A partition based Clustering of two-dimensional data space as an application of p-median problem

Srikanth Babu Thantadi
Final M.Tech(CSE)
Dept of CSE,
KITE, Korangi
Kakinada
shrikanthmca@yahoo.com

Sunil Nadella
Associate Professor
P.G. Dept of Computer Science
Ideal College of Arts & Sciences
Kakinada
sunilnadella@gmail.com

Hari Prasad Kanchi
First M.Tech(CSE)
Department of CSE,
KITE, Korangi
Kakinada
hariprasadmca07@gmail.com

Abstract

The p-median problem is the most popular location allocation model. It is a well known NP-Hard and combinatorial optimization problem. The field of location allocation modeling is widely used in different application areas. It is used in marketing to analyze customers and for network establishment in cellular tower arrangement which serves maximum clients, in computer networks and in many other areas. Metaheuristics plays an important role in many areas like Operations Research, Algorithm analysis, Data Mining etc. In this paper a new clustering algorithm k-GRASP is proposed, which determines the number of clusters of user choice similar to k-means, and follows metaheuristic approach. Generally, Metaheuristic is a two phase iterative method. So, the proposed algorithm is also encompassed with two phases. First phase ascertains the cluster of user specified length which is similar to k-means algorithm. At this stage the resultant cluster is considered as a best cluster. The second phase strives for the improvement of the cluster so obtained in the first phase. In the proposed work the first phase is termed as Construction phase and the second phase as Enhancement phase. Our empirical results put forward that the proposed k-GRASP clustering algorithm outperforms the other methods. Clustering is the process of dividing the points into similar groups. The proposed GRASP, method can also be used as a clustering algorithm based on the nature of the p-median problem.

Keywords –Clustering, GRASP, Metaheuristic, construction phase and Improvement phase

I. INTRODUCTION

The p-median problem can be stated as Follows, let F be the set of m potential facilities and C a set of n customers. Let d:C x F→ R a function which evaluates the distance between a customer and a potential facility. Given a positive integer p, $p \leq n$, the p-median problem consisting of identifying a subset R of F such that |R| = p and the sum of the distances from each customer in C to its closest facility in R is minimized. Without loss of generality, in this work we use F=C, that is every customer location there is a potential facility.
Mathematically p-median problem is stated as

$$Minimize\ f(d,x) = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} \quad (1)$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i \quad (2)$$

$$x_{ij} \leq y_j \quad \forall i,j \quad (3)$$

$$\sum_{j=1}^{n} y_j = p \quad (4)$$

$$x_{ij} = 0 \text{ or } 1 \quad \forall i,j \quad (5)$$

$$y_i = 0 \text{ or } 1 \quad \forall j \quad (6)$$

Where

n = total number of demand points
x_{ij}=1 if a point is assigned to facility located at j,
=0 other wise
y_i = 1 if the facility is located to point j
= 0 other wise
d_{ij} = distance from points i to j

p = desired number of locations

The objective function (1) minimizes the sum of the distances between the demand points and the desired locations. Constraint (2) guarantees that all demand points are assigned to exactly one facility location. Constraint (3) forbids the assignment of a demand points to a facility that was not selected as a desired location. Constraint (4) defines the total number of desired locations as p. constraints (5 and 6) guarentees that the values of x and y are binary(0 or 1).Since the solution of the p-median problem partions the solution space so we can classify the given space as groups and hence we use the p-median problem as a clustering technique, in this paper we propose a heuristic method GRASP(Greedy Randomized Adaptive Search Procedure) to solve p-median problem which is discussed in detail in section II.

Clustering is the classification of similar objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait. Data clustering is an important Data Mining task and is a common technique for statistical data analysis, which is used in many fields; including mechanical process industries machine learning, pattern recognition, image analysis and bioinformatics.

Metaheuristics represent an important class of approximate techniques for solving hard combinatorial optimization problems, for which the use of exact methods is impractical. They are general-purpose high-level procedures that can be instantiated to explore efficiently the solution space of a specific optimization problem. Over the last decades, metaheuristics, like genetic algorithms, tabu search, simulated annealing, ant systems, GRASP, and others, have been proposed and applied to real-life problems of several areas of science [13].

The GRASP (Greedy Randomized Adaptive Search Procedures) metaheuristic [2, 3], has been successfully applied to solve many optimization problems [4]. The solution search process employed by GRASP is iterative and each iteration consists of two phases: construction and local search. A feasible solution is built in the construction phase, and then its neighbourhood is explored by the local search in order to find a better solution. The result is the best solution found over all iterations.

The rest of the paper is organised as follows: In section II, the proposed k-GRASP algorithm and its phases – Construction and Enhancement phases are described in detail. In section III, conventional k-means algorithm is presented. In section IV, experimental results and comparisons of cluster quality and execution times are anticipated. Section V provides the conclusions.

II. THE GRASP METAHEURISTIC

GRASP [14] is a metaheuristic already applied successfully to many optimization problems [4]. It is a two- phase iterative process. The first phase of GRASP iteration is the construction phase, in which a complete solution is built. Since this solution is not guaranteed to be locally optimal, a local search is performed in the second phase. This iterative process is repeated until a termination criterion is met and the best solution found over all iterations is taken as result.

```
procedure k-GRASP()
1.  best_sol ←∅
2.  repeat
3.   sol ← Construction_phase();
4.   best_sol ← Local_Search_Phase(sol);
5.  if Quality(sol) > Quality(best_sol)
6.    best_sol ← sol;
7.  end if
8.  until Termination criterion();
9.  return best_sol;
```

FIGURE 1: k-GRASP procedure

A pseudo-code of the k-GRASP process is illustrated in Figure 1. At first, the variable that stores the best solution found is initialised. Then the construction phase is executed and then the local search is applied to the constructed solution. The quality of the obtained solution is compared to the current best found and, if necessary, the best solution is updated. At last the best solution is returned.

Figure 2 contains the pseudo-code of the GRASP construction phase implemented in this work. At first the variable *sol*, which stores the best solution found, is initialised and all potential facilities in C are inserted into the candidate list CL, which stores all elements that can be part of the solution. The construction iterations are executed until the solution is completed with p elements. In each iteration, an

element is inserted into the solution.

```
procedure Construction ()
1.  sol ←∅;
2.  CL ← C;
3.  repeat
4.  for each element e in CL
5.  cost[ e] ← Sum of distances between each
    element in C – { sol ∪ {e} } and its closest
    element in sol ∪ { e};
6.  end for;
7.  RCL ←{ e ∈ CL / cost[e] ∈ [mic, mic +
    (mac – mic) * α] };
8.  s ← element randomly selected from RCL;
9.  sol ← sol ∪ {s};
10. CL ← CL – {s};
11. until sol has p elements;
12. return sol;
```

FIGURE 2: Proposed Construction phase used in GRAP

Each element e in CL is evaluated by a greedy function that calculates the cost of the partial solution after the insertion of the element e. The restricted candidate list RCL is generated with all elements in CL whose returned evaluations are in the interval $[mic, mic + (mac - mic) * \alpha]$, where mic and mac are the worst and the best returned values. Then an element s is randomly selected from RCL and it is inserted into the solution sol and the CL is updated. Finally, the best solution found is returned.

In Figure 3, the pseudo-code of the GRASP Enhancement is presented. At first initialize control variables. The function *cost_eval*() evaluates the cost of a solution by computing the sum of the distances between all customers and their closest facilities. Then the neighbourhood of the current solution is visited and if a better solution is found, it becomes the current one, which starts this process again, until no more improvement is made. It is iterated p times.

Next the best solution found is returned. In each iteration, one element r_i of the solution is exchanged by all elements close to it in its partition (cluster) P_i. Here consider that an element e is close to r_i in its partition P_i if the distance between e and r_i is less or equal to the average of distances between r_i and all elements in P_i.

```
procedure Enhancement( sol )
1.  best_sol ← sol;
2.  best_cost ← cost_eval( sol);
3.  repeat
4.  no_improvements ← true;
5.  for i = 1 to p
6.  approx_best_sol ←∅;
7.  approx_best_cost ← ∞;
8.  for each element e in P_i close to r_i
9.  approx_sol ← exchange(best_sol, r_i, e);
10. approx_cost ←
    approx_cost_eval(approx_sol);
11. if approx_cost < approx_best_cost then
12.         approx_best_sol ← approx_sol;
13.         approx_best_cost ←
    approx_cost;
14. end if
15. end for
16. exact_sol_cost ←
    cost_eval(approx_best_sol);
17. if exact_sol_cost < best_cost then
18.         best_sol ← approx_best_sol;
19.         best_cost ← exact_sol_cost;
20. no_improvements ← false;
21. end if
22. end for;
23. until no_improvements;
24. return best_sol;
```

FIGURE 3: Proposed Local Search Phase used in GRASP

In order to reduce the computational effort of the local search, the solution obtained by each exchange is approximately evaluated and only the best one is exactly evaluated. The function *approx_cost_eval*() evaluates approximately the cost of a solution by recalculating the distances only within the partition P_i, without making this computation within the other partitions, which would be necessary for the exact calculation, since there was a change of location. Then it is verified if a better solution than the current one was reached. If so, this new one becomes the current solution and the local search starts again.

III. THE k-MEANS ALGORITHM

The k-means algorithm is one of the most popular heuristics for solving clustering problems and still enjoys widespread use. The algorithm partitions the data into k groups (clusters). The center of each cluster is termed centroids. The algorithm partitions the objects

so as to minimize the sum of the squared distances between the centroids of the clusters and their objects. Figure 4 illustrates the pseudo code.

procedure k-means

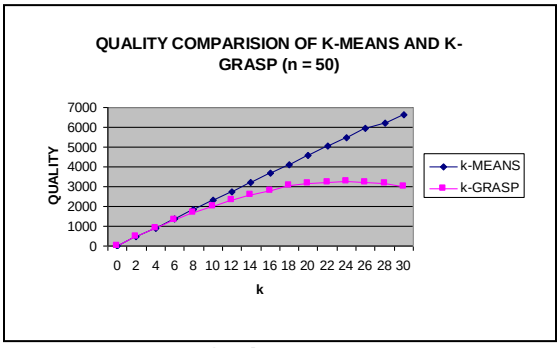
1. Choose k data points to initialize the clusters.
2. For each data point find the cluster center that is closest and assign that data point to the corresponding cluster.
3. Update the cluster centers in each cluster using the mean (centroids) of the data points which are assigned to that cluster.
4. Repeat steps 2 and 3 until there is not any more change in the value of the means

FIGURE 4: k-means algorithm

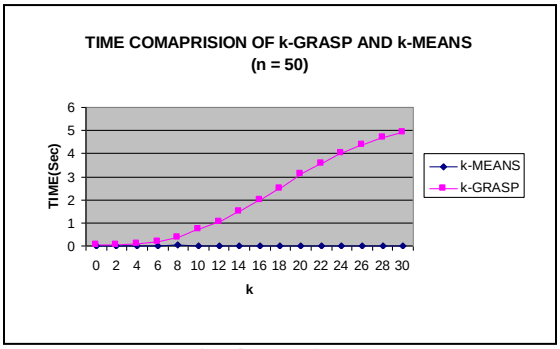
IV. EXPERIMENTAL RESULTS

In this section, the computational results obtained for k-GRASP and k-MEANS are presented and the results are compared on the bases of quality and computation time against k. Experiments are conducted on data sets with 50, 75, 100 points. Results are tabulated and graphs are plotted.

GRAPH-1 and GRAPH-2 are for quality and time comparisons for the data set with 50 points.

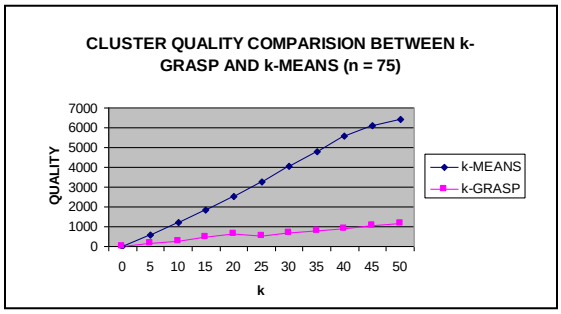


GRAPH - 1



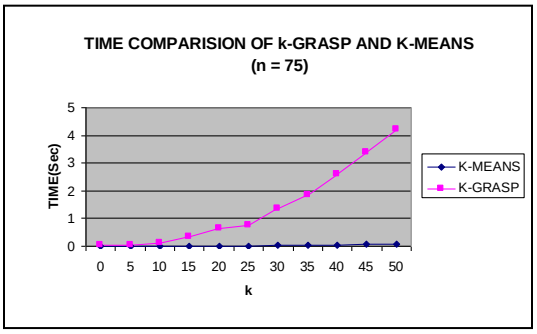
GRAPH - 2

GRAPH-3 and GRAPH-4 are for quality and time comparisons for the data set with 75 points.

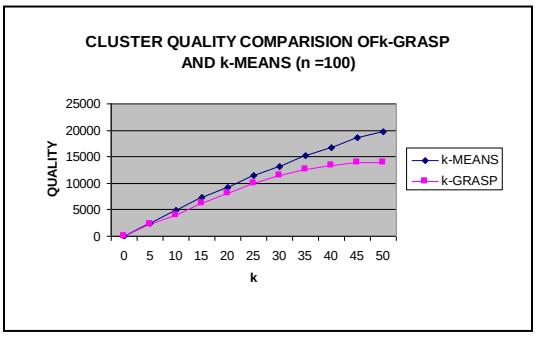


GRAPH - 3

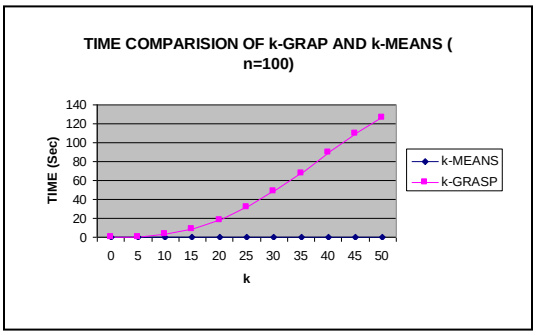
GRAPH-5 and GRAPH-6 are for quality and time comparisons for the data set with 100 points.



GRAPH - 4



GRAPH - 5



GRAPH - 6

V. CONCLUSIONS

It is observed that k-GRASP outperforms k-means in quality aspect and it consumes much more time than k-means because in its Enhancement phase, exchanges are made for improved clusters and this exchanges outperforms k-nearest neighborhood algorithm

REFERENCES

[1] R. Agrwal and R. Srikanth, *Fast algorithms for mining association rules*, Proceedings of the Very Large Data Bases Conference, pp. 487-499, 1994.

[2] T. A. Feo and M. G. C. Resende, *A probabilistic heuristic for a computationally difficult set covering problem*, Operational Reseach Letters, 8 (1989), pp.67-71.

[3] T. A. Feo and M. G. C. Resende, *Greedy randomized adaptive search procedures*, Journal of Global Optimization, 6 (1995), pp. 1609-1624 .

[4] T. A. Feo and M. G. C. Resende, GRASP: *An annotated bibliography*, Essays and Surveys in Metaheuristics, Kluwer Acadamic Publishers, 2002.

[5] M. D. H. Gamal and Salhi, *A cellular heuristic for the multisource Weber Prolem*, computers & Operations Research, 30 (2003), pp.1609-1624.

[6] B. Geothals and M. J. Zaki, *Advances in Frequent Item set Mining Implementations: Introduction to FIMI03*, Proceedings of the IEEEICDM workshop on Frequent Item set Mining Implementations, 2003.

[7] G. Grahne and J. Zhu, *Efficiently using prefix-trees in mining frequent item-sets*, Proceedings of the IEEEICDM Workshop on Frequent Itemset Mining Implementations, 2003.

[8] J. Han, J. Pei and Y. Yin, *Mining frequent patterns without candidate generation*, Proceedings of the ACMSIGMOD International conference on Management of Data, pp. 1-12, 2000.

[9] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd Ed., Morgan Kaufman Publishers, 2006

[10] O. Kariv and L. Hakimi, *An algorithmic approach to network location problems, part ii: the p-medians*, SIAM Journal of Applied Mathematics, 37 (1979), pp.539-560

[11] N. Mladenovic, J. Brimberg, P. Hansen and Jose A. Moreno-Perez, *The p-median problem: A survey of metaheuristic approaches*, European Journal of Operational Research, 179 (2007), pp.927-939.

[12] S. Orlando, P. Palmerimi and R. Perego, *Adaptive and resource- aware mining of frequent sets*, Proceedings of the IEEE International conference on Data Mining, pp.338-345, 2002

[13] I. Osman and G. Laporte, *Metaheuristics: A bibliography*, Annals of Operations Research, 63 (1996), pp. 513-623.

[14] M. G. C. Resende and C. C. Ribeiro, *Greedy randomized adaptive search procedures*, Handbook of Metaheuristics , Kulwer Academic Publishers, 2003.

[15] M. H. F. Ribeiro, V. F. Trindade , A. Plastino and S. L. Martins, *Hybradization of GRASP metaheuristic with datamining techniquess*, Proceedings of the ECAI Workshop on Hybrid Metaheuristics, pp.69-78,2004.

[16] M. H. F. Ribeiro, V. F. Trindade, A. Plastino and S. L. Martins, *Hybrazation of GRASP Metaheuristic with data mining techniques*, Journal of Mathematical Modelling and Algorithms, 5 (2006), pp.23-41.

[17] S. Salhi, *Heuristic Search: The Science of Tomorrow*, OR48 Keynote Papers, Operational Research Society, pp.38-58, 2006.

[18] L. F. Santos, M. H.F. Ribeiro, A. Plastinoand S. L. Martins, *A hybrid GRASP with data mining for the maximum diversity problem*, proceedings of the International Workshop on Hybrid Metaheuristics, LNCS 3636, pp. 116-127, 2005.

[19] L.F. Santos, C.V.Albuquraque, s. L. Martins and A. Plastino, *A hybrid GRASP with data mining for efficient server replication for reliable multicast*, Proceedings of the IEE GLOBECOM Conference, 2006.

[20] L. F. Santos, S. L. Martins and A. Plastino, *Applications of the DM-GRASP heuristic: A survey*, International Transanctions in Operational Research, 15 (2008), p.387-416.

[21] E. G. Talbi, *A taxonomy of hybrid metaheuristics*, Journal of Heuristics, 8 (2002), pp.541-564.

[22] B. C. Tansel, R. L. Fransis, and T. J. Lowe. *Location on networks*: A survey, Management Science, 29 (1983),pp.482-511.