is to be partitioned into clusters so that the total weight of objects in each cluster is less than a given value (cluster 'capacity'). The objective is to minimize the sum of dissimilarities to all other objects in the cluster from the 'centre' of the cluster to which they have been allocated. The following formulation is proposed in [8]:

$$\text{minimize} \sum_{i \in I} \sum_{j \in J} |a_i - z_j|^2 y_{ij} \tag{5}$$

Subject to constraints (2), (3), (5) and two additional constraints:

$$\sum_{j \in J} y_{ij} = n_j, \forall\ j \in J \tag{6}$$

$$\sum_{i \in I} a_i y_{ij} = n_j z_j, \forall\ j \in J \tag{7}$$

where z_j = centroid of cluster j, n_j = the number of points in cluster j, a_i is the geometric position of i^{th} point in a two dimensional space, Q_j is the maximum capacity of cluster j, q_i is the demand of i^{th} customer in cluster j, $y_{ij} = 1$ if point i is assigned to cluster j and 0 otherwise, I is the set of demand points and J is the set of clusters $= p$. Here the objective function in (5) minimizes the sum distance between each point and the centroid of the cluster to which it is assigned. Constraint (6) specifies the number of points in each cluster and (7) locates the centroid of each cluster at its geometric center.

In this paper, the CCCP is solved using a hybridized approach where the initial partitions of the k clusters are identified using hybridized genetic algorithm with k-means clustering. The algorithm is known as genetic k-means algorithm (GKMA) which was first proposed by Krishna and Murty [11].

A. K-Means Clustering for CCCP

Due to its relative computational efficiency and ease of implementation, the k-means algorithm is commonly used for clustering large data set. K-means iteratively computes a set of k centers that implicitly represents a partition. Given any set P of centers, each $p \in P$ has a neighborhood defined as the set of data points that are closer to p than to any other center in P. K-means starts with a set P of centers and computes their neighborhoods. In successive iterations, every center is moved to the centroid of its neighborhood and then the neighborhoods are recomputed based on the updated positions of the k centers. This process continues until a convergence criterion is satisfied, such as when two successive iterations produce no changes to any of the k neighborhoods. The collection of neighborhoods that result is taken to be the partition of the data points produced by k-means applied to the initial set of centers. The steps for k-means clustering algorithm are as follows:

Procedure K_means(D,n,k)
Select K points as initial centroids
repeat
  Create K clusters by assigning all points to the closest centroid
  Recompute the centroid of each cluster
until *The centroids does not change*

The k-means algorithm described above is a generic one which works for capacitated centered clustering problems as well. As explained in section 2.3 in CCCP, the problem is to define p clusters with limited capacity. Therefore, each of the iteration of KMA assigns the points to a cluster such that the capacity constraint is not violated.

Unfortunately, k-means is extremely sensitive to the initial choice of centers, and a poor choice of centers may lead to a local optimum that is quite inferior to the global optimum. Because of the limitations of KMA, hybridized genetic algorithm is used for k-means. Genetic algorithms are randomized search and optimization techniques guided by the principles of evolution and natural genetics, and have a large amount of implicit parallelism. They provide global near optimal solutions of an objective or fitness function in complex, large, and multivariate location problems.

B. Genetic K-Means Algorithm – A Hybrid Approach for Solving CCCP

Genetic algorithms are biologically inspired search methods, which are loosely based on molecular genetics and natural selection. In general, a GA contains a fixed-size population of potential solutions over the search space. These potential solutions of the search

space are encoded as binary, integer, or floating-point strings and called chromosomes. The initial population can be created randomly or based on the problem specific knowledge. In each evolution step, a new population is created from the preceding one using crossover and mutation operators. This process is repeated until a fixed number of generations or when no more improvements are seen. In [11], the authors proposed a novel hybrid genetic algorithm that finds a globally optimal partition of a given dataset into a specified number of clusters for k-means algorithm. In this paper an attempt has been made to develop the GKMA process for the capacitated centered clustering problem (CCCP). For genetic k-means clustering a chromosome with k genes is represented by a set of k cluster centers. Each center is a d-dimensional vector containing the center's coordinates. The ordering of the centers in a chromosome is immaterial. A chromosome represents the partition that is obtained by running k-means on its k centers until convergence, which occurs when two successive iterations of k-means yields no change in the partition. Based on these assumptions we summarize the GKA as follows:

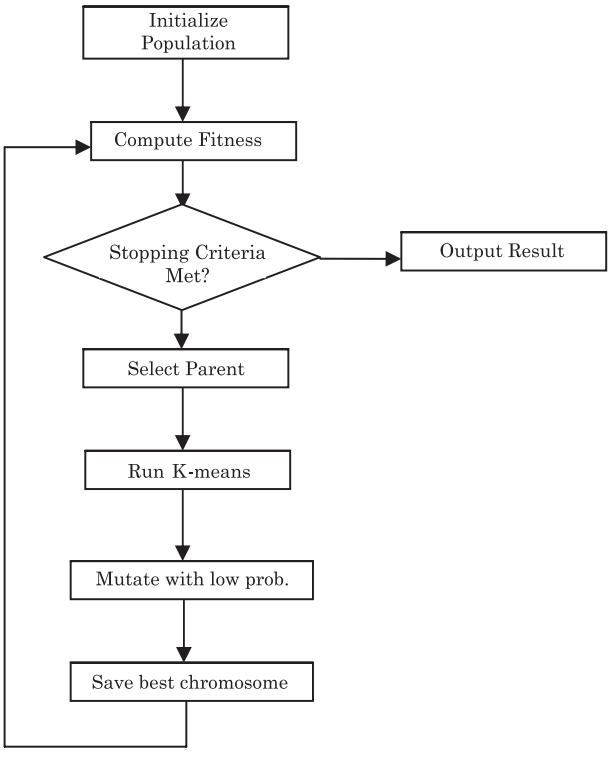

Figure 1. Genetic k-means with crossover operator is replaced with the k-means

//Pseudo Code for Genetic *K*-means
Initialize the population P of points.
Set $A = P_1 // P_i$ *is the* i^{th} *string of P*
for $g = 1$ *to Gen* **do**
Calculate FitnessScore of each string in P using objective function in (5)
Set $P^* = Selection(P)$
for $i = 1$ **to** N **do**
$P_i = Mutation(P^*)$
end for
for $i = 1$ **to** N **do**
$kmeans(P_i)$;
end for
Set B = a solution string in P with highest FitnessScore
if *(FittnessScore(A) > FitnessScore(B))*
then
Set $A = B$
end if
end for

The selection operator randomly selects a chromosome from the previous population according to the distribution given by:

$$P(A) = \frac{F(A_i)}{\sum_{j=1}^{n} F(A_j)} \tag{8}$$

where $F(A_i)$ represents fitness value of the string A_i in the population. The fitness score $F(A_j)$ of point j is computed by solving Eq. (5) for all the demand points within cluster j. The mutation operator mutates the chromosome with a low probability say 0.2. In GKMA k-means operator is hybridized with genetic algorithm instead of traditional crossover operator. For each of g generations, k-means partitions the demand points in k clusters. The fitness score of each demand point in each of k clusters is computed and the fittest point is selected for next population.

2. Simulation Results

We experiment our algorithm on following data set. Data files used in these experiments are chosen among a large range given by MATLAB©. Experiments were done over six different datasets. We performed our tests over a 2D data set that is generated randomly using normal distribution. Dataset 1 consists of 50 points to be clustered into 3 clusters. Dataset 2 Consists of 100 points scattered around 4 specific Clusters. Dataset 4, 5 and 6 Consists of 300,

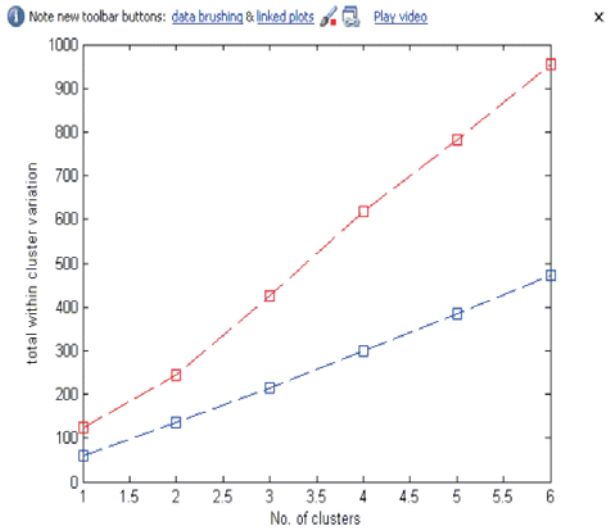

Figure 2. Global convergence of GKMA vs local optimum of KMA

400 and 500 points scattered around 4, 6 and 6 clusters respectively. For each of the previously mentioned datasets the number of clusters is chosen based on our choice; moreover, choosing the number of clusters is still a wide research area that we are not going to discuss in our work. In addition, for each dataset we tested running K-means for 5 or 10 iterations only in addition to running it until it satisfies its original termination condition. Also, we fixed number of GA generations for testing to 1000 generations at most, and then we considered the fittest generation as the error rate. Each dataset was tested for each algorithm for 20 times and then we calculated the average time and error as listed in Table 1.

As explained earlier and shown in Table 1, for each dataset we considered specific number of clusters. For K-Means, we tried to test the effect of number of iterations in the clustering process and on the initialization process as well. KM was always the fastest, but not the most accurate. In addition, random initialization leads kmeans to fall early into local minima.

Comparison with k-means Algorithm (KMA): Figure 2 shows the average error measure obtained in ten independent runs of GKA during the first 100 generations.

It can be observed that, in case of GKA, the average error is always linear with respect to the number of clusters, whereas it is not in case of KMA. This again shows that almost every run of GKA eventually converges to a globally optimal partition. The performance of KMA is not surprising because KMA typically converges to a local optimum. Therefore,

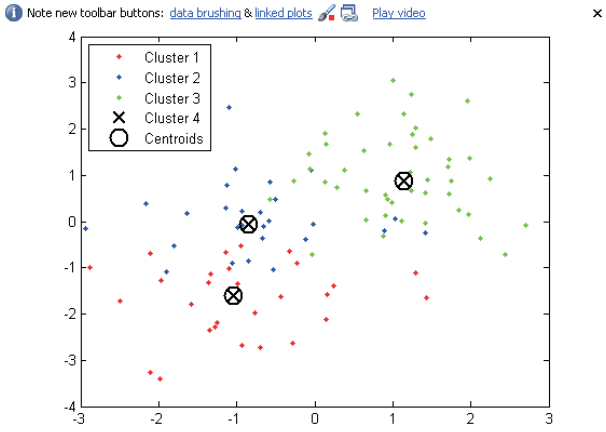

Figure 3a. Scatter of 50 points around 3 cluster using KMA

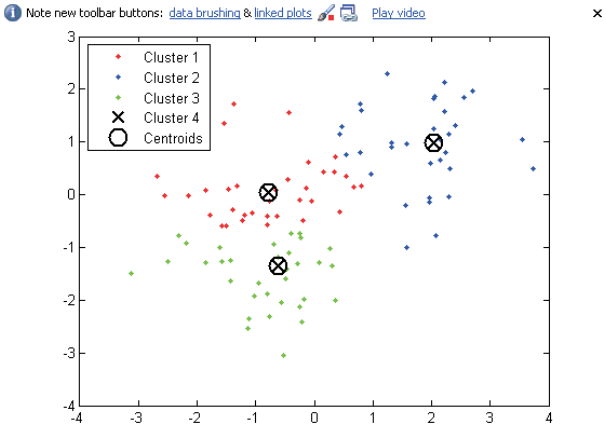

Figure 3b. Scatter of same points around 3 clusters using GKMA

from this graph we can infer that even if KMA starts with the same number of initial configurations as in GKA, it is not assured to reach the global optimum. The situation becomes worse when the search space is large and there are many local optima. The figure also shows that in every iteration/generation, the best and average error corresponding to GKA is less than those corresponding to KMA. The extra computational effort made by GKA in every generation, is that of fitness function and selection operators.

In the next set of experiments shown in Figure 3, GKMA was applied on same data sets. The error measures corresponding to KMA and GKMA with different number of clusters are shown in the table. Figure 3 shows the scatter of points within these clusters. It is observed that GKMA took more time to reach the optimal partition as the number of clusters increases. This is quite obvious since the increase in the number of clusters increases the search space as

Table 1. Performance comparison of KMA and GKMA.

No. of Demand Points	No. of Clusters (k)	Avg. Error (Total within Cluster Variation)		Avg. Time	
		Local Search using K-means (10 Iterations)	Genetic k-means (500 Genrations)	KMA	GKMA
50	3	124.7836	61.3582	0.007	3.728
100	4	243.6541	135.5576	0.008	4.326
200	5	424.3237	216.0598	0.009	11.326
300	5	617.2785	300.7694	0.009	14.842
400	6	780.852	384.2979	0.013	16.203
500	6	955.7227	471.5655	0.014	18.647

well, hence it is more difficult to find a globally optimal solution. However, it is also observed that in all the cases, average scatter of points eventually converged to the global optimum. This is in concurrence with the convergence result derived in the previous section. Moreover, the first two clusters' points have horizontal interleaving on the boundary in case of KMA.

3. Conclusion and Future Scope

Our work is motivated by the observation that the popular k-means method for clustering is very sensitive to the initial set of centers with which it is seeded. The genetic algorithm presented in this paper is used to evolve a good initial set of centers for k-means for the purpose of producing near optimal partitions. Here we look for some other approach such as partitioning the points by facilities rather than by clients or using some better heuristics which is left as a future work. Moreover, genetic k-means clustering approach requires the number of clusters to be known in advance. But there are some better clustering algorithms such as Density Means clustering which can be hybridized with genetic algorithm in place of the k-means approach. Analysis showed that the proposed algorithm is computationally faster than the local search approach. We carried out an experiment on a real data set for locating an emergency service in four different regions of a city which demonstrate that the proposed algorithm has improved significantly the computational time to search demand points to be reallocated. Moreover, the hybrid genetic approach used in our algorithm is independent of initial cluster centers in that it evolves a number of solutions many times and chooses the best one. Hence, the approach of using an evolutionary technique has a significant effect on the performance of our algorithm. Future scope of our work is to enhance the algorithm by using density means clustering in place of k-means in hybrid genetic algorithm as it does not require the number of k clusters to be known in advance.

Acknowledgment

I take this opportunity to express my profound gratitude and deep regards to my guide Dr. Deepak Garg for his exemplary guidance, monitoring and constant encouragement.

References

[1] Zvi drezner and Horst W. Hamacher, Faility Location: Applications and Theory, ISBN 3-530-21345-7, Second Edition, Springer Verlag, Berlin Heidelberg, New York, 2004.

[2] K. Ghoseiri and S. F. Ghannadpour, Solving Capacitated P-Median Problem using Genetic Algorithm, *Proceedings of the 2007 IEEE IEEM*, ISSN 1-4244-1529-2, pp. 885–889, 2007.

[3] R. L. Francis and John A. White, Facility Layout and Location: An Analytical Approach, Prentice Hall, 1992.

[4] E. S. Correa, M. T. A. Steiner, A. A. Freitas, and C. Carnieri, A genetic algorithm for solving a capacitated p-median problem, *Numerical Algorithms*, 35, pp. 373–388, Springer, 2004.

[5] P.-N. Tan, M. Steinbach and V. Kumar, Introduction to data mining, Addison-Wesley, 2005.

[6] Hakimi S. On locating new facilities in a competitive environment, *European Journal of Operations Research*, 12, pp. 29–35, 1983.

[7] Garey, Michael R., Johnson and David, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, ISBN 0-7167-1045-5, 1979.

[8] M. J. N. Negreiros and A. W. C. Palhano, The capacitated centered clustering problem, *Europeon Journal of Computers and Operations Research*, 33(6), pp. 1639–63, 2006.

[9] Sérgio Barreto, Carlos Ferreira, José Paixaõ and Beatriz Sousa, Santos using clustering analysis in a capacitated location-routing problem, *European Journal of Operational Research*, 179, pp. 968–977, 2007.

[10] P. J. Flynn, A. K. Jain and M. N. Murty, Data Clustering: A Review, *ACM Computing Surveys*, 31(3), pp. 264–323, 1999.

[11] K. Krishna and M. Narasimha Murty, Genetic K-Means Algorithm, IEEE Transactions on Systems, *Man, and Cybernetics*, 29(3), June 1999.

[12] Antonio Augusto Chaves and Luiz Antonio Nogueira Lorena, Clustering search algorithm for the capacitated centered clustering problem, *Computers & Operations Research*, 37, pp. 552–558, 2010.

[13] Sanghamitra Bandyopadhyay and Ujjwal Maulik, An evolutionary technique based on K-Means algorithm for optimal clustering, *International Journal of Information Sciences*, 146, pp. 221–237, 2002.

[14] M. Laszlo, S. Mukherjee, A genetic algorithm using hyperquadtrees for low-dimensional k-means clustering, *IEEE Trans. Pattern Anal. Machine Intell.*, 28(4), pp. 533–543, 2006.

[15] Michael Laszlo and Sumitra Mukherjee, A genetic algorithm that exchanges neighboring centers for k-means clustering, Pattern Recognition Letters, 28, pp. 2359–2366, 2007.

[16] Y. Lu, S. Lu, F. Fotouhi, Y. Deng and S. Brown, Fast genetic K-means algorithm and its application in gene expression data analysis, Technical Report TR-DB-06-2003, 2007.
http://www.cs.wayne.edu/∼luyi/publication/tr0603.pdf, 2003.

[17] L. A. Lorena and E. L. F. Senne, Local search heuristic for the capacitated p-median problem, *Networks and Spatial Economics*, 3, pp. 407–19, 2003.

[18] Bashar Al-Shboul and Sung-Hyon Myaeng, Initializing K-Means using Genetic Algorithms, World Academy of Science, *Engineering and Technology*, 54, 2009.

[19] C. S. ReVelle, H. A. Eiselt and M. S. Daskin, A bibliography for some fundamental problem categories in discrete location.