J. Parallel Distrib. Comput. 66 (2006) 1353–1358

Journal of Parallel and Distributed Computing

Research note

# Efficient algorithm for placing a given number of base stations to cover a convex region

Gautam K. Das, Sandip Das, Subhas C. Nandy*, Bhabani P. Sinha

*Indian Statistical Institute, Kolkata 700 108, India*

Received 12 July 2005; received in revised form 12 April 2006; accepted 8 May 2006
Available online 7 July 2006

**Abstract**

In the context of mobile communication, an efficient algorithm for the base-station placement problem is developed in this paper. The objective is to place a given number of base-stations in a given convex region, and to assign range to each of them such that every point in the region is covered by at least one base-station, and the maximum range assigned is minimized. It is basically covering a region by a given number of equal radius circles where the objective is to minimize the radius. We develop an efficient algorithm for this problem using Voronoi diagram which works for covering a convex region of arbitrary shape. Existing results for this problem are available when the region is a square [K.J. Nurmela, P.R.J. Ostergard, Covering a square with up to 30 equal circles, Research Report HUT-TCS-A62, Laboratory for Theoretical Computer Science, Helsinky University of Technology, 2000] and an equilateral triangle [K.J. Nurmela, Conjecturally optimal coverings of an equilateral triangle with up to 36 equal circles, Exp. Math. 9 (2) (2000)]. The minimum radius obtained by our method favorably compares with the results presented in [K.J. Nurmela, P.R.J. Ostergard, Covering a square with up to 30 equal circles, Research Report HUT-TCS-A62, Laboratory for Theoretical Computer Science, Helsinky University of Technology, 2000; K.J. Nurmela, Conjecturally optimal coverings of an equilateral triangle with up to 36 equal circles, Exp. Math. 9 (2) (2000)]. But the execution time of our algorithm is a fraction of a second, whereas the existing methods may even take about two weeks' time for a reasonable value of the number of circles ($\geqslant 27$) as reported in [K.J. Nurmela, P.R.J. Ostergard, Covering a square with up to 30 equal circles, Research Report HUT-TCS-A62, Laboratory for Theoretical Computer Science, Helsinky University of Technology, 2000; K.J. Nurmela, Conjecturally optimal coverings of an equilateral triangle with up to 36 equal circles, Exp. Math. 9 (2) (2000)].

*Keywords:* Base-station placement; Covering region; Voronoi diagram

## 1. Introduction

In a mobile radio network, a set of base-stations are appropriately positioned in a desired area, and their transmission ranges are assigned. The mobile terminals communicate with its nearest base-station, and the base-stations communicate with each other over scarce wireless channels in a multi-hop fashion. Each base-station emits signals periodically, and all the mobile terminals within its range can identify it as its nearest base-station after receiving such signals. We study the problem of positioning the base-stations and the assignment of transmission ranges such that the entire area under consideration is covered, and the total power consumed by all the base-stations is minimum.

We assume that, the region to be covered is a convex polygon in $\mathbb{R}^2$, the number of base-stations is given a priori, and the range assigned to each of them is same. If the range of a base-station is $\rho$, it can communicate with all the mobile terminals present in the circular region of radius $\rho$ and centered at the position where the base-station is located. Our problem is to minimize $\rho$ by identifying the positions of the base-stations appropriately.

We shall adopt a geometric approach using Voronoi diagram for solving this problem. Experimental results say that the solutions produced by our algorithm is favorably comparable with that of the existing methods for small values of $k$ ($\leqslant 30$) when the region to be covered is a square [15] and an equilateral triangle [14]. The execution time of our algorithm is a fraction of a second for reasonably large values of $k$, but the existing methods need about two weeks' time [15,14]. Thus, our algorithm is very useful for practical applications.

Apart from the base-stations placement for mobile communication, the proposed problem will find relevant applications in the energy aware strategic deployment of the sensor nodes in wireless sensor networks (WSN) [2,5]. In particular, [5] deals with the case where the sensor nodes are already placed. It proposes a distributed algorithm for activating the sensors such that the entire area is always covered, and the total lifetime of the network is maximized. Voronoi diagram is used in another variation of the coverage problem in sensor networks where a set of sensors are distributed in $\mathbb{R}^2$, and a pair of points $s$ and $t$ are given; the objective is to find a path from $s$ to $t$ such that for any point $p$ on the path, the distance of $p$ from its closest sensor is maximized [10].

The organization of the paper is as follows. In Section 2, we will mention some related works on this topic. The algorithm and the experimental results are given in Sections 3 and 4, respectively. Finally, the concluding remarks appear in Section 5.

## 2. Related works

The problem considered in this paper is slightly different from the well-known $k$-center problem in $\mathbb{R}^2$, where we need to place a set $S$ of $k$ supply points on the plane such that the maximum Euclidean distance of a demand point from its nearest supply point is minimized. For a given set $D$ of $n$ demand points, the $k$-center problem can be solved using parametric search technique when $k$ is small. For a fixed value of $k$, the best known algorithm for this problem runs in $O(n^{O(\sqrt{k})})$ time [8]. But, if $k$ is a part of the input, then the problem becomes NP-complete [6]. For a detailed discussion on this problem, see [1]. Another variation of this problem available in the literature is that the center of the (equal-radius) disks are fixed and the objective is to cover the points in $S$ with minimum number of disks. Stochastic formulations of different variations of this problem appeared in [4].

In our case, the set of demand points $D$ is the entire convex region under consideration, and the problem is referred to as a *covering problem* in the literature. Two variations of this problem are studied:

(i) finding the minimum number of unit-radius circles that are necessary to cover a given square, and
(ii) finding the arrangement (positioning) of the members in $S$ and determining a real number $\rho$ such that the circles of radius $\rho$ centered at positions in $S$ can cover the unit square, but for any real number $\rho' < \rho$, there exists no arrangement of $S$ which can cover the entire unit square.

In [18], a lower bound was given for problem (i); it says that if $m$ is the minimum number of unit circles required for covering a square with each side of length $\sigma$, then $\frac{3\sqrt{3}}{2}m > \sigma^2 + c\sigma$, where $c > \frac{1}{2}$. Substantial studies have been done on problem (ii). People tried to cover a unit square region with a given number (say $k$) of equal radius circles, and the objective is to minimize the radius. In [17], graph theoretic approach was adopted to obtain a locally optimal circle covering of a square with up to 10 equal circles. No proof for optimality was given, but later it was observed that their solution for $k = 5$ and 7 are indeed optimal. The same idea was then extended in [7] for covering a rectangle with up to 5 equal circles. Several results exist on covering rectangles and squares with $k$ equal circles for small values of $k$ ($= 6, \ldots 10$, etc.) [13,12]. For a reasonably large value of $k$, the problem becomes more complex. In [15], simulated annealing approach was used to obtain near-optimal solutions for the unit square covering problem for $k \leqslant 30$. As it is very difficult to get a good stopping criteria for a stochastic global optimization problem, they used heuristic approach to stop their program. It is mentioned that, for $k = 27$ their algorithm runs for about 2 weeks to achieve the stipulated stopping criteria. For $k > 28$, the time requirement is very high. So, they have changed their stopping criteria, and presented the results. In [14], the same approach is adopted for covering a equilateral triangle of unit edge length with circles of equal radius, and results are presented for different values of $k$ less than or equal to 36.

## 3. Algorithm

Consider a set of points $P = \{p_1, p_2, \ldots, p_k\}$ inside a convex polygon $\Pi$ where the $i$th base-station is located at point $p_i \in \Pi$. We will use Voronoi diagram [3] of the point set $P$, denoted by $VOR(P)$, to formulate the update mechanism of the positions of the members in $P$ to achieve the optimal placement. $VOR(P)$ is a data structure that stores the partition of the plane into $n$ disjoint convex polygonal region (closed/open), such that, (i) each region contains a member of $P$, (ii) the region containing point $p_i \in P$ is denoted by $vor(p_i)$, and (iii) for any arbitrary point $q \in vor(p_i)$, $\delta(p_i, q) \leqslant \delta(p_j, q)$, for all $p_j \in P$. Here $\delta(p, q)$ denotes the Euclidean distance of the pair of points $p, q$. Since we need to establish communication inside $\Pi$, if a part of the region $vor(p_i)$ goes outside $\Pi$ for some $i$, then the region $vor(p_i) \bigcap \Pi$ is used as $vor(p_i)$. Thus, in our case, $vor(p_i)$ is a closed convex polygon for each $p_i \in P$ (see Fig. 1 for a demonstration).

Note that, all the points inside $vor(p_i)$ are closer to $p_i$ than any other point $p_j \in P$. Thus a mobile user inside $vor(p_i)$ will directly communicate with $p_i$. As all the base-stations are of equal range, our objective is to arrange the points in $P$ inside the region such that the maximum range required ($\rho$) among the points in $P$ is as minimum as possible. Our algorithm is an iterative one. At each step, it perturbs the point set $P$ as described below, and finally, it attains a local minimum.

In each iteration, we compute $VOR(P)$ [3], and then compute the circumscribing circle $C_i$ of each $vor(p_i)$ using the algorithm proposed in [9], for each $i = 1, 2, \ldots, k$. Let $r_i$ denote the radius of $C_i$. It is easy to understand that in order to cover a convex polygon by a base-station with minimum range, we need to place the base-station at the center of the circumscribing polygon of that convex region, and the range assigned to that base-station is equal to the radius of that circle. Thus, for

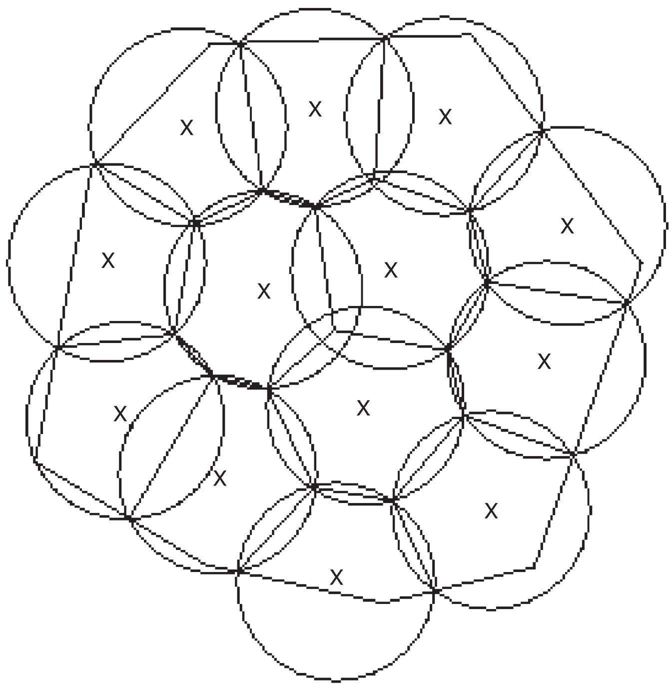


Fig. 1. Illustration of our problem.

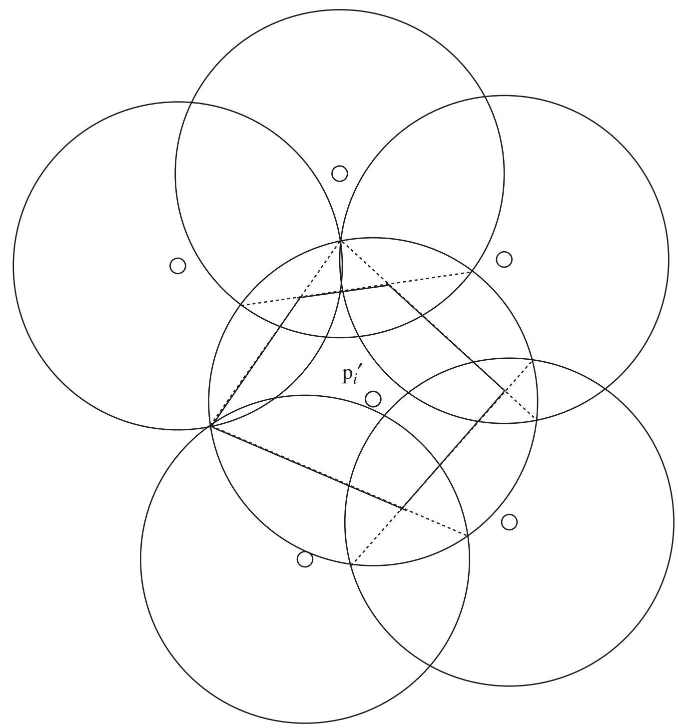


Fig. 2. Illustration of $vor(p_i')$.

each $i = 1, 2, \ldots, k$, we move $p_i$ to the center of $C_i$ and assign range $r_i$ to it. Next, we compute $\rho = \max\{r_i, i = 1, 2, \ldots, k\}$.

**Lemma 1.** *At each iteration*, (i) *the newly assigned position of each point $p_i$ lies inside the corresponding $vor(p_i)$, and* (ii) *the value of $\rho$ decreases*.

**Proof.** (i) The smallest enclosing circle of a convex polygon either passes through the farthest pair of vertices of the polygon, and the line-segment joining that pair of vertices define the diameter of the smallest enclosing circle, or it passes through three or more vertices of polygon. In the first case, lemma obviously follows, and in the second case, if the center lies outside the convex polygon, there exists a vertex-free arc of the smallest enclosing circle of the convex polygon whose length is greater than half of the perimeter of the said circle. This is impossible (see Chapter 16 of [16]).

(ii) Let $\{p_1, p_2, \ldots, p_k\}$ be the position of the base-stations prior to an iteration, and $\rho_{\text{old}}$ be the corresponding value of $\rho$. During the iteration, we compute the Voronoi diagram, and then compute the smallest enclosing circle of each $vor(p_i)$. Let $\{p_1', p_2', \ldots, p_k'\}$ be the center of these circles, and $C^*$ be the largest one among these circles. In other words, $\rho_{\text{old}}$ is equal to the radius of $C^*$. Note that at this iteration, the positions of the base-stations are revised to $\{p_1', p_2', \ldots, p_k'\}$. The Voronoi polygon around each $p_i'$ in the next iteration is obtained as follows:

> Draw copies of $C^*$ with centers at $\{p_1', p_2', \ldots, p_k'\}$. Let $C_i^*$ be the copy of $C^*$ with center at $p_i'$. Consider the line-segments defining the chord-of-intersection of each $C_j^*$ with $C_i^*$ $(j \neq i)$ and the edges of the convex polygon $\Pi$. Next, compute the *envelope* of these line-segments around $p_i'$. This defines the Voronoi polygon $vor(p_i')$ around $p_i'$ for the next iteration. In Fig. 2, the region bounded by the solid lines demonstrates the Voronoi region of $p_i'$; this is the envelope around $p_i'$ of the dotted line segments. It needs to be mentioned that the envelope of a set of line-segments around a point $\alpha$, is a closed polygon containing $\alpha$ such that every point on the boundary of this polygon lies on one or two line-segments of the aforesaid set, and for any point $\beta$ on the boundary of this polygon, the line-segment joining $\alpha$ and $\beta$ will not intersect any of those line-segments excepting the one on which $\beta$ lies.

Note that, for each point $p_i'$, $vor(p_i')$ is properly inscribed by the corresponding circle $C_i^*$ with center at $p_i'$. In the next iteration, we compute the smallest enclosing circle of each $vor(p_i')$ which is completely enclosed in $C_i^*$. This proves that, if $\rho_{\text{new}}$ is the revised value of $\rho$ in the next iteration, then $\rho_{\text{new}} \leqslant \rho_{\text{old}}$. □

**Remark 1.** The iteration terminates when the value of $\rho$ reaches to a local minima, or in other words, $\rho_{\text{new}} = \rho_{\text{old}}$ is attained.

We also apply a refinement step to improve the solution. Note that, if a point (base-station) $p_i$ is on the boundary of $\Pi$, then at least 50% of the area of $C_i$ lies outside $\Pi$, and hence this region need not be covered. This indicates, the scope of further reduction in the area of $C_i$. Thus, if a point goes very close to the boundary of $\Pi$, we move it to the centroid of $\Pi$, which is computed as follows:

> Let $\Pi$ be a $n$ vertex convex polygon, and $(x_j, y_j)$ denote the $j$th vertex of $\Pi$, $j = 1, 2, \ldots, m$. The centroid of $\Pi$ is the point having the coordinates $\left(\frac{1}{n}\sum_{j=1}^{n} x_j, \frac{1}{n}\sum_{j=1}^{n} y_j\right)$.

Table 1
Covering a unit square

| $k$ | $\rho_{opt}$ using method in [10] | $\rho^*_{opt}$ using our method |
|---|---|---|
| 4 | 0.35355339059327376220 | 0.353553 |
| 5 | 0.32616054400398728086 | 0.326165 |
| 6 | 0.29872706223691915876 | 0.298730 |
| 7 | 0.27429188517743176508 | 0.274295 |
| 8 | 0.26030010588652494367 | 0.260317 |
| 9 | 0.23063692781954790734 | 0.230672 |
| 10 | 0.21823351279308384300 | 0.218239 |
| 11 | 0.21251601649318384587 | 0.212533 |
| 12 | 0.20227588920818008037 | 0.202395 |
| 13 | 0.19431237143171902878 | 0.194339 |
| 14 | 0.18551054726041864107 | 0.185527 |
| 15 | 0.17966175993333219846 | 0.180208 |
| 16 | 0.16942705159811602395 | 0.169611 |
| 17 | 0.16568092957077472538 | 0.165754 |
| 18 | 0.16063966359715453523 | 0.160682 |
| 19 | 0.15784198174667375675 | 0.158345 |
| 20 | 0.15224681123338031005 | 0.152524 |
| 21 | 0.14895378955109932188 | 0.149080 |
| 22 | 0.14369317712168800049 | 0.143711 |
| 23 | 0.14124482238793135951 | 0.141278 |
| 24 | 0.13830288328269767697 | 0.138715 |
| 25 | 0.13354870656077049693 | 0.134397 |
| 26 | 0.13176487561482596463 | 0.132050 |
| 27 | 0.12863353450309966807 | 0.128660 |
| 28 | 0.12731755346561372147 | 0.127426 |
| 29 | 0.12555350796411353317 | 0.126526 |
| 30 | 0.12203686881944873607 | 0.123214 |

Table 2
Covering a equilateral triangle

| $k$ | $\rho_{opt}$ using method in [8] | $\rho^*_{opt}$ using our method |
|---|---|---|
| 4 | 0.2679491924311227065 | 0.267972 |
| 5 | 0.2500000000000000000 | 0.250006 |
| 6 | 0.1924500897298752548 | 0.192493 |
| 7 | 0.1852510855786008545 | 0.185345 |
| 8 | 0.1769926664029649641 | 0.177045 |
| 9 | 0.1666666666666666667 | 0.166701 |
| 10 | 0.1443375672974064411 | 0.144681 |
| 11 | 0.1410544578570137366 | 0.141252 |
| 12 | 0.1373236156889236662 | 0.137633 |
| 13 | 0.1326643857765088351 | 0.133379 |
| 14 | 0.1275163863998600644 | 0.127829 |
| 15 | 0.1154700538379251529 | 0.115811 |
| 16 | 0.1137125784440782042 | 0.114574 |
| 17 | 0.1113943099632405880 | 0.112141 |
| 18 | 0.1091089451179961906 | 0.109890 |
| 19 | 0.1061737927289732618 | 0.107288 |
| 20 | 0.1032272183417310354 | 0.104049 |
| 21 | 0.0962250448649376274 | 0.099165 |
| 22 | 0.0951772351261450917 | 0.095877 |
| 23 | 0.0937742911094478264 | 0.094625 |
| 24 | 0.0923541375945022204 | 0.093982 |
| 25 | 0.0906182448311340175 | 0.091688 |
| 26 | 0.0887829248953373781 | 0.090231 |
| 27 | 0.0868913397937031505 | 0.088238 |
| 28 | 0.0824786098842322521 | 0.086795 |
| 29 | 0.0818048133956910115 | 0.084545 |
| 30 | 0.0808828500258641436 | 0.082246 |
| 31 | 0.0798972448089536737 | 0.081665 |
| 32 | 0.0788506226168764215 | 0.080457 |
| 33 | 0.0776371221483728244 | 0.079604 |
| 34 | 0.0763874538343494465 | 0.078827 |
| 35 | 0.0751604548962267707 | 0.076918 |
| 36 | 0.0721687836487032206 | 0.075950 |

It can be shown that, the centroid of a convex region is always inside that region.

It is observed that, such a major perturbation moves the solution from a local minima, and it leads to a scope of further reduction in $\rho$. We again continue the iterations with the perturbed set (mentioned above) as the initial placement until it again reaches another local minima.

The following theorem analyzes the time complexity of each iteration of our heuristic algorithm.

**Theorem 1.** *The worst case time complexity of an iteration is* $O(n + k \log k)$.

**Proof.** The factors involved in this analysis are as follows:

- Computing $VOR(P)$—this can be done in $O(k \log k)$ time [3].
- $VOR(P)$ splits the convex polygonal region $\Pi$ into $k$ closed cells. Each edge of $VOR(P)$ appears in at most two cells. As the number of edges in the region $\Pi$ is $n$, identifying these $k$ cells need $O(n + k)$ time.
- Computing the circular hull of a convex polygon needs time linear in its number of edges [9]. Thus, computing $C_i$ for all $i = 1, 2, \ldots, k$, needs $O(n + k)$ time. □

It is observed that the number of iterations needed to reach to a local optima from an initial configuration is reasonably small. The overall time complexity depends on the number of times we apply the refinement step.

## 4. Experimental results

An exhaustive experiment is performed with several convex shapes of the given region and with different values of $k$. It is easy to show that, for a given initial placement of $P$, at each iteration the value of $\rho$ is decreased. As the process reaches a local minima, the quality of the result completely depends on the initial choice of the positions of $P$. We have studied the problem with random distribution of $P$. It shows that in an ideal solution, the distribution of points is very regular. So, while working with unit square region, we choose the initial placement of the points in $P$ as follows: compute $m = \lfloor \sqrt{k} \rfloor$. If $m^2 = k$, we split the region into $m \times m$ cells, and in each cell place a point of $P$ randomly. If $(k - m^2) < m$, then split the region into $m$ rows of equal width. Then, arbitrarily choose $(k - m^2)$ rows and split each of these rows into $(m+1)$ cells; the other rows are split into $m$ cells. Now place one point in each cell. If $(k - m^2) > m$, then split the square into $(m + 1)$ rows, and each row is split into $m$ or $(m + 1)$ rows to accommodate all the points in $P$.

For each $k$, we have chosen 1000 initial instances. For each of these instances, we have run our algorithm, and have computed

Table 3
Performance evaluation of the algorithm

| $k$ | $\rho^*_{opt}$ | $\rho_{average}$ | SD | Time (s) |
|---|---|---|---|---|
| 4 | 0.353553 | 0.395284 | 0.040423 | 0.052 |
| 5 | 0.326165 | 0.326247 | 0.000201 | 0.073 |
| 6 | 0.298730 | 0.309837 | 0.008433 | 0.090 |
| 7 | 0.274295 | 0.27603 | 0.001668 | 0.107 |
| 8 | 0.260317 | 0.26131 | 0.003079 | 0.124 |
| 9 | 0.230672 | 0.231119 | 0.000540 | 0.143 |
| 10 | 0.218239 | 0.218244 | 0.000004 | 0.164 |
| 11 | 0.212533 | 0.213855 | 0.000894 | 0.184 |
| 12 | 0.202395 | 0.205567 | 0.000908 | 0.206 |
| 13 | 0.194339 | 0.194960 | 0.000645 | 0.228 |
| 14 | 0.185527 | 0.189217 | 0.001722 | 0.258 |
| 15 | 0.180208 | 0.182782 | 0.001883 | 0.279 |
| 16 | 0.169611 | 0.174669 | 0.003178 | 0.303 |
| 17 | 0.165754 | 0.168231 | 0.002336 | 0.327 |
| 18 | 0.160682 | 0.164347 | 0.001092 | 0.351 |
| 19 | 0.158345 | 0.160797 | 0.000885 | 0.377 |
| 20 | 0.152524 | 0.156772 | 0.000877 | 0.405 |
| 21 | 0.149080 | 0.153131 | 0.001253 | 0.436 |
| 22 | 0.143711 | 0.148640 | 0.000582 | 0.465 |
| 23 | 0.141278 | 0.145498 | 0.001738 | 0.499 |
| 24 | 0.138715 | 0.142105 | 0.001507 | 0.531 |
| 25 | 0.134397 | 0.139549 | 0.001572 | 0.557 |
| 26 | 0.132050 | 0.136489 | 0.001618 | 0.587 |
| 27 | 0.128660 | 0.133725 | 0.001298 | 0.623 |
| 28 | 0.127426 | 0.131589 | 0.001357 | 0.655 |
| 29 | 0.126526 | 0.129241 | 0.000964 | 0.688 |
| 30 | 0.123214 | 0.127069 | 0.000881 | 0.719 |

$\rho_{min}$ which is the minimum value of $\rho$ observed during the experiment. Finally, we report $\rho^*_{opt}$ = minimum value of $\rho_{min}$ over all the 1000 instances. Thus, $\rho^*_{opt}$ indicates the minimum value of $\rho$ that is achieved by our experiment. In Table 1, we have compared $\rho^*_{opt}$ with the value of $\rho_{opt}$ obtained by the algorithm in [15] for different values of $k$.

We have also compared our method with that of [14] when the region is an equilateral triangle. The experimental results for different values of $k$ appear in Table 2. Fig. 1 demonstrates the output of our algorithm for covering a given convex polygon with 13 circles.

In order to present the performance of our heuristic, we report the minimum, average and standard deviation of the value of $\rho_{min}$ over all the 1000 instances for different values of $k$ with unit square region (see Table 3). Thus, column 3 of Table 1 is equal to the column 2 of Table 3. We have performed the entire experiment in SUN BLADE 1000 machine with 750 MHz CPU speed, and have used LEDA [11] for computing the Voronoi diagram. The average time for processing each instance is also given. Similar results are observed with equilateral triangles.

## 5. Conclusion

We have presented a very simple algorithm for placing a given number of base-stations in a convex region, and assigning range to them in the context of mobile communication such that every point in the region is covered by at least one base-station, and the maximum range assigned is minimized. This problem is equivalent to covering a convex region by equal radius circles such that the radius of the circles is minimized. Existing methods for this problem can work for only squares, rectangles and equilateral triangles [15,12,14]. But, we did not notice any algorithm for this problem which can work for covering any arbitrary shaped region by a given number of circles of same radius.

We could compare the results produced by our algorithm with that of the existing ones when the region under consideration is a square or an equilateral triangle. Experimental results indicate that the solutions produced by our algorithm are very close to those of the existing results on this problem where the region is a square [15] and an equilateral triangle [14]. It is also mentioned in [14,15] that for a reasonably large value of $k$ ($\geqslant 27$), their algorithms need to run couple of weeks to get the solution, whereas our method needs a fraction of a second. This is very important in the context of the particular application mentioned here.

## References


[1] P.K. Agarwal, M. Sharir, Efficient algorithms for geometric optimization, ACM Comput. Surveys 30 (4) (1998) 412–458.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Comm. Magazine 40 (8) (2002) 102–113.

[3] M. de Berg, M. Van Kreveld, M. Overmars, O. Schwarzkopf, Computational Geometry Algorithms and Applications, Springer, Berlin, 1997.

[4] L. Booth, J. Bruck, M. Franceschetti, R. Meester, Covering algorithms, continuum percolation and the geometry of wireless networks, Ann. Appl. Probab. 13 (2) (2003) 722–741.

[5] A. Boukerche, X. Fei, R.B. Araujo, An energy aware coverage-preserving scheme for wireless sensor networks, in: Proceedings of the second ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, 2005, pp. 205–213.

[6] R.J. Fowler, M.S. Paterson, S.L. Tanimoto, Optimal packing and covering in the plane are NP-complete, Inform. Process. Lett. 12 (1981) 133–137.

[7] A. Heppes, J.B.M. Melissen, Covering a rectangle with equal circles, Period. Math. Hungar. 34 (1997) 65–81.

[8] R.Z. Hwang, R.C.T. Lee, R.C. Chang, The slab dividing approach to solve the Euclidean $p$-center problem, Algorithmica 9 (1993) 1–22.

[9] N. Megiddo, Linear-time algorithms for linear programming in $R^3$ and related problems, SIAM J. Comput. 12 (1983) 759–776.

[10] A. Meguerdichian, F. Koushanfar, M. Potkonjak, M.B. Srivastava, Coverage problems in wireless ad-hoc sensor networks, in: Proceedings of IEEE INFOCOM, 2001, pp. 1380–1387.

[11] K. Mehlhorn, S. Näher, The LEDA Platform of Combinatorial and Geometric Computing, Cambridge University Press, Cambridge, 1999.

[12] J.B.M. Melissen, P.C. Schuur, Improved covering a rectangle with six and seven circles, Electron. J. Combin. 3 (1996) R32.

[13] J.B.M. Melissen, P.C. Schuur, Covering a rectangle with six and seven circles, Discrete Appl. Math. 99 (2000) 149–156.

[14] K.J. Nurmela, Conjecturally optimal coverings of an equilateral triangle with up to 36 equal circles, Exp. Math. 9 (2) (2000).

[15] K.J. Nurmela, P.R.J. Ostergard, Covering a square with up to 30 equal circles, Research Report HUT-TCS-A62, Laboratory for Theoretical Computer Science, Helsinky University of Technology, 2000.

[16] H. Rademacher, O. Toeplitz, The Enjoyment of Mathematics, Princeton University Press, Princeton, NJ, 1957.

[17] T. Tarnai, Z. Gasper, Covering a square by equal circles, Elementary Math. 50 (1995) 167–170.

[18] S. Verblunsky, On the least number of unit circles which can cover a square, J. London Math. Soc. 24 (1949) 164–170.

**Gautam K. Das** received his B.Sc. (Hons.) and M.Sc. degrees in Mathematics from the University of Calcutta in the year 1998 and 2000, respectively, and his M. Tech. in Computer Science from the Indian Statistical Institute, Kolkata in the year 2002. Currently, he is a Senior Research Fellow of Indian Statistical Institute, Kolkata. His current research interest includes algorithmic aspects of Mobile computing, Computational Geometry, etc.

**Sandip Das** received his Bachelor of Statistics (Hons.), Master of Statistics, Master of Technology in Computer Science, and Ph.D. in Computer Science from the Indian Statistical Institute, Kolkata, in 1986, 1989, 1990 and 2001, respectively. From 1990 to 1999, he worked as a Lecturer in the Department of Computer Science, North Bengal University. Since 1999, he is working in the Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata. Currently he is an Assistant Professor. His current research interests include graph theory, combinatorics, geometric algorithm, data structure etc.

**Subhas C. Nandy** received his M.Sc. degree in Statistics from the University of Calcutta, India, M. Tech. degree in Computer Science from the Indian Statistical Institute, Kolkata, India and Ph.D. degree in Computer Science from the University of Calcutta, India, in 1982, 1985 and 1996, respectively. He joined the Indian Statistical Institute, Kolkata in 1986, and is currently a Professor in the Advanced Computing and Microelectronics Unit of Indian Statistical Institute. From 1998 to 1999, he was a Research Associate with the Japan Advanced Institute of Science and Technology, Ishikawa, Japan. In the summer of 2000, he was a Guest Researcher at the City University of Hong Kong, Kowloon. His current research interests include the algorithmic aspects of graph theory and computational geometry.

**Bhabani P. Sinha** received his M. Tech and Ph.D. degrees from the University of Calcutta, India in 1975 and 1979, respectively. In 1976, he joined the Electronics Unit, Indian Statistical Institute, Kolkata, where he is currently a Professor and Head of the Advanced Computing and Microelectronics Unit. He was a visiting faculty at the Southern Illinois University, Carbondale from 1986 to 1987, the University of Central Florida, Orlando in 1998, and Clemson University, Clemson, SC from 2001 to 2002. He has published more than 100 archival research papers in the areas of computer architecture, fault diagnosis, graph algorithms, interconnection networks, parallel processing, and mobile computing. His current research interest includes areas of parallel architecture and algorithms and mobile computing.

Dr. Sinha is a Fellow of the IEEE, USA, Indian National Academy of Engineering, National Academy of Science, India, the West Bengal Academy of Science and Technology, India, and the Institution of Electronics and Telecommunication Engineers, India. He was an Alexander von Humboldt Fellow of Germany from 1979 to 1981 and from 1990 to 1991. In 2000, he was the Indian Space Organization Chair Professor at the Indian Institute of Technology, Kharagpur, India.