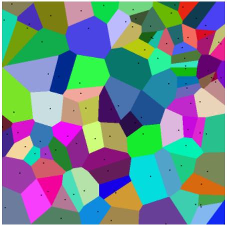
BTP Presentation : Discrete Disc Cover Problem

Raj Kamal
Indian Institute of Technology, Guwahati
r.kamal@iitg.ernet.in
Under the Guidance of Dr. Gautam K. Das

April 29, 2013

Covering Problem

- Minimization problem
- One geometric structure covers another geometric structure with given constraints
- How large dimension should be
- NP Complete Problem
- Approximation and Statistical methods



: Covering with voronio diagram

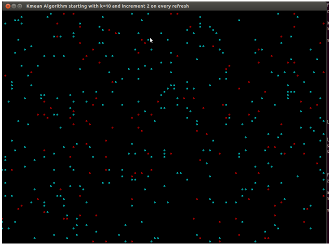
Motivation

- Wireless Networking
  1. setting wireless gateways
  2. selecting wireless gateways
- Variety of facility location problems
- Selecting weather antenna to cover set of cities.
- Image processing

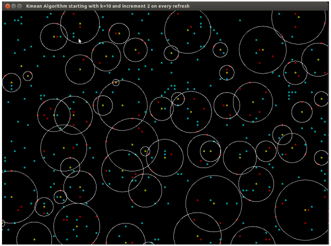
Description

- Red points and Blue points are given in 2D plane.
- Circles can be drawn centered at blue points.
- Red point is covered if it lies inside a circle drawn.
- **Objective**: Minimum number of blue points such that all red points are covered.

- **Input**
  1. n Red Points
  2. m Blue Points
  3. Radius *rad* to be used of drawing circle



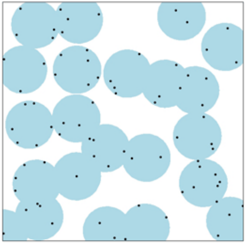
: Input



: Output

Related Works

- **Discrete Unit Disc Cover Problem**
  - a set P of n points.
  - a set D of m unit disks.
  - DUDC problem is finding minimum cardinality subset D^* covering all the points in P.
  - Geometric property is explored.
  - Gautam et al.,[1]



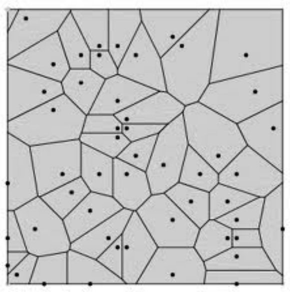
: DUDC Problem

Related Works

- **Minimum geometric disk cover** [[2], [3]].
  - set of n points are given
  - find unit disk of minimum cardinality whose union covers the points
  - disk centers are not restricted to the set but can be arbitrary points from the plane.
- **Discrete K centers**
  - two sets of points P and Q in two dimensional plane and an integer K.
  - find a set of K disks centered on points in P whose union covers Q such that the radius of the largest disk is minimized

Related Works

- Covering with Convex structure. [4].
- Voronoi Diagram are used.



: Voronoi Diagram

Feasibility

- **Input**
  1. set R of n Red Points
  2. set B of m Blue Points
  3. Radius *rad* to be used of drawing circle
- **Feasibility**
  1. Select a red point.
  2. Find distance with every blue point
  3. Any distance $<$ rad, the red point can be covered
  4. Repeat the process for all red point
  5. All red point are covered, feasible otherwise not.

Cluster based approach

- **K Mean Continuous Version**
  - It is similar to standard K Mean algorithm
  - Continuous Case
- **K Mean Discrete Version**
  - Modified K Mean Algorithm
  - Discrete Case

K Mean Continuous Version

Input

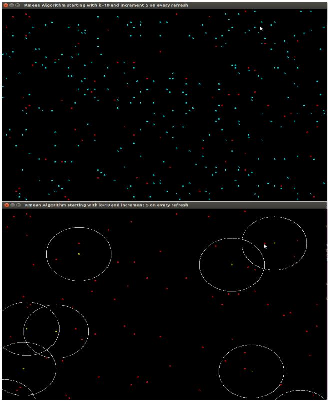
1. n Red Points
2. m Blue Points
3. Radius rad

Preprocessing assuming conditions are feasible

- Create set of red and blue points, each element has property
  1. Coordinates
  2. Cover

K Mean Continuous Version

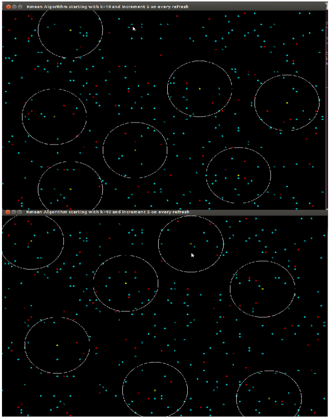
- Fix K
- Select K randomly blue points.
- **Create K cluster.**
  1. Cluster head
  2. set of Red points
  3. size
  4. radius



: cluster Example

Only those red points are in the cluster whose distance from head is less than rad and are covered

- **Change Cluster Head**
  1. We centroid of all red points in set
  2. we change the Cluster head to the centroid
  3. We calculate the difference
- We calculate the $max_d iff$ as maximum of difference
- If max_diff > tolerance repeat.
- We remove unfavorable cluster head.



: Convergence

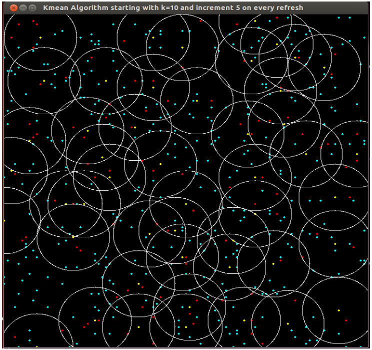
K Mean Continuous Version

- We then check whether all red points are covered or not.
- if not then K is incremented and create cluster and change cluster is repeated.
- We have the cover according to original K mean algorithm
- Cluster head may not coincide with blue point
- We then find nearest blue point to cluster head and create cluster.

K Mean Continuous Version

- We eliminate unfavorable blue points.
- We cover remaining red points

Hence cover is obtained.



: Cluster Example with 100 red point, 400 blue point K=48

K Mean Continuous Version

Lemma: Algorithm Converges in every feasible condition.

- We show that the maximum radii of the cluster decreases or remains same.
- Radius is changed while change in cluster head.
- New cluster head is nearest to previous cluster head.
- New point may come and some may go out.
- New point that has come must have lesser radii than previous maximum.
- Old point goes to new cluster which has lesser radii.
- Hence in all cases maximum radius decreases or remains same.
- Number of red points are fixed .

K Mean Continuous Version

Lemma: Algorithm will find cover.

- Above lemma ensure that the algorithm will converge.
- Converged but than all the red point might not covered.
- Increase the K and repeat.
- At some point we will be able to find cover due to feasibility case.
- hence we will be able to get the cover.

We obtain the following observations

Table: Convergence with 100 red points and 400 blue points

K	Actual Cluster size	Red Points Covered
10	9	34
15	14	57
20	20	81
25	25	85
30	27	87
35	31	92
40	33	93
45	39	97
50	41	99
55	40	98
60	47	99
65	43	100

K Mean Discrete Version

Input

1. n Red Points
2. m Blue Points
3. Radius rad

Preprocessing assuming conditions are feasible

- We create red point and blue point set,each element containing
  1. Coordinates
  2. Cover

Cover in both entity are different.

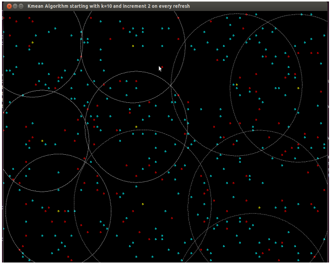
K Mean Discrete Version

- We fix K
- We select K randomly blue points.
- Create K cluster.
  1. Coordinates of cluster head.
  2. Set containing red points.
  3. Radius
  4. Size

Only those red points are in the cluster whose distance from head is less than rad.

K Mean Discrete Version

- Clusters head are the selected blue points
- All Clusters are put in a Cluster Set.



: Cluster Example with 200 red point, 400 blue point K=11

K Mean Discrete Version

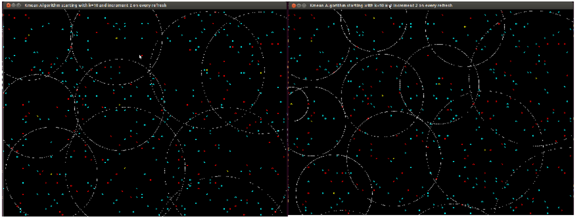
- We calculate radius of each cluster in the Cluster set
- we update the radius of Cluster

Table: Radius of each cluster

Cluster	Radius	size
01	36.2353	24
02	26.4197	16
03	28.3019	11
04	19.2354	14
05	14.0357	3
06	35.4683	24
07	25.4591	23
08	40.6079	40
09	17.8885	9
10	28.0179	15
11	19.2094	11

K Mean Discrete Version

- **Change cluster head**
  - for each cluster, we find a blue point nearest to all points.
  - We update the cluster head to that blue point, update cover.

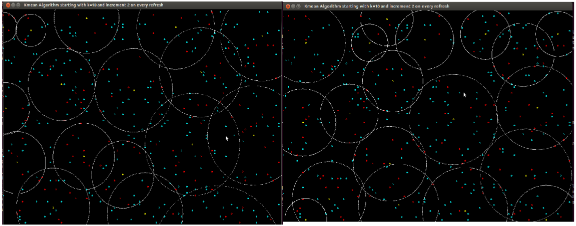


: Change in cluster head

- Calculate the difference b/w old and new cluster head
- Calculate maximum radius.

K Mean Discrete Version

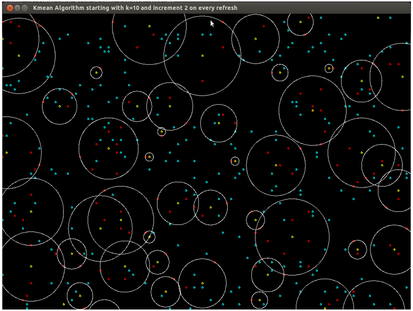
- We calculate maximum difference in change of cluster head
- If maximum difference $>$ tolerance then repeat the change in cluster.
- if maximum radius $>$ rad then increase the K and repeat.



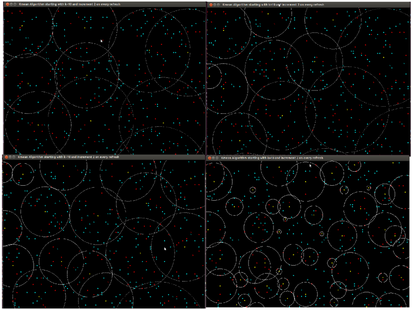
: Change in K

K Mean Discrete Version

- Cover remaining red points.



: Final Cover



: Process of covering

K Mean Discrete Version

Lemma:The K Mean Discrete Version Algorithm converges.

- The maximum radii of the cluster decreases or remains same.
- Radius is changed while change in cluster head
- New cluster head is nearest to previous cluster head to the set.
- New point may come and some may go out
- New point that has come must have lesser radii than previous maximum
- Old point goes to new cluster which has lesser radii.
- Hence in all cases maximum radius decreases or remains same.

K Mean Discrete Version

Lemma:K Mean Discrete Version Algorithm finds the cover.

- Above lemma ensure that the algorithm will converge.
- Converged but than max_rad is greater than rad
- Problem is feasible and above lemma
- Increase the K and we will be able to find a cover.
- Cover remaining
- Hence we will be able to get the cover.

K Mean Discrete Version

- Experimental Results:

Table: Variation of radius

Radius	K	actual K	red points covered
30.46310	010	10	011
23.76970	010	10	013
23.76970	010	10	019
24.35160	025	22	050
20.02500	025	22	060
17.00000	025	22	067
17.00000	025	22	071
16.27880	040	31	057
16.27880	040	31	072
16.27880	040	31	075
17.20470	055	37	064
15.23150	055	37	087

Table: Variation of radius

Radius	K	actual K	red points covered
18.97310	070	40	086
12.64910	070	40	093
12.64910	070	40	096
15.13270	085	44	080
11.31370	085	44	093
11.31370	085	44	095
15.26430	100	46	083
12.08300	100	46	098
16.27880	145	53	095
15.81140	145	53	096
11.40180	160	55	093
09.48683	160	55	100

K Mean Discrete Version

Table: Variation of radius

Radius	K	actual K	red points covered
18.97310	070	40	086
12.64910	070	40	093
12.64910	070	40	096
15.13270	085	44	080
11.31370	085	44	093
11.31370	085	44	095
15.26430	100	46	083
12.08300	100	46	098

Table: Variation of radius

Radius	K	actual K	red points covered
12.08300	100	46	098
12.64910	130	52	095
10.04990	130	52	096
16.27880	145	53	095
15.81140	145	53	096
11.40180	160	55	093
09.48683	160	55	100

- **Algorithm converged in almost every case.**

References

Gautam K. Das, Robert Fraser, Alejandro Lopez-Ortiz, and Bradford G. Nickerson *On the descrete unit disk cover problem*. 5th International Workshop, WALCOM 2011, New Delhi, India, February 18-20, 2011. LNCS-6552 (2011) (146-157)

T. Gonzalez, *Covering a set of points in multidimensional space*, Inf. Proc. Let., 40, (1991) (181-188).

D. Hochbaum and W. Maass, *Approximation schemes for covering and packing problems in image processing and VLSI*, J. ACM, 32, (1985) (130-136).

Gautam K. Das, Sandip Das, Subhas C. Nandy, Bhabani P. Sinha, *Efficient Algorithm for placing a given number of base stations to cover a covex region*J. Parallel Distrib. Comput. 66 (2006) (1353-1358).