J Glob Optim (2012) 52:729–741
DOI 10.1007/s10898-011-9704-3

Radar placement along banks of river

Zhao Zhang · Ding-Zhu Du

Received: 16 March 2010 / Accepted: 24 February 2011 / Published online: 8 April 2011

Abstract In this paper, we consider the Radar Placement and Power Assignment problem (RPPA) along a river. In this problem, a set of crucial points in the river are required to be monitored by a set of radars which are placed along the two banks. The goal is to choose the locations for the radars and assign powers to them such that all the crucial points are monitored and the total power is minimized. If each crucial point is required to be monitored by at least k radars, the problem is a k-Coverage RPPA problem (k-CRPPA). Under the assumption that the river is sufficiently smooth, one may focus on the RPPA problem along a strip (RPPAS). In this paper, we present an $O(n^9)$ dynamic programming algorithm for the RPPAS, where n is the number of crucial points to be monitored. In the special case where radars are placed only along the upper bank, we present an $O(kn^5)$ dynamic programming algorithm for the k-CRPPAS. For the special case that the power is linearly dependent on the radius, we present an $O(n \log n)$-time $2\sqrt{2}$-approximation algorithm for the RPPAS.

Keywords Radar placement · Power assignment · Coverage

1 Introduction

The detection of targets in a region of interest is an important application in the real world. In such an application, the sensing equipment collect information from regions in their observation range, make preliminary decisions about the absence or presence of the targets, and then transmit the information to a base station for collective decision. One possible sensing technology is radar. Since the power consumed by a radar is dependent on its observation range, a natural question is how to locate the radars and how to assign the powers such that the

Z. Zhang (✉)
College of Mathematics and System Sciences, Xinjiang University,
830046 Urumqi, Xinjiang, People's Republic of China
e-mail: hxhzz@sina.com

D.-Z. Du
Department of Computer Science, University of Texas at Dallas, Richardson, Dallas, TX 75080, USA

total power is minimized while the detection requirement is fulfilled. We call such a problem as the Radar Placement and Power Assignment problem (RPPA).

In this paper, we study the RPPA problem along a river, in which a set of crucial points in the river are required to be monitored by a set of radars placed along the two banks. In some real applications, it is not sufficient for every crucial point to be merely monitored, but also monitored by at least k radars. This can be modeled as the k-Coverage Radar Placement and Power Assignment problem (k-CRPPA). In some cases, the locations of the radars are known. To save energy, one only needs to activate a small subset of the radars and assign proper energy to them such that the monitoring requirement is satisfied. Activating may be done from time to time, when the query of crucial points changes dynamically. Such an application can be modeled as a Discrete RPPA problem (DRPPA), in which the task is to choose a subset of locations from a set of given locations instead of placing radars at any place along the two banks.

Suppose the shape of the river is sufficiently smooth. Under this assumption, it is reasonable to view the river as piecewise strips of finite lengths. Hence we may focus on the problem for a strip (the coverage of the whole river can be found by piecing such solutions together with a little additional payment at the concatenation points). The above problems for a strip are abbreviated as RPPAS, k-CRPPAS, and DRPPAS, respectively.

In this paper, we give an $O(n^9)$ algorithm for the RPPAS problem using dynamic programming, where n is the number of crucial points in the strip. Unfortunately, similar idea can not be used to solve the k-CRPPAS problem. For a simpler case in which radars are only placed along the upper bank, we give an $O(kn^5)$ algorithm for k-CRPPAS. It should be noted that our dynamic programming algorithm can also solve the DRPPAS problem. Suppose there are m_1 and m_2 possible locations for the radars on the upper bank and the lower bank, respectively, our method produces an $O(n(m_1m_2)^2)$ algorithm. In particular, if the width of the river can be ignored, i.e., if we regard the river as a line segment, then the algorithm takes time $O(nm^2)$, where m is the number of possible locations for the radars. This improves on the $O((n+m)^3)$ algorithm in [13]. In the special case that $\alpha = 1$, the total energy power is the sum of the radii. We present an $O(n \log n)$-time $2\sqrt{2}$-approximation algorithm for this special case.

2 Related work

Our work falls into the field of *geometric covering*, in which some points or some areas are required to be covered by geometric objects [14]. For example, Chvátal [7] introduced the *Art Gallery Problem*, in which cameras are to be placed to watch every wall of an art gallery room. The room is assumed to be a polygon with n sides and h holes, and the cameras are assumed to have a viewpoint of 360° and rotate at an infinite speed. It was proved that at most $\lfloor (n+h)/3 \rfloor$ cameras are sufficient [11]. A powerful approach called *shifting strategy* was proposed by Baker [3] and Hochbaum and Maas [10], by which numerous geometric covering problems have polynomial time approximation schemes (PTAS), i.e., which can be approximated to a degree of $1+\varepsilon$ for any real positive number ε. As a special case, the disk cover problem in which disks have the same radii and the locations for the disks are not restricted to a given set of possible locations but rather may be chosen at any point of the plane has a PTAS. Disk cover problem with given possible locations was studied in [8], in which the MAX-TSN problem and MIN-BS problem are considered. For the MAX-TSN problem, the aim is to find at most k base stations such that the number of totally supplied demand nodes is maximal. For the MIN-BS problem, the aim is to find a minimum number of base stations such that at least k demand nodes are totally supplied. Under the restriction that the distance between any two base stations is at least a constant d, they presented a PTAS.

The coverage problem has a wide application in wireless sensor networks. In [12], Huang and Tseng studied the problem of deciding whether an area is sufficiently covered, in the sense that every point in the area is covered by at least k sensors. They proved that as long as the perimeter of the sensors are sufficiently covered, the whole area is sufficiently covered. In [15], Wang et al. studied the relationship between coverage and connectivity. They proved that if a region is k-covered then the sensor network is k-connected as long as the communication range is at least twice of the sensing range. In [9], Gupta et al. studied the problem of query execution, in which when a query is sent to the sensor network asking an interested geometric region to be covered, a small subset of sensors has to be selected such that the selected network is connected and satisfies the coverage requirement. They presented a greedy algorithm with performance guarantee $O(\log n)$, where n is the number of sensors in the network. Some works studying the energy-efficient coverage problems can be found in [5,6] etc.

The river coverage problem was also studied in [1], in which the authors ignored the width of the river and modeled the problem as covering a line segment with variable radius disks. In their model, every point on the whole line segment has to be covered. When the radars are of the same type, a simple solution was given. When the radars are of different types, they designed a branch-and-bound algorithm.

The line segment coverage problem was also studied in [2,4,13]. For the problem in which a set of m fixed locations for radars and a set of n locations for the points are given, all located along a fixed line, Lev-Tov and Peleg [13] gave an $O((n+m)^3)$ dynamic programming algorithm. Bilò et al. [13] showed that the problem is polynomial time solvable by reducing it to an integer linear program with totally unimodular matrix. To lower the time complexity, Lev-Tov and Peleg [13] gave a linear-time 4-approximation algorithm. Alt et al. [2] gave a linear-time 3-approximation algorithm and a near-linear-time 2-approximation algorithm. For the problem in which radars are placed on a line to cover points in the upper half plane, Alt et al. [2] designed an $O(n^4 \log n)$ dynamic programming algorithm, and an $O(n \log n) 2\sqrt{2}$-approximation algorithm. In fact, their method is valid for such covering by any L_p disks, with varying corresponding approximation ratios.

3 Dynamic programming algorithm for RPPAS

We adopt the convention that the power consumed by a radar u with observation range $r(u)$ is proportional to $r(u)^\alpha$, where α is usually taken to be a constant between 1 and 4. Hence to assign power to a radar is equivalent to determine its observation range. For two points u, v, we use $d(u, v)$ to denote the Euclidean distance between u and v. The RPPAS problem is formally defined as follows:

Definition 1 (RPPAS) Let $\mathcal{P}$ be a set of points in a horizontal strip. The problem is to place a set $\mathcal{S}$ of radars along the two boundaries of the strip and determine the observation range $r(u)$ for each radar $u \in \mathcal{S}$ such that for each point $p \in \mathcal{P}$, there is at least one radar $u \in \mathcal{S}$ satisfying $d(u, p) \le r(u)$, and the total power $w(\mathcal{S}) = \sum_{u \in S} w(u)$ is as small as possible, where $w(u) = r(u)^\alpha$ is the power assigned to u and $\alpha \in [1, 4]$ is a constant.

In this section, we present a dynamic programming algorithm which can find the exact solution in time $O(n^9)$.

Suppose the strip is between lines $y = y_1$ and $y = y_2$ with $y_1 < y_2$, called *lower bank* and *upper bank* respectively. A *candidate disk* is the smallest disk whose center is located on the upper bank or the lower bank such that it covers a given set of points (which is equivalent to that its circumference contains one or two points). Since $\alpha \geq 1$, it is easy to see that there exists an optimal solution composed of only candidate disks. For simplicity of statement, we call a candidate disk whose center is on the upper (resp. lower) bank as an upper (resp. a lower) disk. We also regard the empty set as an upper disk as well as a lower disk, with radius zero. Suppose there are t_u upper disks and t_l lower disks. Clearly, $t_u, t_l \leq 2\left(n + \frac{n(n-1)}{2}\right) + 2 = O(n^2)$.

For two upper disks u and u', we say that u is to the left of u' if the x-coordinate of the center of u is no greater than the x-coordinate of the center of u'. The same convention applies to two lower disks. Suppose the points in the strip are ordered from left to right as $p_1, p_2, \ldots, p_n$ (breaking ties arbitrarily). For $1 \leq j \leq n$, an upper disk u and a lower disk l with $p_j \in u \cup l$, define $S(j; u, l)$ to be the optimal solution containing only candidate disks such that

(i) all points $p_1, \ldots, p_j$ are covered by the disks in $S(j; u, l)$;
(ii) u is the rightmost one among all upper disks in $S(j; u, l)$;
(iii) l is the rightmost one among all lower disks in $S(j; u, l)$.

For convenience of statement, a set of disks satisfying the above conditions but not necessarily having the minimum total weight is called a *feasible solution for the triple* (j, u, l). The total weight of disks in $S(j; u, l)$ is denoted by $s(j; u, l)$. Then $s(j; u, l)$ can be determined in the following way. If $p_j \notin u \cup l$, set $s(j; u, l) = \infty$. If $p_j \in u \cup l$, then

$$s(j; u, l) = \min\{s(j-1; u', l') + \delta(u' \neq u)w(u) + \delta(l' \neq l)w(v)\}, \tag{1}$$

where $\delta(u' \neq u)$ is 1 if $u' \neq u$ and is 0 if $u = u'$, and the minimum is taken over all upper disks u' to the left of u and all lower disks l' to the left of l. We are to show that the solution corresponding to $\min\{s(n; u, l) \mid \text{u is an upper disk and l is a lower disk}\}$ is an optimal solution to RPPAS.

Theorem 2 *The above dynamic programming algorithm correctly computes an optimal solution to RPPAS in time $O(n^9)$.*

Proof We are to show the equality of (1).

First, we show the '$\leq$' part. Suppose $p_j \in u \cup l$. For every pair of u', l' as described above such that $s(j-1; u', l') < \infty$, $S(j-1; u', l') \cup \{u, l\}$ is clearly a feasible solution for (j, u, l). Hence

$$w\left(S(j-1; u', l') \cup \{u, l\}\right) \geq s(j; u, l). \tag{2}$$

Since in an optimal solution, no disk is contained completely in another one, we see that if $u' \neq u$, then $u \notin S(j-1; u', l')$ (otherwise u and u' would have the same center by condition (ii) of $S(j; u, l)$ and $S(j-1; u', l')$, and thus one is contained in the other). The same is true for l' and l. Hence

$$w\left(S(j-1; u', l') \cup \{u, l\}\right) = s(j-1; u', l') + \delta(u' \neq u)w(u) + \delta(l' \neq l)w(v). \tag{3}$$

Combining inequality (2) and equality (3) with the arbitrariness of u', l', the '$\leq$' part of (1) is proved.

For the '$\geq$' part, consider an optimal solution $S(j; u, l)$. If $p_{j-1} \in u \cup l$, then $S(j; u, l)$ is a feasible solution for $(j-1, u, l)$, and thus $s(j; u, l) \geq s(j-1; u, l) \geq$ the righthand

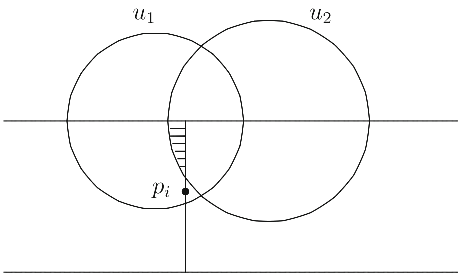


Fig. 1 The points covered by u_2 to the left of p_i are all covered by u_1

side of (1). Hence suppose $p_{j-1} \notin u \cup l$. We first claim that for an integer $i < j$ and two upper disks $u_1, u_2 \in S(j; u, l)$ such that $p_i \in u_1 \setminus u_2$ and u_1 is to the left of u_2, all the points to the left of p_i which are covered by u_2 are also covered by u_1. In fact, since in an optimal solution, no disk is contained completely in another disk, we see that the positions of u_1 and u_2 are as in Fig. 1, and p_i locates to the left of the vertical line through the two intersection points of u_1 and u_2. Then the claim follows easily. Similar claim also holds for lower disks. Suppose, without loss of generality, that p_{j-1} is contained in an upper disk. Let u' be the rightmost upper disk of $S(j; u, l)$ *containing* p_{j-1}. Then u' must be the rightmost one *among all upper disks* in $S(j; u, l) \setminus \{u\}$. Otherwise, let $u''(\neq u')$ be the rightmost upper disk in $S(j; u, l) \setminus \{u\}$. Then $p_{j-1} \in u' \setminus u''$ and u' is to the left of u''. By the above claim, we see that points in u'' which are to the left of p_{j-1} are all covered by $u' \cup u \cup l$. Hence deleting u'' from $S(j; u, l)$ results in another feasible solution for (j, u, l) with lower weight, which contradicts the optimality of $S(j; u, l)$. Also by the above claim, since $p_{j-1} \in u' \setminus u$ and u' is to the left of u, all points in u which are to the left of p_{j-1} are also covered by u'. Now we see that $S(j; u, l) \setminus \{u\}$ is a feasible solution for $(j - 1, u', l)$, and thus $s(j; u, l) \geq s(j - 1; u', l) + w(u) \geq$ the righthand side of (1).

For each $j = 1, 2, \ldots, n$, the algorithm maintains a table of size $t_u \times t_l$, each entry can be computed in at most $O(t_u t_l)$ time. Since $t_u, t_l \leq O(n^2)$, the computation time is at most $O(n(t_u t_l)^2)$ which is $O(n^9)$. □

The above method can also be applied to other Radar Placement models with the following conditions, and possibly some of their combinations.

(a) The set of candidate positions for radars is given.
(b) The radii of the radars are fixed.
(c) The weight is a more complicated function.

4 Dynamic programming algorithm for k-CRPPAS along one bank

In this section, we consider the k-CRPPAS problem which is formally defined as follows:

Definition 3 **(k-CRPPAS)** The k-Coverage Radar Placement and Power Assignment problem for a strip is a RPPAS problem with the additional requirement that for each point $p \in \mathcal{P}$, there are at least k radars $u_1, \ldots, u_k \in \mathcal{S}$ satisfying $d(u_i, p) \leq r(u_i)$ $(i = 1, \ldots, k)$.

Assume that multiple radars can be placed at a same location. Hence a solution to the problem is a *multiset*, counting multiplicity when some radars with the same radii are located at the same position. To distinguish with the union operation '∪' for an ordinary set, we use

$S + \{u\}$ to denote adding an element u to the multiset S, even though some copy of u is already in S. For simplicity of statement, $S + \{0\}$ means nothing is added into S.

In the following, we assume that radars can only be placed along the upper bank. We call such a problem a *One Bank RPPAS* problem. In this case, the k-CRPPAS problem can be solved dynamically in the following way.

For $1 \le j \le n$, an integer $1 \le m_j \le k$, an upper disk u with $p_j \in u$, define $S(j, m_j; u)$ to be the optimal solution consists of only upper disks such that

(i) every point p_i with $i \le j-1$ is covered by at least k disks in $S(j, m_j; u)$, and p_j is covered by at least m_j disks in $S(j, m_j; u)$;
(ii) u is the rightmost one among all upper disks in $S(j, m_j; u)$.

A solution satisfying the above conditions but not necessarily optimal is called a *feasible solution for* (j, m_j, u). For convenience of statement, we allow $m_j = 0$ and let $S(j, 0; u) = S(j-1, k; u)$. Denote by $s(j, m_j; u) = w(S(j, m_j; u))$. Then $s(j, m_j; u)$ can be calculated as follows. If $p_j \notin u$, set $s(j, m_j; u) = \infty$. If $p_j \in u$, then

$$s(j, m_j; u) = \min\{s(j, m_j - 1; u') + \gamma \cdot w(u)\}, \tag{4}$$

where the minimum is taken over all upper disks u' to the left of u, and the function γ is determined as follows: in the case that $u' \neq u$, set $\gamma = 1$; in the case $u' = u$, if $S(j, m_j - 1; u')$ covers p_j by at least m_j times, then $\gamma = 0$, if $S(j, m_j - 1; u')$ covers p_j by exactly $m_j - 1$ times, then $\gamma = 1$.

We are to show that the solution corresponding to $\min\{s(n, k; u)\}$ is an optimal solution to the One Bank k-CRPPAS.

Theorem 4 *The above dynamic programming algorithm computes an optimal solution for One Bank k-CRPPAS problem in time $O(kn^5)$.*

Proof Suppose $p_j \in u$. For any optimal solution $S(j, m_j - 1; u')$ for $(j, m_j - 1, u')$ with $s(j, m_j - 1; u') < \infty$, where u' is to the left of u, by the definition of γ and the assumption $p_j \in u$, we see that $S(j, m_j - 1; u') + \{\gamma \cdot u\}$ is a feasible solution for (j, m_j, u). Hence $s(j, m_j; u) \le s(j, m_j - 1; u') + \gamma \cdot w(u)$. The '$\le$' part of (4) follows from the arbitrariness of u'.

For the '$\ge$' part, if $m_j > 1$, then $S(j, m_j; u)$ is a feasible solution for $(j, m_j - 1, u)$, and thus $s(j, m_j; u) \ge s(j, m_j - 1; u) \ge$ the righthand side of (4). Hence suppose $m_j = 1$. We are to show that

$$s(j, 1; u) \ge \min\{s(j-1, k; u') + \gamma \cdot w(u)\}. \tag{5}$$

If $p_{j-1} \in u$, then $S(j, 1; u)$ is also a feasible solution for $(j-1, k, u)$ and thus (5) holds. Hence suppose $p_{j-1} \notin u$. Suppose the set of upper disks in $S(j, 1; u)$ containing point p_{j-1} are $u_1, u_2, \ldots, u_q$ ordered from left to right. Clearly, $u_1, \ldots, u_q \neq u$. Since p_{j-1} is covered by at least k upper disks in $S(j, m_j; u)$, we have $q \ge k$. By the claim in the proof of Theorem 2,

$$\begin{array}{c}\text{every point to the left of } p_{j-1} \text{ which is covered by } u\\ \text{is also covered by } u_1, u_2, \ldots, u_q.\end{array} \tag{6}$$

Hence $S(j, 1; u) - \{u\}$ (counting multiplicity) covers every point $p_1, \ldots, p_{j-1}$ at least k times. Set $u' = u_q$. Similarly to the proof of Theorem 2, we see that u' is the rightmost disk among all disks in $S(j, 1; u) - \{u\}$. Hence $S(j, 1; u') - \{u\}$ is a feasible solution for $(j-1, k; u')$. It follows that $s(j, 1; u) \ge s(j-1, k; u') + w(u) \ge$ the righthand side of (5).

The time complexity of the algorithm is $O(nkt_u^2)$ which is $O(kn^5)$. □

Unfortunately, the above idea failed to be generalized to solve the k-CRPPAS problem when radars can be placed on both banks. The difficulty lies in (6), which does not hold if some u_i is a lower disk, and thus some point to the left of p_{j-1} might be covered less than k times by $S(j, 1; u) - \{u\}$.

5 Approximation algorithm for radii sum

Since the time complexity of the dynamic programming algorithm in Sect. 3 is very high, we look for approximation algorithms to the RPPAS problem. In this section, we present an $O(n \log n)$-time $2\sqrt{2}$-approximation algorithm for the special case $\alpha = 1$. Such a case occurs when the transmission focuses in a narrow angle beam whose direction can change from time to time and adapt to the needs of the network. The study for such a case may serve as a basis towards more general non-linear cases.

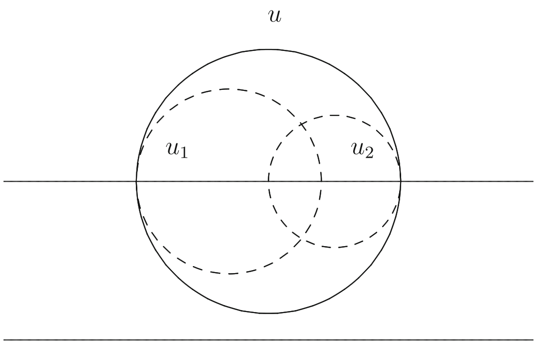
When $\alpha = 1$, the total power to be minimized is the radii sum $\sum r(u)$. The advantage of this case is that in an optimal solution, the upper disks are mutually disjoint. In fact, if two upper disks u_1 and u_2 have a non-empty intersection (suppose u_1 is to the left of u_2 and one is not contained in the other), then they can be replaced by a new upper disk u with radius at most $r(u_1) + r(u_2)$ whose center is at the middle point of the line segment between the leftmost point of u_1 and the rightmost point of u_2 (see Fig. 2). We call such a replacement as *mergence*. Clearly, mergence does not increase the radii sum. The same observation holds for lower disks.

We first introduce some terminologies. For a set $\mathcal{U}$ of upper disks, the *diameter* $Diam(\mathcal{U})$ is the distance between the leftmost point and the rightmost point of all disks in $\mathcal{U}$. The same concept $Diam(\mathcal{L})$ applies to a set $\mathcal{L}$ of lower disks. Suppose the height of the strip is a. A point in $\mathcal{P}$ is called an *upper point* if it lies at most $a/2$ distance away from the upper bank, otherwise, it is called a *lower point*.

Our algorithm finds a set of upper disks $\mathcal{U}$ covering the set of upper points, and a set of lower disks $\mathcal{L}$ covering the set of lower points, respectively. Then it outputs $\mathcal{U} \cup \mathcal{L}$. Next, we show how to find $\mathcal{U}$. The set $\mathcal{L}$ can be found similarly.

Clearly, disks in $\mathcal{U}$ are mutually disjoint. By replacing 'upper' with 'lower' in Algorithm 5, we obtain a set $\mathcal{L}$ of mutually disjoint lower disks covering the set of lower points. It should be pointed out that finding $\mathcal{U}$ and $\mathcal{L}$ are two independent processes, upper disks can only be merged into upper disks, lower disks can only be merged into lower disks. It should be noted that although $\mathcal{U}$ is found by considering only upper points, some lower points can also be

Fig. 2 Merging u_1 and u_2 into u



Algorithm 5

Input: An set of upper points $\mathcal{P}$.
Output: A set $\mathcal{U}$ of upper disks covering all points in $\mathcal{P}$.
Order points in $\mathcal{P}$ from left to right as $p_1, \ldots, p_s$ (breaking ties arbitrarily).
Set $\mathcal{U} = \emptyset$, $u_0 = \emptyset$, $i = 1$.
while $\mathcal{P} \neq \emptyset$ **do**
  Let p be the leftmost point in $\mathcal{P}$.
  Let u_i be the upper disk which has p as the lower extreme.
  if $u_i \cap u_{i-1} \neq \emptyset$ **then**
    merge u_{i-1} and u_i to a larger upper disk u.
    Set $\mathcal{U} = (\mathcal{U} - \{u_{i-1}\}) \cup \{u\}$, $u_{i-1} = u$.
    Delete from $\mathcal{P}$ all points covered by u.
  else
    Set $\mathcal{U} = \mathcal{U} \cup \{u_i\}$
    Delete from $\mathcal{P}$ all points covered by u_i.
    set $i = i + 1$.
  end if
end while
Output $\mathcal{U}$.

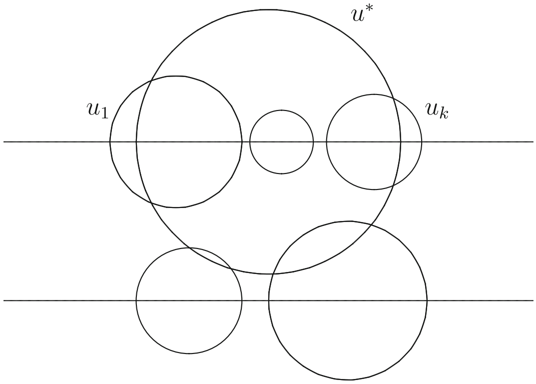


Fig. 3 An illustration for the case $|\mathcal{U}^*| = 1$ and $\mathcal{L}^* = \emptyset$

covered by $\mathcal{U}$ (if a mergence occurs such that the radius of the upper disk is enlarged to be larger than half of the width of the strip).

Theorem 6 *Algorithm 5 computes in linear time an approximation solution to RPPAS problem for the case $\alpha = 1$ with approximation ratio $2\sqrt{2}$.*

Proof The time complexity is obvious. Next, we show the approximation ratio.

Let $\mathcal{U}^* \cup \mathcal{L}^*$ be an optimal solution, and $\mathcal{U} \cup \mathcal{L}$ be the output of Algorithm 5. We illustrate the proof step by step.

Step 1 First, consider the case that $|\mathcal{U}^*| = 1$ and $\mathcal{L}^* = \emptyset$. In this case, all points belong to u^*, where u^* is the only disk in $\mathcal{U}^*$. Let $u_1, \ldots, u_s$ be the upper disks in $\mathcal{U}$ and $l_1, \ldots, l_t$ be the lower disks in $\mathcal{L}$, ordered from left to right, respectively. In the following, we show that

$$Diam(\mathcal{U}) \leq \sqrt{2} Diam(u^*) \tag{7}$$

and

$$Diam(\mathcal{L}) \leq \sqrt{2} Diam(u^*). \tag{8}$$

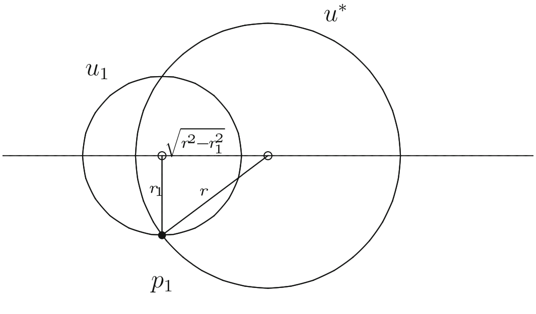


Fig. 4 The protruding length of u_1 is at most $r_1 - (r - \sqrt{r^2 - r_1^2})$

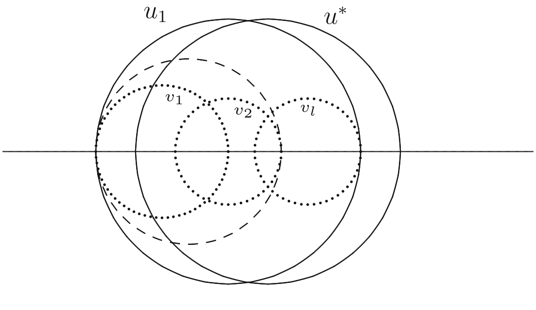


Fig. 5 The case when u_1 is obtained from merging $v_1, \ldots, v_l$

Since upper disks in $\mathcal{U}$ are mutually disjoint, we see that among all upper disks, only u_1 can protrude to the left of u^*, only u_k can protrude to the right of u^* (see Fig. 3). If u_1 is a disk without being merged, then it has p_1 as its lower extreme. Suppose the radii of u^* and u_1 are r and r_1 respectively. As shown in Fig. 4, the protruding length is at most $r_1 - (r - \sqrt{r^2 - r_1^2}) \triangleq f(r_1)$ and the maximum of $f(r_1)$ is reached when $r_1 = r/\sqrt{2}$, which yields $f_{\max} = (\sqrt{2} - 1)r$. If u_1 is a disk resulted from a sequence of merging disks $v_1, v_2, \ldots, v_l$, we see from Fig. 5 that the protruding length does not exceed that of v_1. Hence $(\sqrt{2} - 1)r$ is also the maximum protruding length of u_1 to the left of u^*. The same analysis holds for u_s by considering the last disk merged into u_s. As a consequence, $Diam(\mathcal{U}) \leq 2r + 2(\sqrt{2} - 1)r = 2\sqrt{2}r = \sqrt{2}Diam(u^*)$. Hence inequality (7) is proved.

Next, consider the diameter of $\mathcal{L}$. If there is no lower point, then the algorithm yields $\mathcal{L} = \emptyset$, and thus $Diam(\mathcal{L}) = 0$. Hence we suppose there exists at least one lower disk. As a consequence $r > a/2$, and all the lower points lie in the shaded part of Fig. 6. Project the leftmost lower point and the rightmost lower point onto the lower bank, suppose the projection points are b_l and b_r respectively (see Fig. 6). Then by a simple geometric analysis as shown in Fig. 7, we see that the distance between b_l and b_r is at most $2\sqrt{r^2 - a^2/4}$. Next, consider the protruding part of l_1 to the left of b_l. If l_1 is a disk without being merged, then it has the leftmost lower point as its upper extreme. Since a lower point is at most $a/2$ away from the lower bank, we see that l_1 protrudes to the left of b_l by at most $a/2$. If l_1 is a disk obtained by a sequence of mergence, then by a same argument as for the upper disks, we see that l_1 still protrudes to the left of b_l by at most $a/2$. Similarly, l_t protrudes to the right of b_r by at most $a/2$. Hence

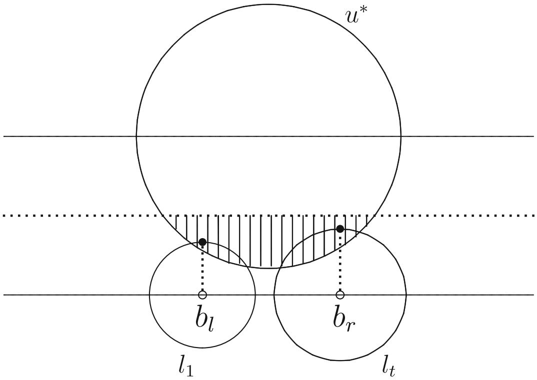


Fig. 6 All lower points lie in the shaded part

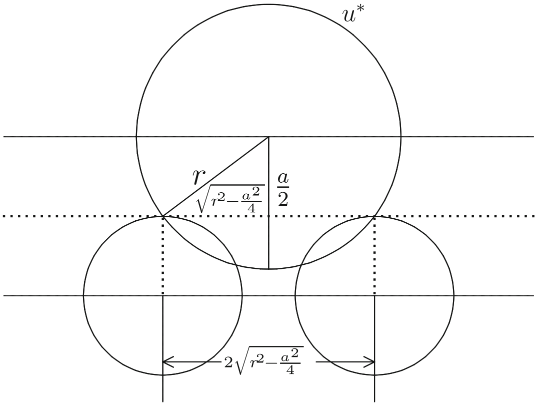


Fig. 7 The projection distance between the left most lower point and the rightmost lower point is at most $2\sqrt{r^2 - \frac{a^2}{4}}$

$$Diam(\mathcal{L}) \le 2\sqrt{r^2 - a^2/4} + a = 2r\left(\sqrt{1 - \left(\frac{a}{2r}\right)^2} + \frac{a}{2r}\right). \tag{9}$$

Recall that $0 < a/(2r) < 1$, we see from inequality (9) that $Diam(\mathcal{L}) \le 2\sqrt{2}r = \sqrt{2}Diam(u^*)$, with equality when $a/(2r) = 1/\sqrt{2}$. Inequality (8) is proved.

Since disks in $\mathcal{U}$ are mutually disjoint and disks in $\mathcal{L}$ are mutually disjoint, we see from inequalities (7) and (8) that

$$\sum_{u\in\mathcal{U}} r(u) + \sum_{l\in\mathcal{L}} r(l) \le \frac{1}{2}(Diam(\mathcal{U}) + Diam(\mathcal{L})) \le \sqrt{2}Diam(U^*) = 2\sqrt{2}r(u^*).$$

The approximation ratio follows.

Step 2 Consider the case that $|\mathcal{U}^*| > 1$ and $\mathcal{L}^* = \emptyset$. Suppose $\mathcal{U}^* = \{u_1, \ldots, u^*_{s^*}\}$. The proof idea is the same as above. The difference here is illustrated in Fig. 8: some u_j may protrude to the right of some u^*_i by more than $(\sqrt{2} - 1)r^*_i$, if u_j is obtained by a sequence of merging disks $v_1, \ldots, v_l$. Such a case occurs only when $v_l \in u^*_{i+1}$. Let q be the last index such that v_q has its lower extreme in u^*_i. Then $q < l$ and $v_{q+1} \in u^*_{i+1}$. Let $u^{(1)}_j$ be the disk obtained from merging $v_1, \ldots, v_q$ and $u^{(2)}_j$ be the disk obtained from merging $v_{q+1}, \ldots, v_l$ (see Fig. 9). Then u_j can be obtained from merging $u^{(1)}_j$ and $u^{(2)}_j$, and $Diam(u_j) \le Diam(u^{(1)}_j) + Diam(u^{(2)}_j)$. Furthermore, $u^{(1)}_j$ protrudes to the right of u^*_i by at

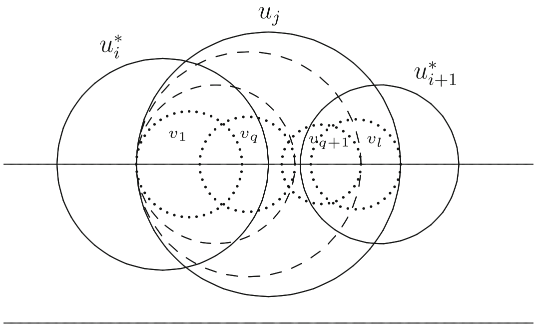


Fig. 8 Disk u_j may protrude to the right of u_i^* by more than $(\sqrt{2}-1)r_i^*$

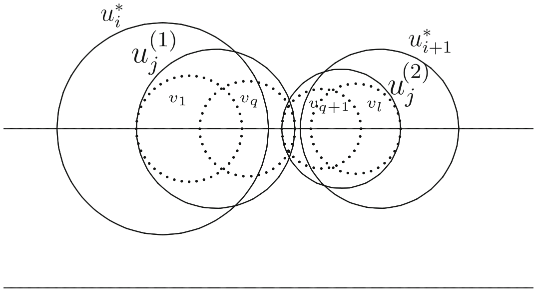


Fig. 9 Replacing disk u_j by two disks $u_j^{(1)}$ and $u_j^{(2)}$

most $(\sqrt{2}-1)r_i^*$ and $u_j^{(2)}$ protrudes to the left of u_{i+1}^* by at most $(\sqrt{2}-1)r_{i+1}^*$. Denote by $\widetilde{\mathcal{U}}$ the set of upper disks obtained from $\mathcal{U}$ by replacing all such u_j's by corresponding $u_j^{(1)}$'s and $u_j^{(2)}$'s. Then similarly to the first step, we see that for each u_i^*, the set of disks in $\widetilde{\mathcal{U}}$ has radii sum not more than $\sqrt{2}r(u_i^*)$. Notice that disks in $\widetilde{\mathcal{U}}$ may intersect only in their protruding part, we see that $\sum_{u\in\mathcal{U}} r(u) \le \sum_{\widetilde{u}\in\widetilde{\mathcal{U}}} r(\widetilde{u}) \le \sqrt{2}\sum_{u^*\in\mathcal{U}^*} r(u^*)$. Similar argument holds for $Diam(\mathcal{L})$. Then the approximation ratio follows similarly to that in Step 1.

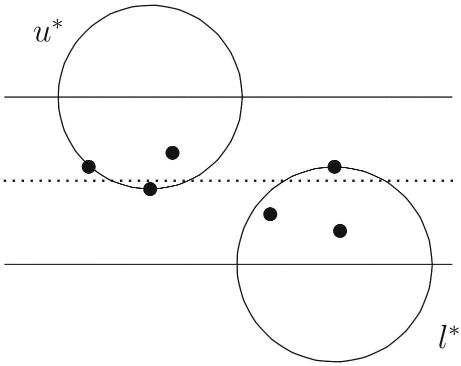
The case when $\mathcal{U}^* = \emptyset$ can be considered symmetrically.

Step 3 For the general case when both $\mathcal{U}^*$ and $\mathcal{L}^*$ are non-empty, we first illustrate the proof idea by considering $|\mathcal{U}^*| = |\mathcal{L}^*| = 1$. Let $\mathcal{U}^* = \{u^*\}$ and $\mathcal{L}^* = \{l^*\}$. If all the upper points are covered by u^* and all the lower points are covered by l^*, then similarly to the proof of (7), it can be shown that $Diam(\mathcal{U}) \le \sqrt{2}Diam(u^*)$ and $Diam(\mathcal{L}) \le \sqrt{2}Diam(l^*)$, and thus $\sum_{u\in\mathcal{U}} r(u) + \sum_{l\in\mathcal{L}} r(l) \le \sqrt{2}(r(u^*) + r(l^*))$.

In the following, we assume that some upper point is in $l^* \setminus u^*$ and some lower point is in $u^* \setminus l^*$ (see Fig. 10). In Fig. 11, we depict $\mathcal{U}$ and $\mathcal{L}$ which are the outputs of the algorithm applied to the set of upper points and the set of lower points, respectively. Let $\mathcal{P}_1$ be the set of points covered by u^* and let $\mathcal{P}_2$ be the set of points covered by l^*. Apply Algorithm 5 to $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. Suppose the outputs are $(\mathcal{U}_1, \mathcal{L}_1)$ and $(\mathcal{U}_2, \mathcal{L}_2)$, respectively (see Fig. 12). By the analysis in Step 1, we have $\sum_{u\in\mathcal{U}_1} r(u) + \sum_{l\in\mathcal{L}_1} r(l) \le 2\sqrt{2}r(u^*)$ and $\sum_{u\in\mathcal{U}_2} r(u) + \sum_{l\in\mathcal{L}_2} r(l) \le 2\sqrt{2}r(l^*)$. By noticing that $\mathcal{U}$ can be obtained by merging $\mathcal{U}_1$ with $\mathcal{U}_2$, without increasing the total power, we have $\sum_{u\in\mathcal{U}} r(u) \le \sum_{u\in\mathcal{U}_1} r(u) + \sum_{u\in\mathcal{U}_2} r(u)$. Symmetrically, $\sum_{l\in\mathcal{L}} r(l) \le \sum_{l\in\mathcal{L}_1} r(l) + \sum_{l\in\mathcal{L}_2} r(l)$. Hence $\sum_{u\in\mathcal{U}} r(u) + \sum_{l\in\mathcal{L}} r(l) \le 2\sqrt{2}(r(u^*) + r(l^*))$.

For the case when $|\mathcal{U}^*| > 1$ or $|\mathcal{L}^*| > 1$, the approximation ratio can be proved by combining the above analysis with the idea in Step 2. □

Fig. 10 Some upper point is in l^* and some lower point is in u^*



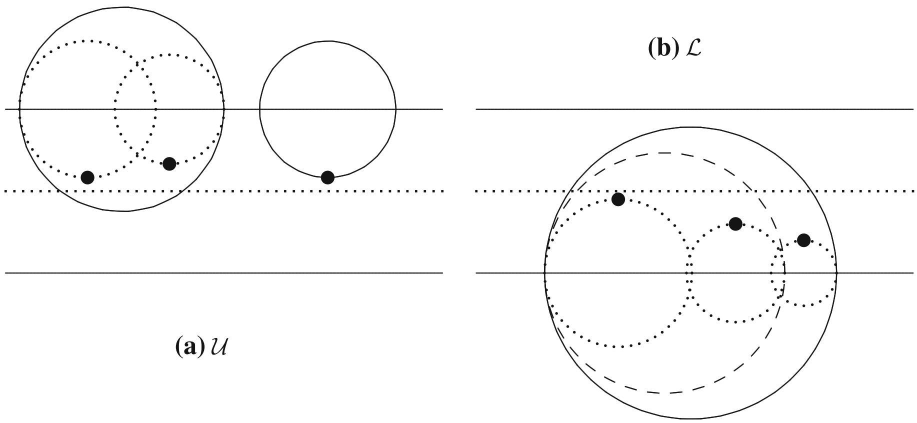


Fig. 11 $\mathcal{U}$ and $\mathcal{L}$ are the outputs of the algorithm applied to the set of upper points and the set of lower points, respectively

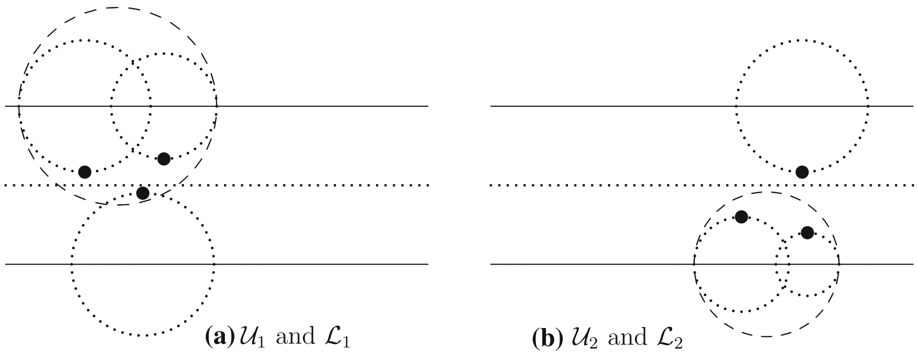


Fig. 12 The outputs of the algorithm applied to $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively

6 Conclusion

In this paper, we study the problem of monitoring crucial points in a river by radars with variable radii which are placed along the banks. Under the assumption that the river is sufficiently smooth, the focus is to solve the problem for a strip. An $O(n^9)$ dynamic programming algorithm is presented, where n is the number of crucial points to be monitored. We also presented an $O(kn^5)$ dynamic programming algorithm for the k-coverage radar placement problem in which radars are placed only along one bank. These ideas can be easily generalized to the discrete radar assignment problems. The problem of finding an exact polynomial algorithm to k-cover the points in which radars can be placed along the *two banks* is a topic for further research. For the case $\alpha = 1$, an $O(n \log n)$-time $2\sqrt{2}$-approximation algorithm is given. It should be noted that the key to this $2\sqrt{2}$-approximation algorithm is that 'mergence does not increase the total power, and thus in an optimal solution all disks are disjoint'. This property holds for any $\alpha \leq 1$. Hence the $2\sqrt{2}$-approximation algorithm is also valid for any $\alpha < 1$.

Acknowledgements This research is supported by NSFC (10971255), Program for New Century Excellent Talents in University (NCET-08-0921), and The Project-sponsored by SRF for ROCS, SEM

References

1. Agnetis, A., Grande, E., Mirchandani, P.B., Pacifici, A.: Covering a line segment with variable radius discs. Comput. Oper. Res. **36**, 1423–1436 (2009)
2. Alt, H., Arkin, E.M., Brönnimann, H., Erickson, J., Fekete, S.P., Knauer, C., Lenchner, J., Mitchell, J.S.B., Whittlesey, K.: Minimum-cost coverage of point sets by disks. In: Proceedings of the 22nd Annual ACM Symposium on Computational Geometry, pp. 449–458 (2006)
3. Baker, B.S.: Approximation algorithms for NP-complete problems on planar graphs. In: Proceedings of the 24th Annual IEEE Symposium on Foundations of Computer Science (1983), pp. 254–273. J. ACM **41**, 153–180 (1994)
4. Bilò, V., Caragiannis, I., Kaklamanis, C., Kanellopoulos, P.: Geometric clustering to minimize the sum of cluster sizes. In: Proceedings of the 13th Annual European Symposium on Algorithms, LNCS, vol. 3669, pp. 60–71 (2005)
5. Cardei, I., Cardei, M., Wang, L.S., Xu, B.G., Du, D.-Z.: Optimal relay location for resource-limited energy-efficient wireless communication. J. Glob. Optim. **36**, 391–399 (2006)
6. Cheng, M.X., Gong, X.: Maximum lifetime coverage preserving scheduling algorithms in sensor networks. J. Glob. Optim. doi:10.1007/s10898-010-9636-3
7. Chvátal, V.: A combinatorial theorem in plane geometry. J. Comb. Theory **18**, 39–41 (1975)
8. Glaßer, C., Reith, S., Vollmer, H.: The complexity of base station positioning in cellular networks. Discrete Appl. Math. **148**, 1–12 (2005)
9. Gupta, H., Zhou, Z., Das, R., Gu, Q.: Connected sensor cover: self-organizaiton of sensor networks for efficient query execution. IEEE/ACM Trans. Netw. **16**, 55–67 (2006)
10. Hochbaum, D.S., Mass, W.: Approximation schemes for covering and packing problems in image processing and VLSI. J. ACM **32**, 130–136 (1985)
11. Hoffmann, F., Kaufmann, M., Kriegel, K.: The art gallery theorem for polygons with holes. In: IEEE 32nd Annual Symposium of Foundations of Computer Science, pp. 39–48 (1991)
12. Huang, C.F., Tseng, Y.C.: The coverage problem in a wireless sensor network. In: ACM International Workshop on Wireless Sensor Networks and Applications, pp. 115–121 (2003)
13. Lev-Tov, N., Peleg, D.: Polynomial time approximation shcemes for base station coverage with minimum total radii. Comput. Netw. **47**, 489–501 (2005)
14. Pardalos, P.M., Resende, M. (eds.): Handbook of Applied Optimization. Oxford University Press, Oxford (2002)
15. Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., Gill, C.: Integrated coverage and connectivity configuration in wireless sensor networks. In: ACM International Conference on Embedded Networked Sensor Systems, pp. 28–39 (2003)