ISSN:2231-0711
www.ijcset.net
Mamatha G et al IJCSET |May 2012| Vol 2, Issue 5,1220-1225

Optimum Placement of Additional Sink Nodes in a Wireless Sensor Network

Mamatha G[1] and Dr B G Premasudha[2]

[1]PG Department of Computer Science - MCA
Acharya Institute of Technology, Soladevanahalli, Bangalore-560090
[2]Department of MCA, Siddaganga Institue of Technology, Tumkur, Karnataka,India.
nidhimam.joshi2@gmail.com , bgpremasudha@gmail.com

Abstract—**A new heuristic algorithm for solving the conditional p-center problem is described. In the new algorithm, the initial locations of the p additional sink nodes are selected based on the greedy algorithm. Then the clusters are grown iteratively around the sink nodes which act as the seeds. In successive iterations, the locations of the p additional sink nodes are updated to minimize the maximum weighted distance from a sensor node to its nearest sink node. When all the sensor nodes are included in their respective clusters and after the final update of the p sink node locations, the iteration process is over.**

Keywords—**Wireless Sensor Network, conditional p-center problem, p-medoid clustering .**

I. INTRODUCTION

The sensor nodes of a Wireless Sensor Network (WSN) forward their collected data to their respective sink nodes for further processing and transmission to the base station. Each sink node has a cluster of sensor nodes assigned to it. Each sensor node communicates with its nearest sink node. The power consumed by a sensor node depends on the transmission distance between it and its sink node. In a cluster, the sensor node which is farthest away from its sink node, is worst placed. To mitigate the worst case scenario, the sink nodes are so placed as to minimize the maximum weighted distance between the sink node and its farthest away sensor node in its cluster. This maximum weighted distance is the objective function. Here, the sink nodes are placed at selected sensor node locations among the existing ones to form the appropriate clusters. This is basically a p-center problem [1], [2], [3]. In certain situations, because of historical and administrative reasons, q (number of) sink nodes have already been placed. Then because of expansion and modifications we want to place p additional sink nodes without disturbing the locations of the original sink nodes. Thus the problem is to find the best locations for the additional p sink nodes and to determine the formation of new clusters for minimizing the objective function without disturbing the original q sink nodes. This is the *conditional p-center* problem [4], [5], [6].

Drezner's algorithm [4] solves this problem by solving the unconditional O(log n) p-center sub-problem. In [5], the unconditional p-center sub-problem has to be solved at most n times.

In our proposed solution to this problem, we use a different approach. Here we use the *cluster grow and sink relocate* algorithm as will be described in section III.

II. TERMS, DEFINITIONS AND ASSUMPTIONS

A. *Index Notation for the Nodes of a given WSN*

Let the given WSN contain a total of n nodes. Each node has several attributes (properties) like, its location given by x, y co-ordinates, battery strength (available energy), data size, type of node, etc. The nodes are commonly denoted as node 1, node 2,…, node j,…, node n. The index of node j is j. The index j identifies that specific node j and X(j) represents a set of attributes of that node. In the data set table, columns represent the attributes and the rows represent the indices. The index notation is very useful in clustering and classification of nodes. Consider the of grouping 10 nodes into 3 clusters as,

CL(1) = { 1, 3, 5 }; CL(2) = {2, 4, 6}; CL(3) = {7,8,9,10}.

This simply means node 1, node 3 and node 5 belong to Cluster CL(1) and so on. Another way of representing the same information is to use cluster index attribute array as,
Clidx=[1 2 1 2 1 2 3 3 3 3].

This means node 1 belongs to cluster 1, node 2 belongs to cluster 2 and so on. Here node j belongs Clidx(j) for j=1 to 10.
Consider an example of 16 nodes with 2 classes.

1) Example 1: Let the Sink node set V = {4, 5, 11} and the Sensor node set U = {1, 2, 3, 6, 8, 9, 10, 12, 13, 14, 15, 16}.
Taking a 4x4 square grid for the node locations, the sink and sensor nodes are shown in Fig.1.

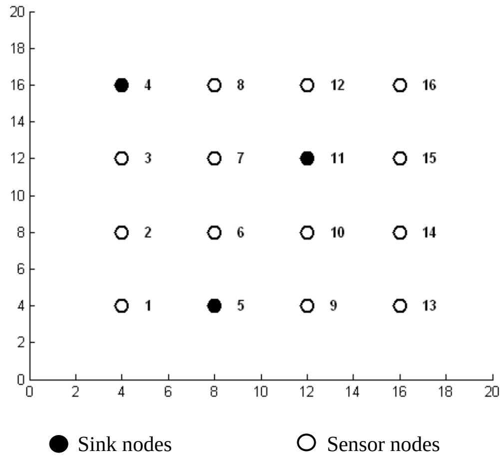


Fig.1. Nodes identified by indices.

The data set corresponding to these nodes could be as shown in Table. 1.

TABLE I

Attributes of 16 nodes. Type 1=Sensor nodes. Type 2= Sink nodes

Node index	x	y	Energy Level	Type
1	4	4	3	1
2	4	8	3	1
3	4	12	3	1
4	4	16	large	2
5	8	4	large	2
6	8	8	3	1
7	8	12	3	1
8	8	16	3	1
9	12	4	2	1
10	12	8	2	1
11	12	12	large	2
12	12	16	2	1
13	16	4	2	1
14	16	8	2	1
15	16	12	2	1
16	16	16	2	1

Thus once the index of a node is known, all its attributes are available from the data set table. In a 2D Euclidean space, the Euclidean distance between two nodes u an v is expressed as,

$$d(u,v) = \sqrt{(x(u) - x(v))^2 + (y(u) - y(v))^2}$$

In the present context the given WSN of n nodes, has q sink nodes and (n-q) sensor nodes. The number of sink nodes q and their locations are already fixed because of administrative and historical reasons. Generally q is far less than n. The sink nodes collect data from the sensor nodes within their respective clusters, aggregate and forward it to the base station. Fixed sink nodes are represented by set V_F as, $V_F = \{v_1, v_2, \dots, v_q\}$. Here $v_1, v_2, \dots, v_q$ are the indices of individual sink nodes. Sensor nodes are represented by set U, as $U = \{u_1, u_2, \dots, u_{n-q}\}$. Here $u_1, u_2, \dots, u_{n-q}$ are the indices of individual sensor nodes. U and V_F are disjoint. The geographical locations and the co-ordinates of these n nodes are fixed and known. Since p number of additional sink node locations are to be chosen from (n-q) locations, p should be less than or equal to (n-q). Normally, p is far less than (n-q). The union of V_F and U gives all the nodes of the WSN. That is,

$G = (V_F) \cup (U)$ represents the whole node set. Here |G| = n.

B. *Cluster Representation*

Every sink node has its own cluster of sensor nodes. The clusters are represented by the cluster set C_r as $C_r = \{C(v_1), C(v_2), \dots, C(v_r)\}$ where r is the total number of sink nodes in the given context. In our algorithm, initially r equals to q and it grows iteratively and finally reaches (q+p). During the growth phase,

$V_r = \{v_1, v_2, \dots, v_q, v_{q+1}, v_{q+2}, \dots, v_r\}$ for $q < r \leq (q + p)$.

C. *Life Time of Sensor Nodes*

The useful life of a sensor node u is taken as [7],

$$T(u) = \frac{E_{init}(u)}{E(u,v)} \quad (1)$$

where, $E_{init}(u)$ is the initial energy available at sensor node u and $E(u,v)$ is the energy consumed per unit time in transmitting data from u to the destination sink node v. Other types of energy consumption like receiving energy, computation energy etc. are neglected as they are relatively small compared to the transmission energy. Assuming that sensor node u transmits its data to its nearest sink node v only, E(u,v) can be expressed as,

$$E(u,v) = R(u) * e(u,v) \quad (2)$$

Here, R(u) is the amount of data sent from u to v in unit time and e(u,v) is the energy required to transmit one unit of data from u to v. The value of e(u,v) depends on the distance between u and v. It can be expressed as [8],

$$e(u,v) = b * d(u,v)^a \quad (3)$$

Here, assuming uniform data transmission speed among the sensor nodes, b is a constant. d(u,v) is the Euclidean distance between u and v. The exponent a is the path loss index having values between 2 to 4. Substituting for E(u,v) in Eq.(1) from Eqs.(2) and (3), we get,

$$T(u) = \frac{E_{init}(u)}{b * R(u) * d(u,v)^a} \quad (4)$$

We introduce the node vulnerability factor F(u) for node u as,

$$F(u) = \frac{b * R(u)}{E_{init}(u)} \quad (5)$$

The value of F(u) is known for each sensor node.
In terms of F(u) from Eq.(5), Eq.(4) becomes,

$$T(u) = \frac{1}{F(u) * d(u,v)^a} \quad (6)$$

Here, u is the sending sensor node and v is the receiving sink node. Since the RHS of Eq.(6) contains u and v as parameters, T(u) depends on both u and v. Hence T(u) is written as T(u,v). Under this modification, Eq. (6) is rewritten as,

$$T(u,v) = \frac{1}{F(u) * d(u,v)^a} \quad (7)$$

Now, maximizing T(u,v), the life time of node u is same as minimizing the denominator $F(u) * d(u,v)^a$. Let this term be designated as weighted distance W(u,v) from u to v. Then W(u,v) is given by,

$$W(u,v) = F(u) * d(u,v)^a \quad (8)$$

For each cluster, the sink node v is unique. Our objective is to locate v and to form the cluster to minimize the maximum intra cluster sensor to sink weighted distance. This in turn maximizes the minimum life time T(u,v).

III. PROPOSED METHOD

A. *Selection of initial p locations for additional sink nodes*

In the given WSN, (n-q) sensor nodes and q sink nodes are already deployed. The initial locations for p additional sink nodes are determined based on the greedy algorithm solution for p-center problem [9]. We place the first additional sink node at the sensor node location farthest away from the existing q sink nodes. That is, the distance from the sensor node to its nearest sink node is maximised. Then we place the next sink node at the location farthest from all the earlier sink nodes, including the recently added one(s). This procedure is repeated until all the p additional sink nodes are placed.

1) Algorithm for initial placement of p sink nodes: Initially q sink nodes are present in the WSN. Let their locations be represented by the location set $A = \{a_1, a_2, \dots, a_q\}$. The sensor node locations are stored in set B as $B = \{b_1, b_2, \dots, b_{n-q}\}$ where (n–q) is the number of sensor nodes. Here, the elements of A and B are the node indices. Note that A = V_F and B = U. A and B are disjoint.

Algorithm 1. INPUT: Set A and B, matrix W(b,a) for b∈B and a∈A. OUTPUT: Augmented sink node set V.

1. Repeat steps 2 through 5, p number of times with updated sets A and B in each repetition.
2. For each b in B, find the minimum weighted distance wmin(b) from b to a, over a∈A as follows.
$$\mathrm{Wmin(b)} = \min\left(\mathrm{W(b,a_1)}, \mathrm{W(b,a_2)}, \dots, \mathrm{W}\left(\mathrm{b,a_q}\right)\right) \quad (9)$$
3. Find that b which maximises wmin(b) over b∈B. Let that b be b_m. That is, get b_m as,
$$b_m = \mathrm{argmax}_{b\in B}\,\{\mathrm{wmin(b)}\} \quad (10)$$
Here, b_m is one of the elements of set B.
4. Add b_m to the sink set to update A by set union as,
$$A = A \cup b_m \quad (11)$$
Now, one additional sink node location b_m has been identified and added to the sink node location set.
5. Delete b_m from set B and update B by set difference as,
$$B = B \setminus b_m \quad (12)$$

Now the updated sink node set V and the unattached sensor node set U are,

$$V = A \quad (13)$$
$$U = B \quad (14)$$

The size of V is (q+p) and it holds the indices of sink nodes. Size of U is (n–q–p).

2) Time Complexity of Algorithm 1: In step 2 or step 3, the number of iterations is (n-q) and these are repeated p times. Hence the time complexity is O(p(n-q)).

3) Example 2: Example 1 of section II is extended. Let n=16. Here the fixed sink node set is V_F = {4, 5, 11} and the sensor node set be U = {1, 2, 3, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16}. Hence, initially set A = V_F and B = U. The additional sinks to be placed are p = 2. In this example q = size(V_F) = 3. These are already shown in Fig.1. Using Algorithm 1, the two additional sink nodes are found at indices 13 and 2. The augmented set A = {4, 5, 11, 13, 2} and V = A and size |V| is (q+p) = 5. Now U = {1, 3, 6, 7, 8, 9, 10, 12, 14, 15, 16}.
The size |U| is (n–q–p) = 11.

B. Formation of Clusters

Now, there are (q+p) sink nodes represented by V. Clusters have to be grown around each of these sink nodes iteratively so that finally, the (q+p) clusters are filled up. While growing the clusters, the newly added p sink node locations are updated so as to be at the 1-*center* of the corresponding clusters.

1) cluster growing : Initially there are (q+p) clusters with one node per cluster. That single node is the corresponding sink node itself. The remaining (n–q–p) sensor nodes represented by U are free. Now, find the nearest distance sensor node sink node pair. That is, find those specific u and v for which W(u,v) is minimum among u∈U and v∈V. The logic here is that the sensor node u, which is nearest to the sink node v, belongs to C(v). In the conditional p-center problem, each additional sink node is placed at location where a sensor node already exists. This holds good for all the p sink nodes added to the WSN.

Initially the cluster corresponding to each sink node contains only that sink node. This fact is represented as,

$$C(v_j) = \{v_j\} \quad (15)$$

for j=1,2,…,(q+p).

One iteration of the cluster growth phase for the given sink node set V and the sensor node location set U can be described as follows.

Algorithm 2. INPUT: Sink and sensor node sets V and U. OUTPUT: Updated C(J) where J belongs to one of the element of V.

1. For each u in U get
m(u) = min(W(u,v)) over v∈V.
2. for each u, store the corresponding v which is nearest to u as
h(u) =v at which the above minimum occurs.
If there are more than one minimum in step 1, use the first minimum.
3. Find the index that minimizes m(u) over u∈U, as,
$$u_m = \mathrm{argmin}_{u\in U}\,\{m(u)\}$$
The sink node nearest to u_m is $h(u_m)$.
4. Add u_m to the cluster set C($h(u_m)$) to grow this cluster by set union as,
$$C(h(u_m)) = C(h(u_m)) \cup u_m$$
Since u_m is added to the cluster, u_m has to be removed from the candidate list U, because one sensor node can join only one cluster. For convenience, call $h(u_m)$=J which is the index of the node found from step 2 and step 4. Then the cluster just grown is C(J). The cluster growth equation above can be rewritten as,
$$C(J) = C(J) \cup u_m$$
6. Remove u_m from U, by set difference operation, as,
$$U = U \setminus u_m \quad (16)$$
7. Exit.

In this algorithm only one node from U is transferred to C(J).

2) Time Complexity of Algorithm 2: Here, step 1 and step 2 are repeated size(U) times. The maximum value for size(U) is n–q. Step 1 and step 2 scan all v's. The size of V is (q+p). Hence, the total number of iteration is 2(q+p)(n–q). Therefore the time comlexity is O(nk), wher k=(q+p) is the total number of sink nodes. That is,

Time Comlexity of Algorithm 2 = O(nk) (17)

C. *Update of sink node locations within its cluster*

Once a cluster is extended to cover a new node, the existing sink node of this cluster may not be its 1-center. Hence, the optimum sink node location is updated so that its new location is the 1-center of this cluster. This update does not apply to the fixed q sink nodes, but applies to the p additional sink node locations determined earlier.

Here, we apply the 1-vertex center algorithm [10] to the cluster C(J) determined in algorithm 2. All the nodes within this cluster are known and is given by the elements of cluster set $C(J_j)$. Let the elements of C(J) be c_1, c_2,.. etc. Then C(J) can be written as,

$$C(J) = \{c_1, c_2, ..., c_m\}$$

where m is the number of elements (nodes) of C(J). The solution to 1-vertex center problem is given by,

$$k = \text{argmin}_{k \in C(J)} \ \max_{i \in C(J)} (W(i,k))$$

That is, for each element k in C(J), find the maximum weighted distance between element i and k over all i∈C(J) as,

$$\text{wmax}(k) = \max_{i \in C(J)} (W(i,k))$$

Now, find that k which minimizes wmax(k) for k∈C(J).

$$k_{min} = \text{argmin}_{k \in C(J)} (\text{wmax}(k)) \qquad (18)$$

This k_{min} specifies the updated sink node location for this cluster. Note that the sink node update operation does not apply to the original fixed q sink node locations.
Update the sink node location set V by removing J from it and adding k_{min} to it. That is replace J in V by k_{min} as,

$$V = V \backslash J \text{ and } V = V \cup k_{min} \qquad (19)$$

The k_{min} given by Eq. (18) need not be different from J. when k_{min} = J, there is no real update because, the operation of Eq. (19) does not produce any change in the sink node locations. V remains same.

IV. CONDITIONAL p-CENTER ALGORITHM

Algorithm 1, 2 and the 1-vortex center algorithms are combined to make up the final algorithm.

Algoritm 3. INPUT: Set V_F, U and the weighted distance matrix W(b,a) for b∈G and a∈G where G is the all node set. That is, G = {1 : n }
OUTPUT: Augmented sink node set V whose size is (q+p), set V_q of additional sink node location set, Cluster sets C(v) for v∈V.

1. Determine the initial locations for the additional p sink nodes and get the set V of all sink node locations using Algorithm 1. See Eq.(13). Size of V is (q+p). The set U of remaining sensor node locations are also available. See Eq.(14). The size of U is (n–q–p). Now V and U are ready. All the (q+p) clusters are single node clusters centered at v∈V.
2. Determine J the cluster index to be grown and grow the cluster C(J) using Algorithm 2. See Eq.(15).
3. Get the updated sensor node location list U from Algorithm 2. See Eq.(16). Size of U decreases by 1.
4. If J∈V_F ,(J is one of the fixed sink node locations,) skip
   Step 5 and jump to step 6.
5. Get k_{min}. See Eq.(18). Relocate the sink node and update the sink node location set as,
   $V = V \backslash J$ and $V = V \cup k_{min}$. See Eq.(19).
6. Check the number of elements in U. If U is not empty go to step 2 for the next iteration.
   When U becomes empty all the additional sink nodes are updated to their final position . Now recluster the entire WSN around these updated sink nodes using the fresh set U which contains the remaining nodes other than the sink nodes.
7. Get the fresh set of all remaining sensor node locations as ,
   $U = G \backslash V$
   Using V and U, build up the cluster around each sink node as follows.
   For each v∈V, reset C(v) = {v}.
   While U not empty
   use Algorithm 2 to grow cluster C(J) by one node.
   (then U decreases by one node)
   Endwhile
   (Now the full cluster set C(v) for all v∈V is ready)
8. Exit.

At the end of Algorithm 3, all the sensor nodes are clustered and the sink node locations of the p additional nodes are determined. The updated set V contains both the fixed sink indices V_F and indices for p additional locations. To separate the p sink node locations from V, use the set difference operation as,

$$V_p = V \backslash V_F \qquad (20)$$

Once the indices of sink nodes are known, their geographical locations can be obtained from the given data set which contains the location attributes (say x, y coordinates) for each indexed node. The cluster information of all the (q+p) clusters are available in C(v) for v∈V. Set $C(v_i)$ gives the indices of the cluster whose sink node is v_i.

A. A. *Time complexity of Algorithm 3*

B. Step 2 which uses Algorithm 2 has the time complexity of O(nk). See Eq. (17). This step itself is repeated a maximum of (n-q) times which is the number of sensor nodes. Hence the time complexity is O(nk(n-q)) which leads to $O(n^2k)$.

C.

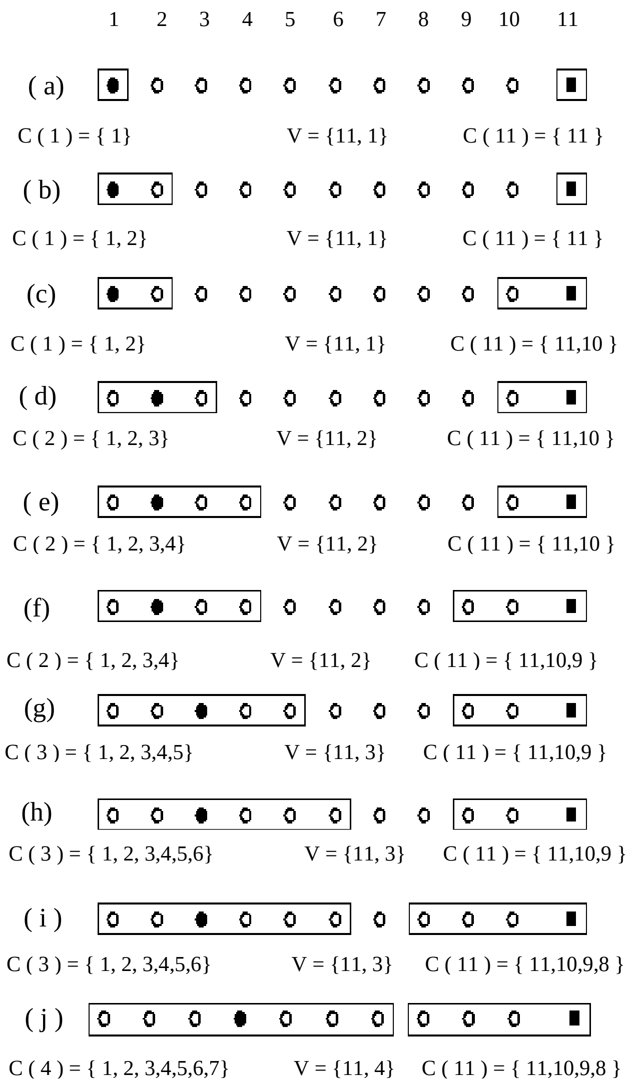
V. EXPERIMENTAL RESULTS

A. *Example 2*

In this simple example, 11 nodes are on a single straight line. This makes the understanding of Algorithm 3 very easy. Here, Node 11 is a fixed sink node. Hence q = 1.Our objective is to optimally locate one additional sink node. Hence p = 1. The weighted distance between adjacent nodes for all nodes from 1 to 10 is taken as 3 while that between 10 and 11 is taken as 4. See Fig 2. The fixed sink node 11 is marked by the black square. Initial placement of the additional sink node at the farthest distance is at location 1. See Algorithm 1. This is indicated by the filled black circle. Initially V_F = {11}. After placing the additional sink at 1, V = {11, 1}. See Fig. 2(a). size of V is (q + p) =2. The clusters are C (11) = {11} and C(1) = {1}.

Here after the cluster growth process and the relocation of sink node activity is shown in Fig 2(a) through 2(j). The cluster to be grown according to to the step 2 of

Algorithm 3 is J = 1 and the node be added to C(1) is 2. Because node pair (2, 1) has the smallest distance compared to that of node pair (10, 11) . Therefore C(1) is grown as C(1)={1, 2}. See Fig. 2(b). In Fig. 2(c), node 10 is added to C(11) . Therefore C(11) = {11,10}.



Fig, 2. Conditional p-center example

In the iteration from Fig. 2(c) to 2(d), not only node 3 is added to cluster C(1), but also the sink node of the cluster is relocated from node 1 to node 2. Thus in Fig. 2(d), V={11, 2}. Similarly, V changes to {11, 3} on transition from Fig. 2(f) to 2(g). Ultimately in Fig. 2(j), the final clustering is over with the additional sink node relocated to location 4 with cluster C(4) = { 1, 2, 3, 4, 5, 6, 7}. The sink node set V = {11,4}.

B. *Example 3*

This example has 81 nodes. Here the number of fixed sink nodes are q = 2. The two fixed sink nodes are indicated by filled black squares in Fig 3. The additional sink nodes to be placed are p = 2. Their locations are found using Algorithm 1 and marked by filled circles. See Fig.3. The weighted distances are taken as the Euclidean distances.

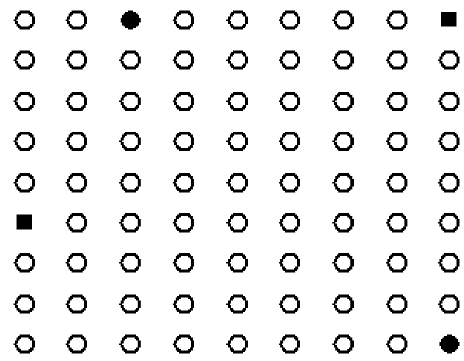


Fig. 3. Node layout of the SNW

■ = Fixed sink nodes
● = Additional sink nodes
○ = Sensor nodes

Fig. 4 gives the solution of conditional p–center roblem. The additional 2 sink nodes have moved to new cations as seen in Fig. 4, from their initial locations as isplayed in Fig.3.

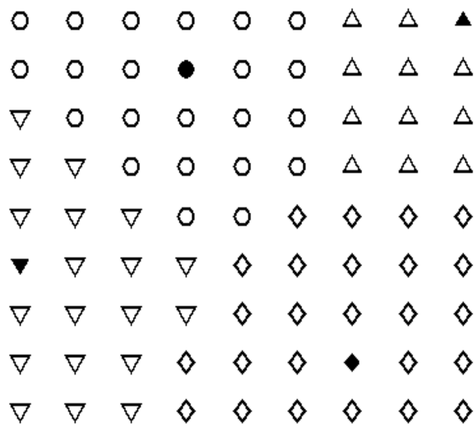


Fig.4. Final solution of the conditional p–center

VI. CONCLUSIONS

A new algorithm is described for solving the conditional p–center problem. This will be very much useful in the design of WSN's.

REFERENCES

[1] Metric k-center. en.wikipedia.org/wiki/Metric_k-center.
[2] The k-center problem.algo2.iti.kit. edu/vanstee/ courses/ **kcenter**.pdf.
[3] D. S. Hochbaum and D. B. Shmoys, "A best possible heuristic for the k-center problem," *Math. Oper. Res.*, vol. 10, no. 2, pp. 180–184, 1985. .
[4] O. Berman and Z. Drezner, "A New Formulation for the *Conditional* p- Median and *p-Center Problems*", Operations Research Letters, Vol. 36, 2008, pp 481-483.
[5] Doron Chen and Reuven Chen, "Relaxation based Algorithm for Solving the conditional p-center problem", Operations Research Letters, 38, 215-217, 2010.
[6] R. Chen and G.Y. Handler ," The Conditional p-Center Problem in the Plane", Naval Res. Log. 40, 117-128 (1993). http://www.tau.ac.il/~chenr/Pubs/chen67.pdf

[7] F. Bouabdallah,, N. Bouabdallah and R. Boutaba," On Balancing Energy Consumption in Wireless Sensor Networks", IEEE Transactions on Vehicular Technology, July 2009, Volume**:** 58, Issue, 6. pp. 2909 – 2924.
[8] E. I. Oyman and C. Ersoy, "Multiple sink network design problem in large scale wireless sensor networks," in *Proc. IEEE ICC*, Paris, France, June. 2004, vol. 6, pp. 3663–3667.
[9] Pankaj K. Agarawal, "k-center Problem". www.cs.duke.edu/courses/spring07/cps296.2/scribe.../lecture14.pdf
[10] Discrete Optimization II. Edited by P. L. Hammer, Ellis L. Johnson, Bernhard H. Korte – 1979. Pp. 341-342. books.google.co.**i**n/books?**i**sbn=0444853227**.**