

# Maximising Lifetime for Fault-Tolerant Target Coverage in Sensor Networks

Thomas Erlebach  
Dept. of Computer Science,  
University of Leicester,  
England  
t.erlebach@mcs.le.ac.uk

Tom Grant  
Dept. of Computer Science,  
University of Leicester,  
England  
tg53@mcs.le.ac.uk

Frank Kammer  
Institut für Informatik,  
Universität Augsburg,  
Germany  
kammer@informatik.uni-augsburg.de

## ABSTRACT

We study the problem of maximising the lifetime of a sensor network for fault-tolerant target coverage in a setting with composite events. Here, a composite event is the simultaneous occurrence of a combination of atomic events, such as the detection of smoke and high temperature. We are given sensor nodes that have an initial battery level and can monitor certain event types, and a set of points at which composite events need to be detected. The points and sensor nodes are located in the Euclidean plane, and all nodes have the same sensing radius. The goal is to compute a longest activity schedule with the property that at any point in time, each event point is monitored by at least two active sensor nodes. We present a  $(6 + \varepsilon)$ -approximation algorithm for this problem by devising an approximation algorithm with the same ratio for the dual problem of minimising the weight of a fault-tolerant sensor cover and applying the Garg-Könemann algorithm. Our algorithm for the minimum-weight fault-tolerant sensor cover problem generalises previous approximation algorithms for geometric set cover with weighted unit disks and is obtained by enumerating properties of the optimal solution that guide a dynamic programming approach.

## Categories and Subject Descriptors

F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems—*Geometrical problems and computations*

## General Terms

Algorithms, Theory

## Keywords

Approximation algorithm, unit disk graph, set multi-cover, dynamic programming

## 1. INTRODUCTION

Consider a sensor network whose task is to detect the occurrence of events at a given set of event points. Since sensors often have a limited battery supply, it is important to address the problem of maximising the lifetime of the network, i.e., the length of time during which the network can carry out its monitoring task successfully. The lifetime of the network can be prolonged by calculating an activity schedule in which only a subset of the sensor nodes is active at any point in time, and the remaining sensors are in a sleep mode that saves energy. The active nodes must be sufficient for performing the required monitoring task. Following [17, 14], we consider the setting where the events to be detected are *composite events*, i.e., events comprised of several simultaneous *atomic events* at the same location detected by different sensor types (e.g., detecting a fire by observing the atomic events of high temperature and smoke), and the sensor coverage is required to be *fault-tolerant*, i.e., the failure of any one sensor does not affect the sensing task. We assume that the sensor nodes and event points are located in the Euclidean plane, and all sensor nodes have the same sensing radius. Each sensor node can monitor a certain set of event types, and the composite event to be detected at each event point is a combination of atomic events corresponding to different event types.

A common approach to lifetime maximisation is to formulate the problem as a linear program and obtain an approximate solution by approximating the dual problem of computing a sensor cover of minimum weight (see, e.g., [2, 3, 8]). We follow the same approach and hence mainly consider the dual problem of minimising the weight of a fault-tolerant sensor cover. We model the latter problem as a weighted multi-cover problem with unit disks and a set of event types.

In the special case of atomic events of just one event type and no fault-tolerance requirements, the minimum weight sensor cover problem is a standard geometric set cover problem where the aim is to cover a given set of points using unit disks of minimum total weight. The best known approximation ratio for that problem is  $4 + \varepsilon$  [9, 20]. Our setting poses the additional challenges of having to cover every point twice (turning the problem into a *multi-cover* problem) while avoiding the loss of a factor of two in the approximation ratio, and of dealing with different event types and composite events. Addressing these challenges requires us to refine the techniques that have been developed for the standard geometric set cover problem with unit disks.

## 1.1 Related Work

Sensor cover problems have been studied in several variants, including target coverage problems where a discrete set of points that need to be monitored is specified in the input, and region coverage problems where the area to be monitored is specified as a (typically convex) region in the plane. We refer to the survey by Thai et al. [16] for an overview. Berman et al. [2, 3] show that the region coverage problem can be reduced to the target coverage problem and present an algorithm with logarithmic approximation ratio. They also show that a minimum cost sensor cover algorithm with approximation ratio  $\rho$  implies an approximation algorithm with ratio  $\rho(1 + \varepsilon)$  for the lifetime maximisation problem using the Garg-Könemann algorithm [11]. Zhao and Gurusamy [18] study the target coverage problem with the additional requirement that the sensors that are active at any time are connected. They obtain an algorithm with logarithmic approximation ratio and also present a performance evaluation based on simulation experiments. Sanders and Schieferdecker [15] show that the target coverage problem for sensors represented by unit disks with the objective of lifetime maximisation is  $\mathcal{NP}$ -hard. They provide a  $(1 + \varepsilon)$ -approximation algorithm using resource augmentation, i.e., their algorithm needs to increase the sensing range of every sensor node by a factor of  $1 + \delta$ , for some fixed  $\delta > 0$ . Vu et al. [17] and Marta et al. [14] consider fault-tolerant sensor cover problems with composite events. They present centralised and distributed heuristics and evaluate them in simulations. In this paper, we aim at designing approximation algorithms with provable performance guarantees for fault-tolerant sensor cover problems with composite events.

A special case of the minimum cost sensor cover problem is the weighted geometric set cover problem with unit disks. This problem has received considerable attention as it includes the weighted dominating set problem for unit disk graphs, which is relevant for routing backbone construction in wireless networks. This relationship also shows that the problem is  $\mathcal{NP}$ -hard [6]. A series of papers has presented approximation algorithms with smaller and smaller constant approximation ratios for weighted set cover with unit disks [1, 13, 7, 9, 20]. The currently best known ratio is  $4 + \varepsilon$  [9, 20]. These results apply also to the minimum-weight dominating set problem in unit disk graphs. If one is interested in a minimum-weight *connected* dominating set, a standard approach is to first compute a cheap dominating set and then solve a node-weighted Steiner tree problem to connect it [1, 7, 9, 13, 20]. The node-weighted Steiner tree problem admits a  $2.5\alpha$ -approximation algorithm in unit disk graphs [10, 19], where  $\alpha$  is the approximation ratio of the best known approximation algorithm for edge-weighted Steiner trees. Since  $\alpha < 1.39$  [4], this gives approximation ratio less than 3.475. Together with the  $(4 + \varepsilon)$ -approximation from [9, 20], this yields a 7.475-approximation algorithm for minimum-weight connected dominating sets in unit disk graphs.

The unweighted set multi-cover problem has been studied in geometric settings by Chekuri et al. [5]. They present an  $O(\log \text{OPT})$ -approximation algorithm for set systems of bounded VC dimension, where  $\text{OPT}$  is the size of an optimal cover, and constant-factor approximation algorithms for covering points by half-spaces in three dimensions or for covering points with pseudo-disks in the Euclidean plane. Their results only apply to the unweighted case.

## 1.2 Our Results

We model the fault-tolerant target coverage problem with composite events as a generalised geometric multi-cover problem with unit disks and present a  $(6 + \varepsilon)$ -approximation algorithm, both for the lifetime maximisation variant and for the minimum cost sensor cover variant of the problem. On a high level, we solve the minimum cost sensor cover problem by providing a 6-approximation algorithm for the case where all event points are located in a square of bounded size (which we refer to as a *block*) and employing the geometric shifting strategy [12, 13]. To obtain the 6-approximation algorithm for a block, we ‘guess’ a number of properties of an optimal solution by enumeration, and then apply dynamic programming along horizontal and vertical strips of smaller squares. Because of the results of the ‘guessing’ step, we only need to handle the case where disks with centre outside a strip are used to cover points inside the strip, which makes a dynamic programming approach feasible. Our algorithm requires significant adaptations compared to previous work because of the multi-cover aspect and because there is more than one sensor type. Firstly, we need a considerably more involved ‘guessing’ step that allows us to classify the points in terms of the location of the disk which provides the covering. In particular, we need a non-trivial generalisation of the sandglass concept and we need to find a solution for the problem such that we do not count the same disk twice (once in a horizontal and once in a vertical strip) in order to ensure a feasible solution. In addition, we have to extend the dynamic programming algorithm such that we can find a 2-covering for the points and such that the algorithm can handle different event types. Using our approximation algorithm for minimum cost sensor cover as a subroutine in the Garg-Könemann algorithm [11], we also obtain approximation ratio  $6 + \varepsilon$  for lifetime maximisation. Furthermore, provided that the communication radius of a sensor node is at least twice its sensing radius, we can use the known approximation algorithm for node-weighted Steiner trees in unit disk graphs and obtain approximation ratio 9.475 for the problem variants where the sets of active sensors are required to form a connected communication network.

The remainder of the paper is structured as follows. Section 2 covers preliminaries. In Section 3, we present the details of our dynamic programming approach to solve the minimum cost fault-tolerant sensor cover problem in a block. In Section 4, we describe the enumeration procedure that ‘guesses’ the properties of a fixed optimal solution that the dynamic program requires as input. Section 5 concludes the paper.

## 2. PRELIMINARIES

Consider the two-dimensional Euclidean plane. The  $x$ -coordinate and  $y$ -coordinate of a point  $p$  is denoted by  $x_p$  and  $y_p$ , respectively. The Euclidean distance between two points  $p$  and  $q$  is denoted by  $\delta(p, q)$ . If  $d$  is a disk, we also use  $d$  to refer to the centre of  $d$ , so that we can write  $\delta(d, p)$  for the Euclidean distance between the centre of  $d$  and a point  $p$ . The power set of a set  $S$  is denoted by  $\mathcal{P}(S)$ . An algorithm for a maximisation, or minimisation, problem is a  $\rho$ -approximation algorithm if it runs in polynomial time and always outputs a solution with objective value at least  $\text{OPT}/\rho$ , or at most  $\rho \cdot \text{OPT}$ , where  $\text{OPT}$  denotes the objective value of an optimal solution.

## 2.1 Problem Definitions

Let  $T$  be a (constant-size) set of different event types (e.g., smoke, temperature, etc.). Consider a set  $P$  of points in the two-dimensional Euclidean plane and a set  $D$  of weighted unit disks. All disks have the same radius  $r$ , and without loss of generality we assume  $r = 2$  throughout this paper. In the context of the target coverage problem, the disks in  $D$  correspond to sensor nodes (with  $r$  representing the sensing radius) and the points in  $P$  correspond to targets (event points) that need to be monitored. The weight of a disk  $d \in D$  is non-negative and denoted by  $w(d)$  or  $w_d$ . The total weight of a set  $D' \subseteq D$  of disks is denoted by  $w(D') = \sum_{d \in D'} w(d)$ . Every disk (sensor)  $d \in D$  has sensing components for a subset  $T_d \subseteq T$  of event types. Every target  $p \in P$  specifies a positive integer  $k_p$  as its coverage requirement and an event type  $t_p \in T$  that needs to be monitored at  $p$ . (A target  $p$  at which composite events composed of the event types in some  $T' \subseteq T$  need to be monitored is represented by  $|T'|$  copies of  $p$ , with each copy  $p'$  associated with a different event type  $t_{p'} \in T'$ .)

We say that a disk  $d \in D$  covers a point  $p \in P$  if  $p$  is in  $d$  and  $t_p \in T_d$ . A set  $D' \subseteq D$  of disks meets the coverage requirements of a point  $p \in P$  if  $p$  is covered by at least  $k_p$  distinct disks in  $D'$ , and  $D'$  is a feasible multi- $T$ -cover if it meets the coverage requirements of all points  $p \in P$ . The objective of the weighted multi- $T$ -cover problem with unit disks, denoted by WMCUD- $T$ , is to compute a feasible multi- $T$ -cover of minimum total weight. We mostly assume that  $k_p \leq 2$  for all  $p \in P$  and refer to this restriction of WMCUD- $T$  as W2CUD- $T$ .

The lifetime maximisation variant is defined as follows: We are given points and disks as in an instance of WM-CUD- $T$ , but additionally each disk  $d \in D$  specifies an initial battery level  $b_d$ , expressed in suitable units so that  $b_d$  is the total duration during which  $d$  can be active before its battery runs out. A schedule is a set of pairs  $(D_i, x_i)$ , where  $D_i \subseteq D$  is a feasible multi- $T$ -cover and  $x_i \geq 0$ . A schedule is feasible if for each  $d \in D$ , the sum of the  $x_i$  values of all pairs  $(D_i, x_i)$  with  $d \in D_i$  does not exceed  $b_d$ . The lifetime of a schedule is the sum of the  $x_i$  values of all its pairs  $(D_i, x_i)$ . The goal is to compute a feasible schedule of maximum lifetime. We refer to this problem as the maximum lifetime multi- $T$ -cover problem with unit disks (MLMCUD- $T$ ), and the restricted version where  $k_p \leq 2$  for all  $p \in P$  as ML2CUD- $T$ .

## 2.2 Lifetime Maximisation and the Algorithm by Garg and Könemann

A linear program  $\Pi$  of the form  $\{\max c^T x \mid Ax \leq b, x \geq 0\}$ , where  $A, b$  and  $c$  are non-negative, is called a *packing problem*. The linear program may be given implicitly, and the number of variables  $x_j$  may be exponential. For a given vector  $w$ , the problem of finding a column  $j$  of  $A$  such that  $\sum_i A_{i,j} w_i / c_j$  is minimised is called the problem of *computing a column of minimum length* with respect to  $\Pi$ . It is known [2] that, if a packing problem  $\Pi'$  admits a  $\rho$ -approximation algorithm for the problem of computing a column of minimum length with respect to  $\Pi'$  for any given vector  $w$ , then the algorithm by Garg and Könemann [11] can be used to compute a  $(1 + \varepsilon)\rho$ -approximate solution to  $\Pi'$ .

The lifetime maximisation problem can be written as a linear program with a variable  $x_{D'}$  for every feasible sensor cover  $D' \subseteq D$ , representing the length of the part of the

schedule during which the set of active sensors is  $D'$ . A linear constraint for every node ensures that the total time during which the node is active does not exceed the battery life of that node. The linear program does not have polynomial size, as the number of variables  $x_{D'}$  can be exponential, but it is a packing problem, and the algorithm by Garg and Könemann [11] can be applied. The problem of computing a column of minimum length is simply the problem of computing a feasible sensor cover  $D'$  of minimum cost, where the cost of a node  $d \in D$  is given by some weight  $w_d$ . In our case, the latter problem is W2CUD- $T$ . As we present a  $(6 + \varepsilon)$ -approximation algorithm for W2CUD- $T$  in the remainder of the paper (Theorem 1), we obtain that, for every fixed  $\varepsilon > 0$ , there is a  $(6 + \varepsilon)$ -approximation algorithm for ML2CUD- $T$ .

## 2.3 Plane Partition

As in previous work (e.g., [13]), our algorithms employ a partition of the plane. Imagine an infinite grid that partitions the plane into squares of side length 1.4 (any number sufficiently close to, but strictly less than,  $\sqrt{2}$  would do). Consider an arbitrary such square  $S_{ij}$ . Note that any disk of radius 2 with centre in  $S_{ij}$  contains the whole square. Let  $P_{ij} \subseteq P$  be the set of points from  $P$  that lie in  $S_{ij}$ . We can assume without loss of generality that no point or disk centre lies exactly on the boundary between two adjacent squares. The neighbouring infinite regions of a square  $S_{ij}$  are referenced as shown in Figure 1, with UL standing for ‘upper left,’ CR for ‘centre right,’ LM for ‘lower middle’, etc. Furthermore, let UPPER be the union of the regions UL, UM, UR, let LOWER be the union of LL, LM, LR, let LEFT be the union of UL, CL, LL, and let RIGHT be the union of UR, CR, LR.

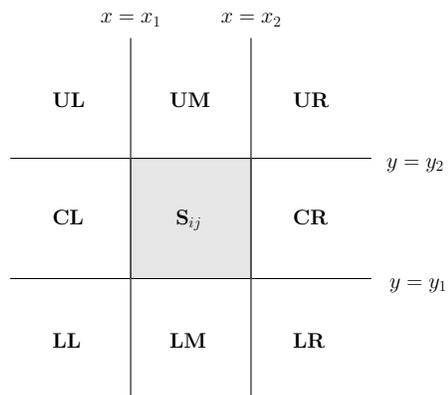


Figure 1: Square  $S_{ij}$  and neighbouring regions

For an integer constant  $K > 0$  (which determines the  $\varepsilon$  term in the final approximation ratio), consider a partition of the plane into *blocks* so that each block  $B$  consists of  $K \times K$  squares  $S_{ij}$ . If we have a  $\rho$ -approximation for W2CUD- $T$  instances whose points lie in one block, we can obtain a  $\rho(1 + O(1/K))$ -approximation for general instances of W2CUD- $T$  using the standard geometric shifting strategy [12]. (We omit further details of this process.) Consequently, the key to obtaining a good approximation algorithm for W2CUD- $T$  is to achieve a good approximation ratio for instances where the points are located in one block.

### 3. 6-APPROXIMATION FOR W2CUD-T IN A BLOCK

Our approach to solve W2CUD- $T$  in a  $K \times K$  block consists of two stages: In the first stage, using enumeration we ‘guess’ properties of a fixed optimal solution. (Here and in the following, any reference to ‘the optimal solution’ refers to that fixed optimal solution.) In the second stage, we approximate the best solution with these properties using dynamic programming. We defer the details of the enumeration stage to Section 4. The outcome of the enumeration stage is that we can assume that we know for every point  $p$  in a square  $S_{ij}$  how often (zero times, once, or at least twice) it is to be covered by disks with centre in  $S_{ij}$  (and what up to two of those disks are), how often by disks with centre in UPPER or LOWER, and how often by disks with centre in LEFT or RIGHT. Moreover, for points that are to be covered once by a disk in UPPER or LOWER and once by a disk in LEFT or RIGHT, for one of the two cases we know that the point is to be covered by a disk from  $UM \cup LM$ , or by a disk from  $CL \cup CR$ . (This ensures that the two disks computed by the two different dynamic programs dealing with the point are distinct.) Dynamic programming is applied to each horizontal and each vertical strip of squares contained in the block  $B$ . In the following, we describe the dynamic program for a horizontal strip. Vertical strips are dealt with analogously.

#### 3.1 Strip problem

Consider a horizontal strip  $H$  of squares, consisting of  $K$  squares  $S_{ij}$ . We are given a set  $P_H$  of points in the strip, and a set  $D_{\bar{H}}$  of disks with centre above or below the strip (i.e., all the disks have centres in the union of the regions  $UPPER \cup LOWER$  for all squares  $S_{ij}$  in the strip  $H$ ). Each disk  $d \in D_{\bar{H}}$  is associated with a weight  $w(d)$  and a set  $T_d$  of event types. Each point  $p \in P$  has an event type  $t_p$  and a coverage requirement  $k_p \in \{1, 2\}$ . If  $k_p = 1$ , the point may additionally specify that it must be covered by a disk with centre in  $UM \cup LM$  (as opposed to being covered by any disk from  $UPPER \cup LOWER$ ). We let  $n_H$  denote the number of points in  $P_H$ . Let the points in  $P_H = \{p_1, p_2, \dots, p_{n_H}\}$  be ordered by non-decreasing  $x$ -coordinates, breaking ties arbitrarily.

A set  $D' \subseteq D_{\bar{H}}$  of disks *meets the coverage requirement* of  $p \in P_H$  if the following holds: If  $k_p = 2$ , then  $D'$  contains two distinct disks that cover  $p$ . If  $k_p = 1$ , then  $D'$  contains a disk that covers  $p$ , and if  $p$  requires to be covered by a disk from  $UM \cup LM$ , then  $D'$  contains a disk that covers  $p$  and lies in that pair of regions. (Note that regions are specified with respect to the square  $S_{ij}$  that contains  $p$ .) It is easy to detect infeasible instances, so we only consider the case that there is a feasible solution, i.e., a set of disks  $D' \subseteq D_{\bar{H}}$  that meets the coverage requirements of all points in  $P_H$ . The goal is to compute a feasible solution of minimum weight. We refer to this problem as the *strip problem*.

#### 3.2 Outer and Inner Envelopes

For every  $T' \in \mathcal{P}(T) \setminus \{\emptyset\}$ , i.e., for every non-empty combination of event types in  $T$ , we consider twelve *envelopes*. Consider an arbitrary set  $D_{T'}$  of disks that intersect a square  $S_{ij}$  and monitor exactly all the event types in  $T'$ , i.e.,  $T_d = T'$  for all  $d \in D_{T'}$ . We consider an *outer  $T'$  envelope* and an *inner  $T'$  envelope* (with respect to  $D_{T'}$ ) for each of the regions UL, UM, UR, LL, LM and LR. The purpose of

envelopes is to represent the disks lying in that region that cover points in  $P_H$  in a solution, in the sense that any point in  $P_H$  that is covered once or twice by disks from that region is also covered at least the same number of times by disks that are on the two envelopes of that region. The algorithm then aims at computing envelopes corresponding to a solution of minimum cost.

We next define the outer and inner  $T'$  envelope for the region UM (with respect to  $D_{T'}$ ). (Envelopes for the remaining five regions can be defined analogously.) The *outer  $T'$  envelope* for the region UM represents the boundary of the union of all disks in  $D_{T'}$  that have centre in the region UM. More specifically, the envelope is the segment of the boundary that lies in the square  $S_{ij}$ . (If at some  $x$ -coordinate there is no disk from UM that overlaps  $S_{ij}$ , we let the upper boundary of the square form a part of the envelope.) The *inner  $T'$  envelope* for UM is the envelope formed by the set of disks in  $D_{T'}$  with centre in UM that remain after discarding the set of disks that form the outer  $T'$  envelope for UM. We view the set of disks in an envelope as ordered by non-decreasing  $x$  coordinates of their centres. Note that this is the order in which the disks appear on the envelope if we trace the envelope from left to right.

#### 3.3 Dynamic Programming

We create a table  $W_{p_i}$  for every point  $p_i \in P_H$ . For every  $T' \in \mathcal{P}(T) \setminus \{\emptyset\}$ , we have the following indexes for the table  $W_{p_i}$ : For each of the six regions UL, UM, UR, LL, LM and LR, we have a set of up to three disks that are candidates for the disk  $d$  that is on the outer  $T'$  envelope of that region at position  $x = x_{p_i}$ , for the disk just before  $d$  on that envelope, and for the disk just after  $d$  on that envelope. For the inner  $T'$  envelope of each of the six regions, we have one disk that is a candidate for being the disk on that envelope at position  $x = x_{p_i}$ . Hence, an entry of the table  $W_{p_i}$  is indexed by  $24 \cdot (2^{|T'} - 1)$  disks (three disks for each of the six outer envelopes, and one disk for each of the six inner envelopes, for each choice  $T' \subseteq T$ ). For ease of presentation, we write the indexes for the table  $W_{p_i}$  as two sets of disks  $D_U$  and  $D_L$ , where  $D_U$  contains all the disks from inner and outer  $T'$  envelopes for any  $T'$  and regions UL, UM and UR, and  $D_L$  contains all the disks from inner and outer  $T'$  envelopes for any  $T'$  and regions LL, LM and LR.

Consider the case that the indexes for the table  $W_{p_i}$  for each  $T'$  are chosen as the disks that actually form the envelopes under consideration in the optimal solution. We can observe that, if  $p_i$  is covered once or twice by the optimal solution, then it is also covered once or twice, respectively, by the disks constituting the indexes for the table.

The value of an entry of table  $W_{p_i}$  is infinity if the disks indexing the table entry do not meet the coverage requirement for  $p_i$ , and otherwise the minimum cost of a set of disks that includes all the disks indexing the table entry of  $W_{p_i}$  and that also meets the coverage requirements of all points preceding  $p_i$  in  $P_H$ . Once all the tables  $W_{p_i}$  have been computed, the set of disks corresponding to the minimum value of any entry of  $W_{p_{n_H}}$  is output as the solution.

The table entries for the leftmost point  $p_1 \in P_H$  are initialised as follows. For every choice of indexes  $D_U$  and  $D_L$ , the table entry  $W_{p_1}(D_U, D_L)$  is set to  $w(D_U) + w(D_L)$  if  $D_U \cup D_L$  meets the coverage requirement of point  $p_1$ , and to  $\infty$  otherwise. For subsequent points  $p_i \in P_H$ , the value of an entry of  $W_{p_i}$  is calculated as the cost for the set of disks

specified as indexes of the entry plus the cost of a cheapest set of disks covering all points up to  $p_{i-1}$ , which can be found in the table  $W_{p_{i-1}}$ , while ensuring that the costs of disks contained in both sets are counted only once:

$$W_{p_i}(D_u, D_l) = \begin{cases} \infty, & \text{if } D_u \cup D_l \text{ does not meet} \\ & \text{the coverage requirement for } p_i \\ \min_{D'_u, D'_l} \left\{ \begin{array}{l} W_{p_{i-1}}(D'_u, D'_l) + \\ w(D_u - D'_u) + w(D_l - D'_l) \end{array} \right\}, & \text{otherwise} \end{cases}$$

Consider the last point  $p_{n_H} \in P_H$ . The minimum value in the table  $W_{p_{n_H}}$  is the cost of the minimum weight solution that covers all points in  $P_H$ . The proof of correctness is omitted due to space limitations.

### 3.4 Solving W2CUD- $T$ in a Block

Consider an instance of W2CUD- $T$  in a block  $B$  of  $K \times K$  squares, and assume that we know from the enumeration stage how often each point is to be covered by disks from different regions. We apply the dynamic program to the strip problem for each of the  $K$  horizontal strips in  $B$  and (by rotating the plane by  $90^\circ$ ) each of the  $K$  vertical strips. The union of the  $2K$  solutions together with the set of disks that has been determined to be in the solution by the enumeration stage is then output as the solution for the entire block  $B$ . This gives a 6-approximation algorithm for instances of W2CUD- $T$  where all points lie in one  $K \times K$  block. (Details are deferred to the full version.) As remarked in Section 2, this implies the following.

**THEOREM 1.** *For every fixed  $\varepsilon > 0$ , there is a  $(6 + \varepsilon)$ -approximation algorithm for W2CUD- $T$ .*

### 3.5 Connected Sensor Cover

Up to now we have considered only the condition that the selected sensors meet the coverage requirement of each point in  $P$ . In many applications, such as the settings described in [14, 17], it is additionally required that the selected sensors form a connected network. For this, it is assumed that each sensor node is equipped with a wireless radio that allows it to transmit messages to any other node that is located within a certain communication radius  $r_c$  from it. (This corresponds to a communication graph where the sensor nodes are represented by disks of radius  $r_c/2$  and two nodes are adjacent if their disks intersect.) It is natural to expect that  $r_c$  is larger than the sensing radius  $r$ . Under the assumption that  $r_c \geq 2r$ , we can extend our approximation algorithms for W2CUD- $T$  and ML2CUD- $T$  to the problem variants with connectivity requirement.

For W2CUD- $T$  with connectivity requirement, we first compute a  $(6 + \varepsilon)$ -approximate solution  $D'$  to the problem without the connectivity requirement, by using the algorithm from Theorem 1. Then, viewing the given disks as disks of radius  $r_c/2$ , we solve the minimum node-weighted Steiner tree problem in the corresponding unit-disk graph for the disks in  $D'$  as terminals, using the algorithm with approximation ratio less than 3.475 for node-weighted Steiner trees in unit disk graphs [10, 19, 4]. Let  $S$  be the set of Steiner nodes output by the algorithm. The set  $D' \cup S$  is then output as a solution to W2CUD- $T$  with connectivity

requirement. Let  $\text{OPT}_c$  be an optimal solution to W2CUD- $T$  with connectivity requirement. Observe that  $\text{OPT}_c$  is a (superset of a) feasible solution to the Steiner tree problem considered above:  $\text{OPT}_c$  is connected and contains disks covering every point in  $P$ . Every disk in  $D'$  covers a point in  $P$ , and hence the centre of any disk in  $D'$  is within distance  $r + r \leq r_c$  of the centre of some disk in  $\text{OPT}_c$ . Consequently,  $\text{OPT}_c \cup D'$  is connected. This shows that the Steiner tree approximation algorithm produces a set  $S$  of cost less than 3.475 times the cost of  $\text{OPT}_c$ . As the cost of  $D'$  is within a factor of  $6 + \varepsilon$  of the optimal solution to W2CUD- $T$  without connectivity requirement, and thus within the same factor of the cost of  $\text{OPT}_c$ , the overall approximation ratio is bounded by 9.475 if  $\varepsilon$  is chosen sufficiently small.

**THEOREM 2.** *There is a 9.475-approximation algorithm for the variants of W2CUD- $T$  and ML2CUD- $T$  where the active disks need to be connected and  $r_c \geq 2r$ .*

## 4. GUESSING PROPERTIES OF THE OPTIMAL SOLUTION BY ENUMERATION

In this section, we describe how the knowledge of properties of the optimal solution that is required by the dynamic programming algorithm in Section 3 can be obtained using enumeration techniques. Fix an arbitrary optimal solution  $\text{OPT}_B$  to the given instance of W2CUD- $T$  in a block  $B$ . We present the enumeration technique using the notion of ‘guessing’. When we write that the algorithm ‘guesses’ a property of  $\text{OPT}_B$ , this means that the algorithm enumerates all possibilities for that property in such a way that one of the possibilities is guaranteed to be the desired property of the optimal solution. The enumeration is done for each of the  $K^2$  squares  $S_{ij}$  contained in  $B$ . Consider one such square  $S_{ij}$ . Let the left and right boundary of  $S_{ij}$  lie on the line  $x = x_1$  and  $x = x_2$ , respectively, and the bottom and top boundary on the line  $y = y_1$  and  $y = y_2$ , respectively (cf. Figure 1).

First, for every event type  $t_l$  in  $T$ , we guess whether  $\text{OPT}_B$  contains 0, 1 or at least 2 disks with centre in  $S_{ij}$  that monitor event type  $t_l$ . Furthermore, in the second and third case we also guess the one disk or two of those disks, respectively. Note that a disk with centre in  $S_{ij}$  must contain the whole square  $S_{ij}$ , as the disk has radius  $r = 2$  and the square has side length less than  $\sqrt{2}$ . Furthermore, two disks with centre in  $S_{ij}$  that cover event type  $t_l$  are sufficient to meet the coverage requirements of all points  $p \in P_{ij}$  with  $t_p = t_l$ , so it is not necessary to guess more than two such disks.

Next, we aim at guessing a partition of the square  $S_{ij}$  into areas such that for the points  $p$  in the same area, we know whether they are covered by  $\text{UPPER} \cup \text{LOWER}$ , by  $\text{LEFT} \cup \text{RIGHT}$ , or once by  $\text{UPPER} \cup \text{LOWER}$  (possibly restricted to  $\text{UM} \cup \text{LM}$ ) and once by  $\text{LEFT} \cup \text{RIGHT}$  (possibly restricted to  $\text{CL} \cup \text{CR}$ ). Moreover, we guess a separate such partition for each event type  $t_l \in T$ . For each  $t_l \in T$ , the steps involved in guessing the partition are as follows. First, we determine areas called *2-watching  $t_l$  sandglasses* (the terminology is motivated by a similar sandglass concept in [13]) for the four regions  $\text{UM}$ ,  $\text{LM}$ ,  $\text{CL}$  and  $\text{CR}$ . For each of these areas, we can require that points are covered only by  $\text{UPPER} \cup \text{LOWER}$ , or only by  $\text{LEFT} \cup \text{RIGHT}$ . Second, we consider 1-watching envelopes, i.e., envelopes of the disks in  $\text{OPT}_B$  with centre in one of the regions  $\text{UM}$ ,  $\text{LM}$ ,  $\text{CL}$  and  $\text{CR}$ , and guess the four points where adjacent envelopes intersect. Based on the

locations of these intersection points, we can segment the square into smaller areas and deduce for each of the smaller areas whether the points located in the area are to be covered from  $\text{UPPER} \cup \text{LOWER}$  or from  $\text{LEFT} \cup \text{RIGHT}$  (or from both). The details of the partition of the square into areas for one specific event type  $t_l$  are presented in the following.

## 4.1 2-Watching Sandglasses

We define the 2-watching  $t_l$  sandglass for the region LM. The sandglasses for the regions UM, CR and CL are defined similarly (by rotation). Let  $P'_{ij}$  be the set of points in  $P_{ij}$  that have event type  $t_l$ , i.e., the set of points  $p$  with  $t_p = t_l$ . Consider the set  $P^2_{LM} \subseteq P'_{ij}$  of all points in  $P'_{ij}$  that are covered by two distinct disks from LM in  $\text{OPT}_B$ , but not covered by any disk in  $\text{OPT}_B$  that does not have centre in LM. For each  $p \in P^2_{LM}$ , consider the line  $l_p$  through  $p$  with slope 1, and let  $p'$  be the point where  $l_p$  intersects the line  $y = y_1$ . Let  $p_l$  be the point in  $P^2_{LM}$  for which  $p'$  is leftmost. Similarly, let  $l'_p$  be the line through  $p$  with slope  $-1$ , and let  $p''$  be the intersection point of  $l'_p$  and  $y = y_1$ . Let  $p_r$  be the point in  $P^2_{LM}$  for which  $p''$  is rightmost. The 2-watching  $t_l$  sandglass for LM is now defined as the area that is obtained as the intersection of the halfplane below  $l_{p_l}$ , the halfplane below  $l'_{p_r}$ , and the square  $S_{ij}$ . See Figure 2 for an illustration.<sup>1</sup> Note that this sandglass is uniquely determined by  $p_l$  and  $p_r$  and there are  $O(|P_{ij}|^2)$  possibilities for guessing  $p_l$  and  $p_r$ .

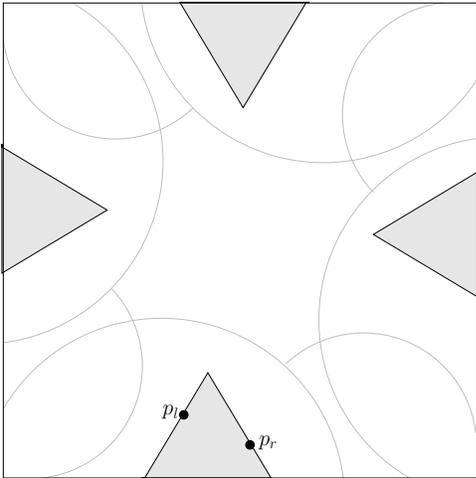


Figure 2: 2-watching sandglasses

We show that the coverage requirements of any point of  $P'_{ij}$  located in the 2-watching  $t_l$  sandglass for LM are met by disks from  $\text{UPPER} \cup \text{LOWER}$ . Let the *lower-shadow* of a point  $p$  be the region that is the intersection of the halfplane below the line with slope 1 through  $p$ , the halfplane below the line with slope  $-1$  through  $p$ , and the square  $S_{ij}$ . The left-shadow, up-shadow, and right-shadow of a point are defined analogously.

We state the following lemma due to Huang et al. [13].

LEMMA 1. [13] *If a point  $p \in P'_{ij}$  is covered by a disk  $d$  from LM, then any point in the lower-shadow of  $p$  is also covered by the same disk from LM.*

<sup>1</sup>Figure 2 and subsequent figures are not drawn to scale as they serve only illustrative purposes.

The lemma directly implies the following corollary.

COROLLARY 1. *If a point  $p \in P'_{ij}$  is covered by two disks  $d_1, d_2$  from LM, then any point in the lower-shadow of  $p$  is also covered by the same two disks  $d_1, d_2$  from LM.*

We also require the following lemma, which we prove by application of Lemma 1.

LEMMA 2. *The coverage requirement for any point  $p \in P'_{ij}$  that lies inside the 2-watching  $t_l$  sandglass for LM is met by disks in  $\text{OPT}_B$  from  $\text{UPPER} \cup \text{LOWER}$ .*

PROOF. Consider the points  $p_l$  and  $p_r$  defining the 2-watching  $t_l$  sandglass for LM. For points in the lower-shadow of  $p_l$  or in the lower-shadow of  $p_r$ , the lemma follows from Corollary 1. Let  $p$  be a point that lies in the 2-watching  $t_l$  sandglass for LM, but not in the lower-shadows of  $p_l$  or  $p_r$ . Assume that  $p$  is covered by a disk  $d$  with centre in CL or CR by  $\text{OPT}_B$ . Then, by Lemma 1 applied to the left-shadow or right-shadow of  $p$ , respectively, we find that  $p_l$  or  $p_r$  is also covered by  $d$ , a contradiction to the choice of  $p_l$  and  $p_r$ .  $\square$

It follows that the coverage requirements of all points in the 2-watching  $t_l$  sandglasses for LM and for UM are satisfied by disks in  $\text{OPT}_B$  from  $\text{UPPER} \cup \text{LOWER}$ , and the coverage requirements of all points in the 2-watching  $t_l$  sandglasses for CL and for CR are satisfied by disks in  $\text{OPT}_B$  from  $\text{LEFT} \cup \text{RIGHT}$ . Hence, all the points from  $P'_{ij}$  that lie in 2-watching  $t_l$  sandglasses can be classified accordingly. These points are ignored for the classifications of points described in the following sections, i.e., their classification is not changed if they are contained in one of the areas under consideration there.

## 4.2 1-Watching Envelopes

It remains to deal with points from  $P'_{ij}$  that do not lie in one of the 2-watching  $t_l$  sandglasses. For each of the four regions UM, LM, CL and CR, we consider a *1-watching  $t_l$  envelope* that represents the boundary of disks in  $\text{OPT}_B$  that monitor event type  $t_l$  and have centre in that region. More precisely, the 1-watching  $t_l$  envelope is formed by the intersection of that boundary with the square  $S_{ij}$ . See Figure 3a, where the 1-watching UM and LM envelopes are drawn in bold, for an illustration. The boundary of the square is used to fill in parts of the envelope where no disk from the respective region intersects the square. We have separate envelopes for each  $t_l \in T$ .

We call the 1-watching envelopes of CL and UM *adjacent*, and similarly those of UM and CR, etc. Allow us to define *intersection points* of adjacent 1-watching envelopes and describe how they are used to partition the remainder of square  $S_{ij}$  into areas such that we can specify for the points in each region whether they are covered by  $\text{OPT}_B$  using disks in  $\text{UPPER} \cup \text{LOWER}$  or in  $\text{LEFT} \cup \text{RIGHT}$ .

LEMMA 3. *Adjacent 1-watching  $t_l$  envelopes intersect in exactly one intersection point.*

PROOF. By the dimension of the square  $S_{ij}$  and the radius of the disks, it follows that the direction of any tangent to the 1-watching  $t_l$  envelope for UM or LM is in the open interval  $(-\pi/4, \pi/4)$ , and the direction of any tangent to the 1-watching  $t_l$  envelope for CL or CR is in the open interval

$(\pi/4, 3\pi/4)$ . Therefore, it is impossible that two adjacent 1-watching  $t_l$  envelopes have more than one intersection point. As each envelope is a curve connecting points on opposite sides of the square, it follows that two adjacent envelopes always intersect.  $\square$

Hence, there are four intersection points between adjacent 1-watching  $t_l$  envelopes. Note that each of these intersection points is uniquely specified by the two disks whose boundaries intersect at that point. Therefore, it suffices to guess 8 disks in order to determine the four intersection points.

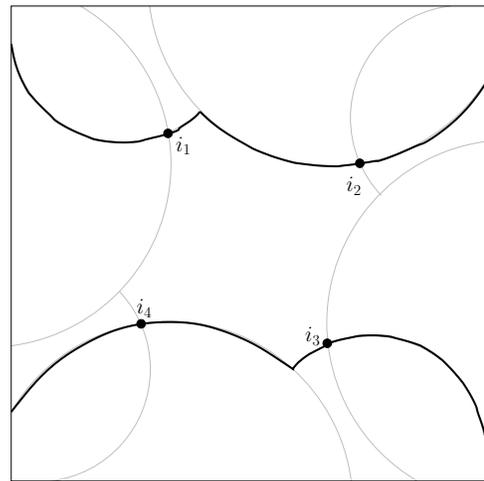
Let the intersection point of the 1-watching  $t_l$  envelopes for CL and UM be denoted by  $i_{CL}^{UM}$ . Similarly, let  $i_{CR}^{UM}$  be the intersection point of the 1-watching  $t_l$  envelopes of UM and CR,  $i_{CR}^{LM}$  the intersection point of the 1-watching  $t_l$  envelopes for CR and LM, and  $i_{CL}^{LM}$  the intersection point of the 1-watching  $t_l$  envelopes for LM and CL.

### 4.3 Areas in a Square

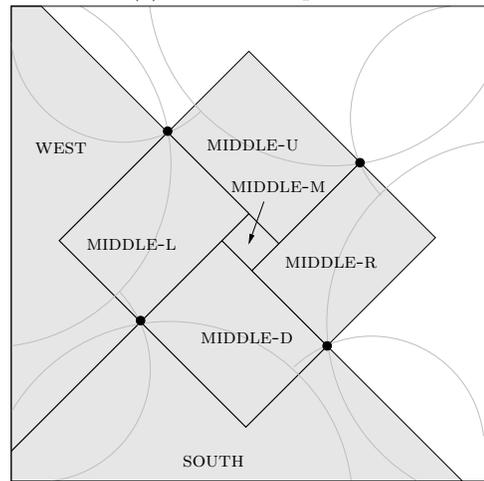
Assume that  $i_{CL}^{UM}$  is to the left of  $i_{CR}^{UM}$  (i.e., has smaller  $x$ -coordinate),  $i_{CR}^{UM}$  is above  $i_{CR}^{LM}$  (i.e., has larger  $y$ -coordinate),  $i_{CR}^{LM}$  is to the right of  $i_{CL}^{LM}$ , and  $i_{CL}^{LM}$  is below  $i_{CL}^{UM}$ . We call this the *standard configuration*. The treatment of other alternative configurations is deferred to subsequent sections. For the standard configuration, define  $i_1$  to be  $i_{CL}^{UM}$ ,  $i_2$  to be  $i_{CR}^{UM}$ ,  $i_3$  to be  $i_{CR}^{LM}$ , and  $i_4$  to be  $i_{CL}^{LM}$ , as shown in Figure 3a. For an arbitrary intersection point  $i_s$ , let  $l_s$  and  $l'_s$  be the two lines through  $i_s$  with slope 1 and  $-1$ , respectively. As illustrated in Figure 3b, define MIDDLE to be the region that is the intersection of the halfplanes below  $l_1$  and  $l'_2$  and the halfplanes above  $l_3$  and  $l'_4$ . Let MIDDLE-L be the region that is the intersection of the halfplanes below  $l_1$  and  $l'_1$  and the halfplanes above  $l_4$  and  $l'_4$ , and similarly let MIDDLE-R be the region that is the intersection of the halfplanes below  $l_2$  and  $l'_2$  and the halfplanes above  $l_3$  and  $l'_3$ , as shown in Figure 3b.

We claim that the coverage requirements of all points in the regions MIDDLE-L and MIDDLE-R are met in  $OPT_B$  by disks with centre in LEFT  $\cup$  RIGHT. We state the arguments for points within the region MIDDLE-L, and identical arguments apply to MIDDLE-R. Observe that the region MIDDLE-L lies entirely below the 1-watching  $t_l$  envelope for UM. This is because MIDDLE-L is contained in the 90 degree cone below  $i_1$  that lies between  $l_1$  and  $l'_1$ , while the 1-watching  $t_l$  envelope for UM lies in the union of the halfplanes above  $l_1$  and above  $l'_1$  (this is because the tangent to any point in the 1-watching  $t_l$  envelope for UM has direction between  $-\pi/4$  and  $\pi/4$ , which follows because  $S_{ij}$  has side length less than  $\sqrt{2}$  and the disks have radius 2). Similarly, MIDDLE-L lies entirely above the 1-watching  $t_l$  envelope for LM. Therefore, it is not possible that a point in MIDDLE-L is covered by a disk with centre in UM or LM in  $OPT_B$ , and so the coverage requirements of all points in MIDDLE-L are indeed met in  $OPT_B$  by disks from LEFT  $\cup$  RIGHT. Thus, we can classify the points in MIDDLE-L and MIDDLE-R accordingly.

The regions MIDDLE-U and MIDDLE-D are defined analogously to MIDDLE-L and MIDDLE-R, with MIDDLE-U enclosed by lines  $l_1, l'_1, l_2, l'_2$  and MIDDLE-D enclosed by lines  $l_3, l'_3, l_4, l'_4$ , see Figure 3b. Furthermore, the region obtained by removing MIDDLE-L, MIDDLE-U, MIDDLE-R and MIDDLE-D from MIDDLE (which may be empty) is denoted by MIDDLE-M. By arguments analogous to those given above, the coverage requirements of points in regions MIDDLE-U and MIDDLE-D are met in  $OPT_B$  by disks from UPPER  $\cup$  LOWER. Hence, the points in these regions can be classified accordingly. Fi-



(a) Intersection points



(b) Regions

Figure 3: Intersection points of 1-watching envelopes, and resulting central and peripheral regions

nally, the region MIDDLE-M lies outside all four 1-watching  $t_l$  envelopes, and so the points in that region can only be covered in  $OPT_B$  by disks from regions UL, UR, LL or LR. These regions are in UPPER  $\cup$  LOWER and in LEFT  $\cup$  RIGHT, so we can (arbitrarily giving preference to the former) classify these points as points that have to be covered by disks in UPPER  $\cup$  LOWER.

We refer to the part of  $S_{ij}$  that is not in MIDDLE as the *peripheral part*. Consider the peripheral area SOUTH shown in Figure 3b. This is the area that is the union of the lower-shadow of  $i_3$  and the lower-shadow of  $i_4$ . Recall that points in SOUTH that are also in a 2-watching sandglass have already been classified and are not considered further. For any remaining point  $p$  located in SOUTH we know that  $p$  is covered by a disk from LM (as SOUTH lies below the 1-watching  $t_l$  envelope for LM). Furthermore, if  $p$  has to be covered by a second disk, we know that  $p$  is covered by at least one disk whose centre is not in LM, because otherwise  $p$  would lie in the 2-watching  $t_l$  envelope for LM and would have been classified already. Hence, if  $k_p = 1$ , we specify that  $p$  must be covered by UPPER  $\cup$  LOWER, and if  $k_p = 2$ ,

we specify that  $p$  must be covered once by  $\text{LEFT} \cup \text{RIGHT}$  and once by  $\text{LM} \cup \text{UM}$ . The areas WEST, NORTH and EAST are defined and handled analogously.

As each point in  $P'_{ij}$  is contained in one of the areas defined above, all these points are classified, i.e, we have determined for each point how often it must be covered by  $\text{LEFT} \cup \text{RIGHT}$  (or  $\text{CL} \cup \text{CR}$ ) and how often by  $\text{UPPER} \cup \text{LOWER}$  (or  $\text{UM} \cup \text{LM}$ ).

#### 4.4 Areas in a Square – Other Configurations

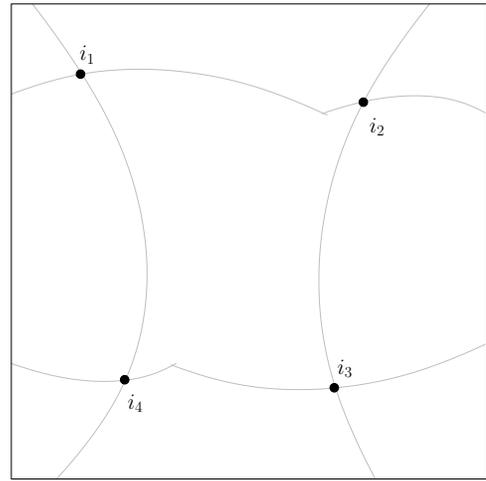
Recall that  $i_{\text{CL}}^{\text{UM}}$  is the intersection point between the 1-watching  $t_l$  envelopes for UM and CL, and  $i_{\text{CL}}^{\text{LM}}$ ,  $i_{\text{CR}}^{\text{LM}}$  and  $i_{\text{CR}}^{\text{UM}}$  are defined analogously. In the standard configuration shown in Figure 3a, we have that  $i_{\text{CL}}^{\text{UM}}$  is to the left of  $i_{\text{CR}}^{\text{UM}}$ ,  $i_{\text{CR}}^{\text{UM}}$  above  $i_{\text{CR}}^{\text{LM}}$ ,  $i_{\text{CL}}^{\text{UM}}$  above  $i_{\text{CL}}^{\text{LM}}$ , and  $i_{\text{CL}}^{\text{LM}}$  to the left of  $i_{\text{CR}}^{\text{LM}}$ . In the following, we discuss how to handle all the other possible configurations.

The definition of 2-watching  $t_l$  sandglasses does not depend on the configuration of intersection points, this means 2-watching sandglasses can be handled as before. The definition of the peripheral areas is adapted as follows. The area SOUTH is the union of the lower-shadow of the lower of the two points  $i_{\text{CL}}^{\text{UM}}$ ,  $i_{\text{CL}}^{\text{LM}}$  and the lower-shadow of the lower of the two points  $i_{\text{CR}}^{\text{UM}}$ ,  $i_{\text{CR}}^{\text{LM}}$ . The area WEST is the union of the left-shadow of the point that is further left among the two points  $i_{\text{CL}}^{\text{UM}}$ ,  $i_{\text{CR}}^{\text{UM}}$  and the left-shadow of the point that is further left among the two points  $i_{\text{CL}}^{\text{LM}}$ ,  $i_{\text{CR}}^{\text{LM}}$ . The definitions of EAST and NORTH are analogous. We observe that each of the four peripheral regions can be handled in the same way as before. For example, we still have that every point in SOUTH is covered by a disk with centre in LM, and if the point has to be covered by a second disk, there is a disk covering it with centre in  $\text{LEFT} \cup \text{RIGHT}$ .

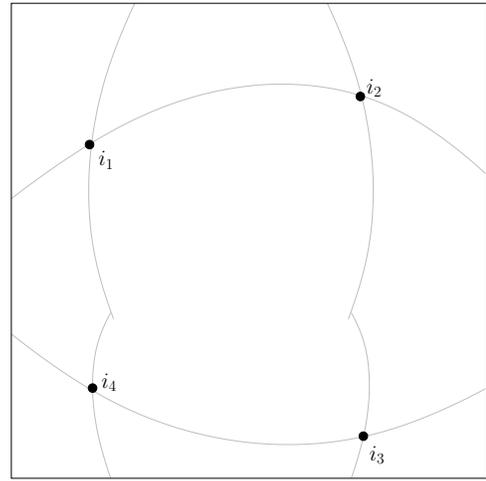
Let us introduce some terminology. We say that the 1-watching  $t_l$  envelopes for CL and CR are *opposite*, and so are the envelopes for UM and LM. Furthermore, we say that the 1-watching  $t_l$  envelopes for CL and CR *overlap* if  $i_{\text{CL}}^{\text{UM}}$  is to the right of  $i_{\text{CR}}^{\text{UM}}$  and  $i_{\text{CL}}^{\text{LM}}$  is to the right of  $i_{\text{CR}}^{\text{LM}}$ . In other words, the relations between  $i_{\text{CL}}^{\text{UM}}$  and  $i_{\text{CR}}^{\text{UM}}$  and between  $i_{\text{CL}}^{\text{LM}}$  and  $i_{\text{CR}}^{\text{LM}}$  are both reversed compared to the standard configuration. Similarly, we say that the 1-watching  $t_l$  envelopes for CL and CR *cross* if only one of the two relations is reversed compared to the standard configuration. For the 1-watching  $t_l$  envelopes for UM and LM, the notions of overlapping and crossing are defined analogously by considering the relations between  $i_{\text{CL}}^{\text{UM}}$  and  $i_{\text{CL}}^{\text{LM}}$  and between  $i_{\text{CR}}^{\text{UM}}$  and  $i_{\text{CR}}^{\text{LM}}$ .

**One Pair of Opposite Envelopes Overlap.** The first alternative configuration that we consider is the case where one pair of opposite envelopes overlap, and the other pair of opposite envelopes neither overlap nor cross. Without loss of generality, assume that the 1-watching  $t_l$  envelopes for UM and LM overlap, as illustrated in Figure 4a. (The other case is symmetric.) Note that  $i_{\text{CL}}^{\text{LM}}$  is above  $i_{\text{CL}}^{\text{UM}}$ , and  $i_{\text{CR}}^{\text{LM}}$  is above  $i_{\text{CR}}^{\text{UM}}$ .

Let  $i_1 = i_{\text{CL}}^{\text{LM}}$ ,  $i_2 = i_{\text{CR}}^{\text{LM}}$ ,  $i_3 = i_{\text{CR}}^{\text{UM}}$ ,  $i_4 = i_{\text{CL}}^{\text{UM}}$ . Now let the regions MIDDLE, MIDDLE-L, etc. be defined in terms of  $i_1, i_2, i_3, i_4$  as in the standard configuration. Consider the regions MIDDLE-L, MIDDLE-M and MIDDLE-R. These regions are always enclosed within both the UM envelope and the LM envelope. Therefore, points within those regions can be classified as requiring to be covered by disks from  $\text{UPPER} \cup \text{LOWER}$ . The regions MIDDLE-U and MIDDLE-D are always outside the 1-watching  $t_l$  envelopes for CL and CR, and therefore the points in these regions can also be classified as requiring to



(a) The UM envelope and the LM envelope overlap



(b) UM envelope overlaps LM envelope, and LM envelope overlaps CR envelope

Figure 4: Configurations with overlapping envelopes

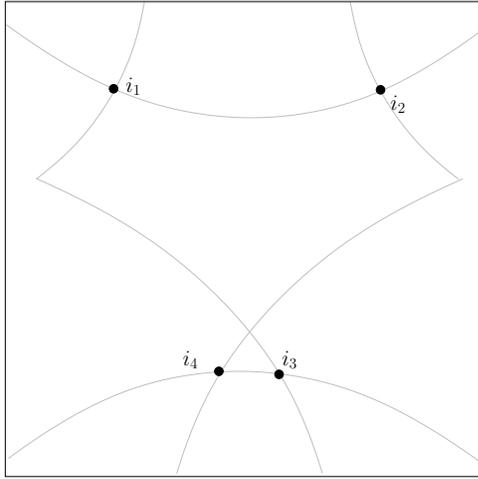
be covered only by disks in  $\text{UPPER} \cup \text{LOWER}$ —for the same reason as in the standard configuration.

**Both Pairs of Opposite Envelopes Overlap.** Now consider the configuration shown in Figure 4b where the CL and CR envelopes overlap, and the UM and LM envelopes also overlap. For the definition of the MIDDLE regions, let  $i_1 = i_{\text{CR}}^{\text{LM}}$ ,  $i_2 = i_{\text{CL}}^{\text{LM}}$ ,  $i_3 = i_{\text{CL}}^{\text{UM}}$ ,  $i_4 = i_{\text{CR}}^{\text{UM}}$ .

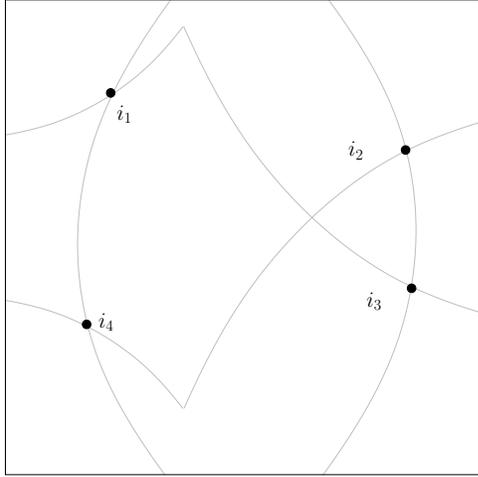
Points within the regions MIDDLE-U and MIDDLE-D are covered twice by disks in  $\text{LEFT} \cup \text{RIGHT}$  as those regions are enclosed within both the CL envelope and the CR envelope. Similarly, the regions MIDDLE-L and MIDDLE-R are contained entirely inside the UM envelope and the LM envelope and as such all points within these regions can be classified as requiring to be covered by disks from  $\text{UPPER} \cup \text{LOWER}$ . Points within the region MIDDLE-M can be classified as requiring to be covered by disks from  $\text{UPPER} \cup \text{LOWER}$ .

**One Pair of Opposite Envelopes Cross.** Now consider a configuration where one pair of opposite envelopes cross, and the other pair of opposite envelopes neither cross nor overlap. Without loss of generality, assume that the 1-watching

$t_l$  envelopes for CL and CR cross, and that  $i_{CL}^{UM}$  is to the left of  $i_{CR}^{UM}$  and  $i_{CL}^{LM}$  is to the right of  $i_{CR}^{LM}$ , as shown in Figure 5a. (The other cases are symmetric.)



(a) CL envelope crosses CR envelope



(b) UM envelope crosses LM envelope, CR and CL envelopes overlap

Figure 5: Configurations with crossing envelopes

Let  $i_1 = i_{CL}^{UM}$ ,  $i_2 = i_{CR}^{UM}$ ,  $i_3 = i_{CL}^{LM}$ , and  $i_4 = i_{CR}^{LM}$ , and define the MIDDLE regions accordingly. The areas MIDDLE-L, MIDDLE-R and MIDDLE-M are outside the 1-watching  $t_l$  envelopes for UM and LM, and thus the points in these regions can be classified as requiring to be covered by disks from LEFT  $\cup$  RIGHT. The area MIDDLE-U is outside the 1-watching  $t_l$  envelopes for CL and CR, and thus the points in MIDDLE-U can be classified as requiring to be covered by disks from UPPER  $\cup$  LOWER. The area MIDDLE-D is inside the 1-watching envelopes for both UM and LM, and thus the points in MIDDLE-D can be classified as requiring to be covered by disks from LEFT  $\cup$  RIGHT.

**One Pair of Envelopes Overlap, One Pair of Envelopes Cross.** Consider now the case where two opposite envelopes overlap and the two other opposite envelopes cross. Without loss of generality, consider the case where the UM envelope crosses the LM envelope and the CL enve-

lope overlaps the CR envelope, and assume that  $i_{CR}^{UM}$  is above  $i_{CR}^{LM}$  and  $i_{CL}^{UM}$  is below  $i_{CL}^{LM}$ , as illustrated in Figure 5b.

The area MIDDLE-L is outside the envelopes for UM and LM, and the points in MIDDLE-L can be classified as requiring to be covered by disks in LEFT  $\cup$  RIGHT. The area MIDDLE-R is within both the UM and the LM envelope, and the points in MIDDLE-R are classified as requiring to be covered by disks in UPPER  $\cup$  LOWER. The regions MIDDLE-U, MIDDLE-M and MIDDLE-D are contained in the envelopes for both CL and CR, and all points in these regions are classified as requiring to be covered by disks in LEFT  $\cup$  RIGHT.

**Both Pairs of Opposite Envelopes Cross.** The only configuration that has not yet been considered is the case where both pairs of opposite envelopes cross. One such configuration would have  $i_{CL}^{UM}$  strictly to the left of  $i_{CR}^{UM}$ ,  $i_{CL}^{LM}$  strictly to the right of  $i_{CR}^{LM}$ ,  $i_{CL}^{UM}$  strictly above  $i_{CR}^{UM}$ , and  $i_{CL}^{LM}$  strictly below  $i_{CR}^{LM}$ . (We can assume strict relationships since if two points coincide, we are free to choose the relation and arrive at a previously considered configuration where it is not the case that both pairs of envelopes cross.) To show that this is impossible, consider the line  $l$  of slope  $-1$  through  $i_{CL}^{UM}$ . Since  $i_{CL}^{UM}$  is above  $i_{CL}^{LM}$  and both points lie on the CL envelope, we get that  $i_{CL}^{LM}$  is strictly below the line  $l$ . Since  $i_{CL}^{UM}$  is on the line  $l$  and  $i_{CR}^{UM}$  is to the right of  $i_{CL}^{UM}$  and also lies on the UM envelope, we get that  $i_{CR}^{UM}$  is above  $l$ . Since  $i_{CR}^{UM}$  is below  $i_{CR}^{LM}$  and both points are on the CR envelope, it follows that  $i_{CR}^{LM}$  is above  $l$ . Since  $i_{CR}^{LM}$  is to the right of  $i_{CL}^{LM}$  and both points are on the LM envelope, we obtain that  $i_{CL}^{LM}$  is strictly above the line  $l$ . This is a contradiction to the conclusion that  $i_{CL}^{LM}$  is strictly below  $l$  that we derived above. Hence, this configuration is not possible. The other configurations where both envelopes cross can be excluded similarly.

**Complexity of Enumeration.** For each of the  $K^2$  squares  $S_{ij}$  in a block, and for each event type  $t_l$  in  $T$ , we enumerate up to two disks with centre in  $S_{ij}$  (these form the set  $OPT'_B$  of disks that are determined to be in the solution by the guessing stage), eight points defining the 2-watching sandglasses, and four intersection points (each identified by two disks) of the 1-watching envelopes. Hence, if there are  $m_B$  disks and  $n_B$  points in the given instance of W2CUD- $T$  in a block  $B$ , there are  $O((m_B^2 n_B^8 m_B^8)^{|T|}) = (m_B + n_B)^{O(|T|)}$  choices per square, and thus  $(m_B + n_B)^{O(K^2|T|)}$  choices for the whole block  $B$ . Since  $K$  and  $|T|$  are constants, this is a polynomial number of choices. For each of these choices, the resulting classification of points and candidate set of disks for  $OPT'_B$  are passed to the second stage of the algorithm from Section 3. (For some choices, the algorithm may find that there is no feasible solution.) The algorithm outputs the cheapest solution among the feasible solutions obtained for any of the  $(m_B + n_B)^{O(K^2|T|)}$  different guesses of the properties of the optimal solution. For the choice in which the guessed properties are indeed those of  $OPT_B$ , the algorithm is guaranteed to produce a 6-approximation. Hence, the solution output by the algorithm is a 6-approximation for instances of W2CUD- $T$  where all points lie in one  $K \times K$  block.

## 5. CONCLUSION

We have presented a  $(6 + \varepsilon)$ -approximation algorithm for the target coverage problem with composite events and fault-tolerance requirements, both for the lifetime maximisation

variant and for the problem of covering all event points by sensors of minimum total cost. Our approach is based on guessing properties of the optimal solution (by enumeration) and then using these properties to guide a dynamic programming algorithm. This is a generalisation of the approach employed by Huang et al. [13] to obtain a  $(6+\epsilon)$ -approximation for weighted set cover with unit disks. For the latter problem, subsequent work has improved the approximation ratio to  $5+\epsilon$  [7] and then to  $4+\epsilon$  [9, 20]. The main idea of these improvements is to perform the dynamic programming in several strips simultaneously. One possible direction for future work would be to see whether these improvements can also be adapted to the fault-tolerant target coverage problem with composite events.

Another question of interest is whether our approach can be adapted to arbitrary coverage requirements  $k_p$ , i.e., without the restriction  $k_p \leq 2$  for all  $p \in P$ . For W2CUD- $T$  and also for the special case of weighted geometric set cover with unit disks, it is an interesting open problem whether a polynomial-time approximation scheme (PTAS) can be obtained. Finally, the study of settings where the sensors and targets are located in three-dimensional space would be interesting.

## 6. REFERENCES

- [1] C. Ambühl, T. Erlebach, M. Mihalák, and M. Nunkesser. Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs. In *Proceedings of the 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2006)*, LNCS 4110, pages 3–14. Springer, 2006.
- [2] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky. Power efficient monitoring schedules in sensor networks. In *IEEE Wireless Communication and Networking Conference (WCNC 2004)*, pages 2329–2334, 2004.
- [3] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky. Efficient energy management in sensor networks. In Y. Xiao and Y. Pan, editors, *Ad Hoc and Sensor Networks, Wireless Networks and Mobile Computing*, volume 2, pages 71–90. Nova Science Publishers, 2005.
- [4] J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità. An improved LP-based approximation for Steiner tree. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC 2010)*, pages 583–592. ACM, 2010.
- [5] C. Chekuri, K. L. Clarkson, and S. Har-Peled. On the set multi-cover problem in geometric settings. In *Proceedings of the 25th Annual Symposium on Computational Geometry (SCG’09)*, pages 341–350. ACM, 2009.
- [6] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.
- [7] D. Dai and C. Yu. A  $(5+\epsilon)$ -approximation algorithm for minimum weighted dominating set in unit disk graph. *Theoretical Computer Science*, 410(8-10):756–765, 2009.
- [8] A. Dhawan, C. T. Vu, A. Zelikovsky, Y. Li, and S. K. Prasad. Maximum lifetime of sensor networks with adjustable sensing range. In *Proceedings of the 7th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing & International Workshop on Self-Assembling Wireless Networks (SNPD-SAWN 2006)*, pages 285–289, 2006.
- [9] T. Erlebach and M. Mihalák. A  $(4+\epsilon)$ -approximation for the minimum-weight dominating set problem in unit disk graphs. In *Proceedings of the 7th International Workshop on Approximation and Online Algorithms (WAOA 2009)*, LNCS 5893, pages 135–146. Springer, 2009.
- [10] T. Erlebach and A. Shahnaz. Approximating node-weighted multicast trees in wireless ad-hoc networks. In *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly (IWCMC 2009)*, pages 639–643. ACM, 2009.
- [11] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM Journal on Computing*, 37(2):630–652, 2007.
- [12] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, 32(1):130–136, 1985.
- [13] Y. Huang, X. Gao, Z. Zhang, and W. Wu. A better constant-factor approximation for weighted dominating set in unit disk graph. *Journal of Combinatorial Optimization*, 18(2):179–194, 2009.
- [14] M. Marta, Y. Yang, and M. Cardei. Energy-efficient composite event detection in wireless sensor networks. In *Proceedings of the 4th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2009)*, LNCS 5682, pages 94–103. Springer, 2009.
- [15] P. Sanders and D. Schieferdecker. Lifetime maximization of monitoring sensor networks. In *Proceedings of the 6th International Workshop on Algorithms for Sensor Systems, Wireless Ad Hoc Networks, and Autonomous Mobile Entities (ALGOSENSORS 2010)*, LNCS 6541, pages 134–147. Springer, 2010.
- [16] M. T. Thai, F. Wang, D. H. Du, and X. Jia. Coverage problems in wireless sensor networks: designs and analysis. *International Journal of Sensor Networks*, 3:191–200, May 2008.
- [17] C. Vu, R. Beyah, and Y. Li. Composite event detection in wireless sensor networks. In *IEEE International Performance, Computing, and Communications Conference (IPCCC 2007)*, pages 264–271, 2007.
- [18] Q. Zhao and M. Gurusamy. Lifetime maximization for connected target coverage in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 16:1378–1391, December 2008.
- [19] F. Zou, X. Li, S. Gao, and W. Wu. Node-weighted Steiner tree approximation in unit disk graphs. *Journal of Combinatorial Optimization*, 18(4):342–349, 2009.
- [20] F. Zou, Y. Wang, X. Xu, X. Li, H. Du, P. Wan, and W. Wu. New approximations for minimum-weighted dominating sets and minimum-weighted connected dominating sets on unit disk graphs. *Theoretical Computer Science*, 412:198–208, January 2011.