Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 918252, 25 pages
doi:10.1155/2012/918252

Research Article

The Complexity of the Minimum Sensor Cover Problem with Unit-Disk Sensing Regions over a Connected Monitored Region

Ren-Song Ko

Department of Computer Science and Information Engineering, National Chung Cheng University, 168 University Road, Min-Hsiung Chia-Yi 621, Taiwan

Correspondence should be addressed to Ren-Song Ko, korenson@cs.ccu.edu.tw

Received 17 June 2011; Revised 12 August 2011; Accepted 16 August 2011

Academic Editor: Yuhang Yang

This paper considers the complexity of the Minimum Unit-Disk Cover (MUDC) problem. This problem has applications in extending the sensor network lifetime by selecting minimum number of nodes to cover each location in a geometric connected region of interest and putting the remaining nodes in power saving mode. MUDC is a restricted version of the well-studied Minimum Set Cover (MSC) problem where the sensing region of each node is a unit-disk and the monitored region is geometric connected, a well-adopted network model in many works of the literature. We first present the formal proof of its NP-completeness. Then we illustrate several related optimum problems under various coverage constraints and show their hardness results as a corollary. Furthermore, we propose an efficient algorithm for reducing MUDC to MSC which has many well-known algorithms for approximated solutions. Finally, we present a decentralized scalable algorithm with a guaranteed performance and a constant approximation factor algorithm if the maximum node density is fixed.

1. Introduction

The research in wireless ad hoc networks has rapidly grown in recent years due to their applications in civil and military domains. Combined with recent developments in micro-electro-mechanical systems and low-cost mass production, various small and low-power devices that integrate sensors with limited on-board processing and wireless communication capabilities begin to emerge. Hence, wireless networks with large numbers of sensors become possible and open up potential of many new applications, such as environment monitoring and surveillance [1].

With the available technology, the sensors are usually battery powered. Due to size and cost constraints, the energy available at each sensor is limited. Therefore, one of the important design considerations in sensor networks is to minimize energy consumption and prolong network lifetime. There is a significant amount of the literature addressing the issue of efficient energy management in generic wireless ad hoc networks from various perspectives, such as medium access control [2, 3], routing [4, 5], broadcasting [6, 7], multicasting [8, 9], and topology control [10, 11]. Of course, similar approaches have been also considered in wireless sensor networks [12–15].

An alternative approach commonly adopted in sensor networks is based on scheduling sensor activity so that some nodes may enter the power saving mode while the remaining active nodes can still provide continuous service [14, 16]. For instance, if all the sensor nodes simultaneously operate in active mode, an excessive amount of energy is wasted and the data collected is highly correlated and redundant. In addition, multiple packet collisions may occur when all the sensors in a certain region try to transmit as a result of a triggering event. Several research results [3, 16] illustrate that a mode of operation alternating active and inactive battery states has a significant reduced energy consumption.

However, such scheduling schemes may face new constraints about sensing coverage introduced by their distributed sensing applications [17]. For example, surveillance applications may require each location of monitored regions to be covered by at least one sensor, while many stronger environmental monitoring, such as military applications, require multiple sensors for fault-tolerant purpose. Besides, triangulation positioning-based tracking applications [18, 19]

may require at least three sensors at any locations. Data sampling applications may require a given percentage of monitored regions to be covered.

Therefore, this paper considers the scheduling approach that extends the network lifetime by minimizing the number of active nodes while maintaining coverage constraints. As mentioned earlier, the advantage of this approach is that less packet collisions may occur since less spatially close sensors try to transmit highly correlated and redundant information as a result of a triggering event. Hence, the lifetime of each sensor cover may be extended.

We model a sensor network as a 2D geometric connected region monitored by a set of deployed sensor nodes with unit-disk sensing regions, a realistic assumption that is well adopted in many network models. The coverage constraint is that each location of the 2D region is covered by at least one active node. Note that optimum sensor cover problems may be solved by partitioning monitored regions into disjoint sectors [20, 21]. Here a sector is a maximum region covered by the same set of nodes. Hence the minimum sensor cover problem is transformed to the Minimum Set Cover (MSC) problem which is NP-complete [22] and has been studied extensively in the literature [23–26]. However, with the additional unit-disk sensing region and geometric connected monitored region restrictions, the problem considered here is only a restricted version of MSC, and, to the best of our knowledge, its complexity is still unknown. (That is, it is not trivial to transform each instance of MSC to an instance in the minimum sensor cover problem with unit-disk sensing regions over a connected monitored region in polynomial time.) Thus, we will answer this fundamental question and present the formal proof of its NP-completeness. Furthermore, we illustrate several related optimum problems under different coverage constraints and show their hardness results as a corollary.

Next, we propose the arc sampling algorithm which may effectively and efficiently reduce MUDC to MSC. Consequently, many well-known algorithms can be applied to find approximated solutions. In addition, we present a decentralized scalable algorithm with a guaranteed performance, and a constant approximation factor algorithm if the maximum node density is fixed. Finally, we illustrate simulation results to evaluate the proposed algorithms.

2. Related Work

2.1. Coverage Problems. Meguerdichian et al. [17] defined the coverage problems from several different application domains including deterministic, statistical, worst, and best cases. They also presented optimum polynomial time algorithms to evaluate paths that are the best and least monitored in the sensor network. The work in [27] further defined the exposure problem as measure of how well an object can be observed by the sensor network while it moves along an arbitrary path with an arbitrary velocity. A localized exposure-based coverage algorithm was proposed in [28] for finding the minimal exposure path between two points.

Furthermore, Gui and Mohapatra [29] considered the object tracking applications in which networks operate between surveillance state and tracking state. During surveillance state, they devised a set of metrics for quality of surveillance for detecting moving objects and quantify the trade-off between power conservation and quality. They also proposed an algorithm for each node to determine when to wake up or sleep during the tracking stage.

Tian and Georganas [30] developed a coverage-preserving scheduling scheme to reduce energy consumption by turning off some redundant nodes based on some eligibility rules. Carbunar et al. [31] proposed distributed algorithms for detecting and eliminating redundancy in a sensor network while preserving the network's coverage via Voronoi diagrams, even in cases of sensor failures or insertion of new sensors.

Huang and Tseng [32] considered the k-coverage problem to determine whether every point in the monitored region is covered by at least k nodes. They reduced this problem to the perimeter-coverage problem which determines the coverage degree of the perimeter of each node's sensing region and presented polynomial-time algorithms in the number of nodes.

In addition to coverage, connectivity also needs to be assured to make sensor networks successfully. It has been shown in [33] that if the communication range of sensors is at least twice as large as their sensing range, then full coverage of a convex region implies connectivity. Wang et al. [34] presented a Coverage Configuration Protocol (CCP) that allows the network to self-configure dynamically to achieve guaranteed degrees of coverage and connectivity.

2.2. Minimum Sensor Cover Problems. In [21], Funke et al. proposed the greedy sector cover algorithm which selects a node that covers the maximum number of uncovered sectors at each iteration step. That is, the problem is reduced to MSC and solved by the greedy algorithm. They proved that the well-known approximation factor $O(\log m)$ remains tight in this restricted version. Here m is the maximum number of sectors covered by a single node. To obtain better approximation factors, they also presented a grid placement algorithm and a distributed dominating cover algorithm. These two algorithms have constant approximation factors, but cannot guarantee the full coverage.

Gupta et al. [35] designed an $O(\log |N|)$ centralized approximation algorithms with the connectivity constraint. Here $|N|$ is the number of sensor nodes. In their definition of the sensor cover problem, the sensing region can take any convex shape. They also mentioned that such a problem is NP-hard as the less general problem of covering discrete points using line segments is known to be NP-hard [36]. On the other hand, the sensing region considered in this paper is restricted to a unit-disk, which is well adopted in many network models. We will prove such a problem remains NP-complete even with the unit-disk restriction. They also proposed a distributed algorithm based on node priorities, but did not provide any guarantee on the solution size.

2.3. Related Optimum Problems. Fowler et al. [37] proved the NP-completeness of the Box Cover problem which aims at

finding the minimum number of identical rectangles to cover a set of given points. Similarly, Megiddo and Supowit [38] considered the Circle Covering problem which is equivalent to the Geometric Disc Covering problem, that is, to find the minimum number of identical disks to cover a set of given points. There are two fundamental differences between MUDC and these two problems. In these two problems, the covered object is a set of discrete points but not a connected region. Hence, in the proofs of the NP-completeness, we have less flexibility in the connected case than in the discrete cases, since we need to ensure the monitored region is connected while constructing a problem instance. Furthermore, the two problems have the flexibility to determine the "good" locations of covering objects (rectangles or disks), which could be anywhere on the plane. On the other hand, in MUDC, the locations of disks are pre-deployed.

Marathe et al. [39] considered several basic optimization problems for unit-disk graphs with hierarchical structures. They presented a general technique to prove the hardness results of several problems. The hardness of these problems, including Box Cover and Circle Covering, was proved via satisfiability problems. The reduction strategy was to use some geometric structures to represent variables. Each clause is represented by a special structure that "glues" the corresponding structures of the variables in the clause.

There are several polynomial approximation algorithms [40–42] for the Geometric Disc Covering problem. Franceschetti et al. pointed out in [43] that the number of possible disk positions can be bounded if any disk that covers at least two points has two of these points on its border. Hence, by performing a search on a subset of the possible disk positions, the running time of these algorithms becomes polynomial and the solution sizes are guaranteed. They also gave a detailed comparison of these algorithms in [44].

3. Preliminaries

In this section, we define the Minimum Unit-Disk Cover (MUDC) problem that aims at finding the least number of nodes with unit-disk sensing regions to fully cover a designated connected region. We prove this problem is intractable; that is, it belongs to the NP-complete class.

Definition 1. Consider a two-dimensional Euclidean metric space $\mathbb{E}$, the unit-disk sensing region of a given node $n \in \mathbb{E}$ is defined as $\text{disk}(n) = \{x \in \mathbb{E} \mid d(x,n) \leq 1\}$. (In the context of discussing MUDC, we represent a node by its geometric location without any confusion.) Here $d(x,n)$ is the distance in Euclidean metric between x and n. Furthermore, the unit-disk sensing region of a set U of nodes is defined as $\text{disk}(U) = \bigcup_{n \in U} \text{disk}(n)$.

Definition 2. A two-dimensional finite region A is said to be *unit-disk covered* by a set U of nodes in a two-dimensional Euclidean metric space if $A \subseteq \text{disk}(U)$. Furthermore, U is called a *unit-disk cover* (UDC) of A.

The objective is to find the minimum unit-disk cover (MUDC) of A. Note that the optimum problems discussed in this paper could be solved by their associated decision problems in polynomial time. Therefore, we discuss the decision version of MUDC instead, and it can be formally stated in the following.

Problem 3 (MUDC). Given a set N of nodes in a two-dimensional Euclidean metric space $\mathbb{E}$, a two-dimensional geometric connected finite region $A \subset \mathbb{E}$, and a positive integer K, determine whether there is a subset $U \subseteq N$ with $|U| \leq K$ such that $A \subseteq \text{disk}(U)$. Here $|U|$ is the cardinality of U.

For simplicity's sake, the geometry of an MUDC problem, that is, the region A and the set N of nodes, is denoted as (A, N).

Thus, we will prove the following theorem.

Theorem 4. *MUDC is NP-complete.*

The NP-completeness of MUDC will be proved by reduction from the Planar 3-SAT problem, which is known to be NP-complete [45].

Problem 5 (**Planar 3-SAT**, P3SAT). Given a set of variables $V = \{v_1, v_2, \ldots, v_{\eta'}\}$ and a set of clauses $C = \{c_1, c_2, \ldots, c_\eta\}$ over V such that each $c \in C$ has $2 \leq |c| \leq 3$ (denoted as a boolean formula B) determine whether there is an assignment for the variables so that all clauses are satisfied. (In Lichtenstein's NP-completeness proof of P3SAT [45], an instance of **3SAT** is transformed to an instance of P3SAT with $|c|$ being 2 or 3. Thus, the restriction, $2 \leq |c| \leq 3$, does not change the complexity of the problem. This restriction is required to prove Lemma 17.) Furthermore, the bipartite graph $G_B = \{V \cup C, E\}$ (in this NP-completeness proof of MUDC, G_B will be used to construct an equivalent MUDC problem for B) is planar, where $E = \{(v_i, c_j) \mid v_i \in c_j \text{ or } \overline{v_i} \in c_j\}$. (We remove the edges $\{(v_i, v_{i+1}) \mid 1 \leq i < m'\} \cup \{(v_{m'}, v_1)\}$ without any changes in the difficulty of the problem [46].)

That is, let B be a boolean formula in P3SAT with η clauses and η' variables. We wish to construct an equivalent MUDC problem with the geometry $\text{MUDC}(B) = (A_B, N_B)$ where A_B and N_B are the region and the set of nodes transformed from G_B, respectively.

Inspired by Lichtenstein's NP-completeness proof of the Geometric Connected Dominating Set problem [45], $\text{MUDC}(B) = (A_B, N_B)$ is constructed via structures. Each structure S is a geometry containing a polygon A and a set of nodes N and denoted as $S = (A, N)$. Variables, clauses, and edges of G_B are represented by various structures. Hence, $\text{MUDC}(B)$ is constructed from G_B by replacing variables, clauses, and edges with their corresponding structures.

Each structure $S = (A, N)$ is constructed in such a way that N can be partitioned into two disjoint subsets of equal size, denoted as N^+ and N^-, and the MUDC of A, except the ones representing clauses, is either N^+ or N^-. Thus, a variable is assigned *true* corresponding to that N^+ is the MUDC of A; *false* corresponds to N^-. For convenience throughout this paper, we assign each node a polarity. The node n has positive polarity if $n \in N^+$ or negative polarity if $n \in N^-$.

The property of structures mentioned above can be formally defined in the following.

Definition 6. One calls a structure $S = (A, N)$ *well aligned* if $A \subseteq \text{disk}(N^+)$ and $A \subseteq \text{disk}(N^-)$.

Definition 7. For a structure $S = (A, N)$ and $N' \subseteq N$, we call S *partially well behaved on* N', if the following preconditions hold:

(i) $|N'^+| = |N'^-| = |N'|/2$

(ii) if $U \subseteq N$ is a UDC of A, $|U \cap N'| \geq |N'|/2$

(iii) if $U \subseteq N$ is an MUDC of A, $U \cap N' = N'^+$ or $U \cap N' = N'^-$. (For a given set of nodes N, we denote the set of nodes with the same polarity as N with superscripts $+$ or $-$ throughout this paper. i.e., $N^+ = \{n \in N \mid n$ has positive polarity$\}$ and $N^- = \{n \in N \mid n$ has negative polarity.)

Furthermore, we call S *well behaved* if S is partially well behaved on N.

Note that, from the above definition, if $S = (A, N)$ is well behaved and $U \subseteq N$ is an MUDC of A, then $|U| = |N|/2$ and U only contains the nodes with the same polarity.

The NP-completeness proof of MUDC will proceed as follows.

(1) Describe structures representing variables, edges, and clauses. These structures have the properties defined in Definitions 6 and 7.

(2) Describe how the above structures may be connected together to represent G_B while preserving the properties defined in Definitions 6 and 7. Here the resulting composite structure is $\text{MUDC}(B) = (A_B, N_B)$.

(3) We claim that B is satisfiable if and only if A_B can be covered by half the nodes of N_B. In the proof of the claim, the properties defined in Definitions 6 and 7 will be used in the forward direction and the backward direction respectively.

4. NP-Completeness Proof of MUDC

We first prove that MUDC belongs to the NP class. This could be done since a nondeterministic algorithm needs only guess a set of nodes, U, and verify whether $A \subseteq \text{disk}(U)$. Besides, as stated in [32], this verification could be done in $O(|U|^2 \log |U|)$.

We continue the proof by reduction from the Planar 3-SAT problem. Let B be a boolean formula in P3SAT with η clauses and η' variables. We wish to construct an equivalent MUDC problem with the geometry $\text{MUDC}(B) = (A_B, N_B)$ transformed from the bipartite graph G_B.

4.1. Structures. We encode each variable by the structure, denoted as $S_v = (A_v, N_v)$, shown in Figure 1. A_v represents the shaded region which is a $d_v \times 1$ rectangle. N_v represents the set of the $2(d_v + 1)$ nodes which are positioned accordingly and used to cover A_v. Each node has a either positive or negative polarity and is represented by a square or triangle, respectively, in the figure. The d_v may be long enough to prevent unwanted interactions between nearby edge structures.

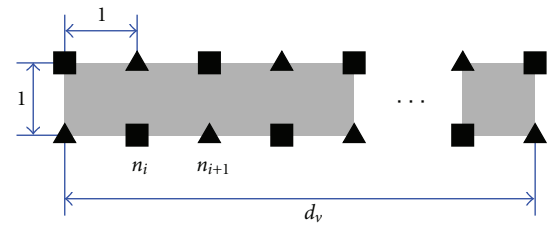


Figure 1: The structure representing a variable.

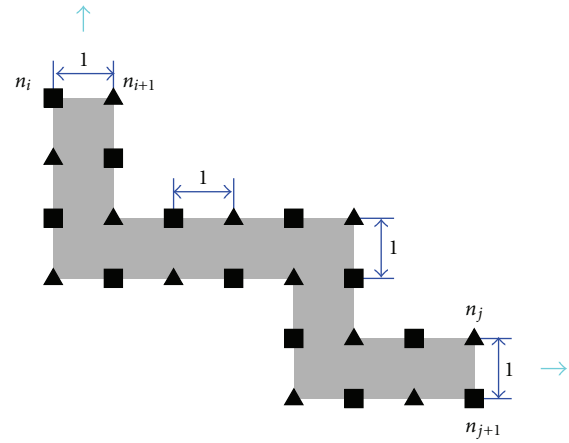


Figure 2: The structure representing an edge.

Next, we may encode an edge by a strip-like structure, denoted as $S_e = (A_e, N_e)$, which may extend horizontally and vertically. Figure 2 illustrates an example. The shaded region, denoted as A_e, is composed of rectangles. N_e represents the set of the positive and negative polar nodes which are positioned accordingly and used to cover A_e.

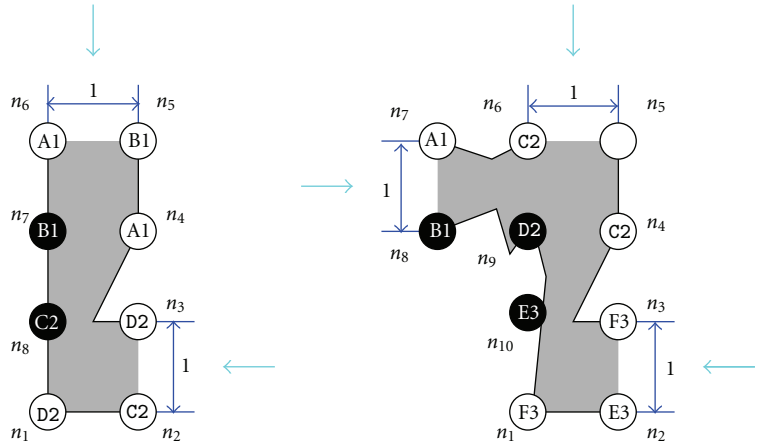
It is not hard to prove the structures S_v and S_e satisfy the following lemma.

Lemma 8. *The structures $S_v = (A_v, N_v)$ shown in Figure 1 and $S_e = (A_e, N_e)$ shown in Figure 2 are well aligned and well behaved.*

Proof. The lemma may be proved by induction. For the sake of brevity, the complete proof is given in Appendix A. □

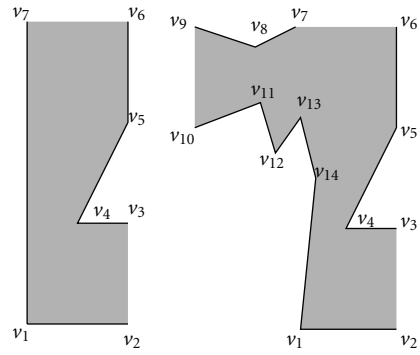
Each clause c may be represented by a structure, called an *n-way connector* and denoted as $S_c = (A_c, N_c\langle \mathcal{P}, H\rangle)$. Here $n = |c|$ and could be the value of 2 and 3. Figure 3 illustrates the possible realization of n-way connectors. A_c represents the shaded polygon that will be covered by the set of the nodes, N_c. The geometries of A_c and relative positions of nodes of N_c are shown in Figures 3 and 4 and Tables 1 and 2.

Furthermore, N_c is divided into n disjoint partitions $P_i \subseteq N_c$ with $1 \leq i \leq n$ and we denote $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$. As shown in Figure 3, each node is labeled as an alphabet



(a) 2-way connector (b) 3-way connector

FIGURE 3: The structures representing clauses.



(a) 2-way connector (b) 3-way connector

FIGURE 4: The labels of A_c.

and a numeral. The nodes with the same numeral belong to the same partition. The alphabets represent the relative polarity of nodes within the same partition. For example, in Figure 3(b), the nodes labeled as C2 and D2 belong to P_2. The two nodes labeled as C2 have the same polarity and have the opposite polarity with the two nodes labeled as D2.

Each partition P_i has a header node $h_i \in P_i$ which is indicated as a dark circle in Figure 3. We can define the polarity of the partition P_i as the polarity of its header node h_i. The set of header nodes is denoted as H. Each partition corresponds to a variable of a clause. What n-way connector should represent a clause depends on how the clause is formed from variables. For example, if c is composed of two positive literals and one negative literal, then it should be represented by a 3-way connector that has two positive partitions and one negative partition as shown in Figure 7. As described later, an edge and its ends in G_B will be transformed by connecting a variable structure to a partition of an n-way connector via an edge structure. Thus, a partition may be viewed as "extended territory" of a variable. (The definition of *territory* is given in Definition 16. Again, the key point to transform G_B to MUDC(B) is to preserve the properties defined in Definitions 6 and 7 for territories.)

TABLE 1: The geometry of the 2-way connector shown in Figure 3(a).

N_c		A_c	
Node	Position	Vertex	Position
n_1	$(0,0)$	v_1	$(0,0)$
n_2	$(1,0)$	v_2	$(1,0)$
n_3	$(1,1)$	v_3	$(1,1)$
n_4	$(1,2)$	v_4	$(\frac{1}{2},1)$
n_5	$(1,3)$	v_5	$(1,2)$
n_6	$(0,3)$	v_6	$(1,3)$
n_7	$(0,2)$	v_7	$(0,3)$
n_8	$(0,1)$		

The main reason why an n-way connector is constructed in this way is the following lemma.

Table 2: The geometry of the 3-way connector shown in Figure 3(b).

N_c		A_c	
Node	Position	Vertex	Position
n_1	$(0,0)$	v_1	$(\frac{1}{20},0)$
n_2	$(1,0)$	v_2	$(1,0)$
n_3	$(1,1)$	v_3	$(1,1)$
n_4	$(1,2)$	v_4	$(\frac{1}{2},1)$
n_5	$(1,3)$	v_5	$(1,2)$
n_6	$(0,3)$	v_6	$(1,3)$
n_7	$(-1,3)$	v_7	$(0,3)$
n_8	$(-1,2)$	v_8	$(-\frac{2}{5},2\frac{4}{5})$
n_9	$(0,2)$	v_9	$(-1,3)$
n_{10}	$(0,1\frac{1}{10})$	v_{10}	$(-1,2)$
		v_{11}	$(-\frac{7}{20},2\frac{1}{4})$
		v_{12}	$(-\frac{1}{5},1\frac{3}{4})$
		v_{13}	$(\frac{1}{20},2\frac{1}{10})$
		v_{14}	$(\frac{1}{5},1\frac{1}{2})$

Lemma 9. *Each n-way connector, $S_c = (A_c, N_c\langle\mathcal{P}, H\rangle)$, of Figure 3 has the following properties.*

(i) *For all $P_i \in \mathcal{P}$, S_c is partially well behaved on P_i. (For an MUDC of A_c, the active nodes of each partition have the same polarity. Thus, a variable of c can be assigned true or false based on the polarity of the active nodes in its corresponding partition. Here, in the context of discussing a given UDC, we call a node active if it is in the UDC.)*

(ii) *If $U \subseteq N_c$ is a UDC of A_c, then $H \cap U \neq \emptyset$. (At least one header node must be active for covering A_c.)*

(iii) *$A_c \subseteq disk(\bigcup_{1\le i\le n} P_i^{p_i})$, if $H \cap \bigcup_{1\le i\le n} P_i^{p_i} \neq \emptyset$. Here $P_i^{p_i} \subset P_i$ contains either all positive polar nodes ($p_i = +$) or all negative polar nodes ($p_i = -$). That is, $P_i^{p_i} = P_i^+$ or P_i^-. (In other words, if the active nodes of each partition have the same polarity and one of them is a header node, then A_c is covered.)*

Proof. The idea is to examine each possible case of partition P_i and ensure A_c will not be covered if, for each P_i, less than $|P_i|/2$ nodes are active or if exactly $|P_i|/2$ nodes but not having the same polarity are active. Furthermore, we need to examine whether A_c will be covered if none of header nodes is active. The complete proof is given in Appendix B. □

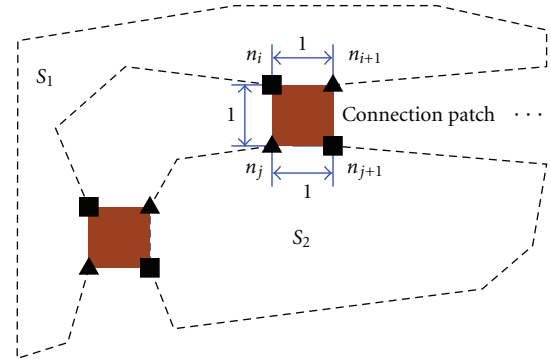


Figure 5: Connection patches for connecting two structures.

Note that, from Lemma 9(ii), at least one header node must be active for covering A_c. Thus, together with Lemma 9(i), an MUDC of A_c will allow each variable of a clause to be assigned *true* or *false*, based on the polarity of active nodes in its corresponding partition, for the clause being satisfied.

4.2. Composite Structures. Next, we illustrate how structures may be connected together to form a complex structure. As shown in Figure 5, two structures, $S_1 = (A_1, N_1)$ and $S_2 = (A_2, N_2)$, are connected via several 1×1 squares called *connection patches*. Each connection patch has two nodes from each structure, for example, $n_i, n_{i+1} \in N_1$ and $n_j, n_{j+1} \in N_2$, located at its vertices. (This is formally defined as precondition 25(i). For convenience sake, we use precondition 25(i) for referring to precondition (i) of Definition 25, and will use this labeling throughout this paper.) Besides, the nodes on the same edge of each connection patch have opposite polarities. (This is formally defined as preconditions 25(ii) and 26(ii).) For the sake of brevity, the formal definitions are given in Appendix C. We call the set $N_p = \{n_i, n_{i+1}\}$ a *port* of S_1, and a port is a *connected port* if there is a connection patch attaching to it. Besides, the nodes from different structures but on the same edge of a connection patch are each other's *connection counterpart*, for example, n_i and n_j. Obviously, it is easy to derive the following lemma.

Lemma 10. *A connection patch can be unit-disk covered by the same polar nodes located at its vertices, for example, $\{n_i, n_{j+1}\}$ or $\{n_{i+1}, n_j\}$ in Figure 5.*

In this NP-completeness proof, in order to preserve the partially well-behaved property, we require two structures to be in such a way, that is, *least interactively connected*, that

(1) at least one node from each connected port is active, (this is formally defined as precondition 27(i));

(2) nonconnected port nodes do not cover any point, except the vertices, of the connection patches (this is formally defined as precondition 27(ii) which states whether a connection patch can be fully covered only depends on its connected port);

(3) nonconnected port nodes of one structure do not cover any region of the other structure, (this is formally defined as precondition 28(i));

(4) the connected ports of one structure cannot cover any point, except their connection counterparts, of the other structure (this is formally defined as precondition 28(ii));

The formal definitions about the least interactive connection are also given in Appendix C.

A variable structure can use any two nearby nodes on the side of border as a port, for example, $\{n_i, n_{i+1}\}$ shown in Figure 1. An edge structure uses its endpoints as ports, indicated by the arrows in Figure 2. For an n-way connector, each partition P_i contains a port indicated by the arrows in Figure 3. The fact that it is possible to make the above structures least interactively connected via the ports described is proved in Appendix D.

We can define the composite structure in the following definition and derive several lemmas about the least interactive connection. For the sake of brevity, the complete proofs of these lemmas are given in Appendix E.

Definition 11. The structures $S_1 = (A_1, N_1)$ and $S_2 = (A_2, N_2)$ are least interactively connected via the connection patches $A_{cp,1}, A_{cp,2}, \ldots, A_{cp,T}$. One calls $S = (A_1 \cup A_2 \cup \bigcup_{1 \le t \le T} A_{cp,t}, N_1 \cup N_2)$ the *composite structure* of S_1 and S_2. Furthermore, one denotes $S = S_1 + S_2$.

Lemma 12. *Suppose $(A, N) = (A_1, N_1) + (A_2, N_2)$ and the cardinality of MUDCs of A_1 and A_2 is l_1 and l_2, respectively. If $U \subseteq (N_1 \cup N_2)$ is a UDC of A and $|U| = l_1 + l_2$, then $U \cap N_1$ and $U \cap N_2$ are MUDCs of A_1 and A_2, respectively.*

Lemma 13 (Connection Lemma). *Consider $S_1 = (A_1, N_1)$, $S_2 = (A_2, N_2)$, and $S = S_1 + S_2$ via one connection patch at the ports $N_{p_1} \subseteq N_1$ and $N_{p_2} \subseteq N_2$. (We note that this lemma may be extended to more than one connection patches with possible minor modification on Definition 28(ii); however we do not use this property and therefore ignore it.) Furthermore, for any $N_1' \subseteq N_1$ with $N_{p_1} \subseteq N_1'$ and $N_2' \subseteq N_2$ with $N_{p_2} \subseteq N_2'$, S will be partially well behaved on $N_1' \cup N_2'$ if the following connection preconditions hold.*

(i) *S_1 and S_2 are partially well behaved on N_1' and N_2', respectively.*

(ii) *There exist an MUDC of A_1, $U_1' \subseteq N_1$, and an MUDC of A_2, $U_2' \subseteq N_2$, such that $U_1' \cap N_1'$ and $U_2' \cap N_2'$ have the same polarity. (Since $U_1' \cap N_1'$ and $U_2' \cap N_2'$ have the same polarity, by Lemma 10, the connection patch is automatically covered without the help of the nodes not in U_1' and U_2'. Hence, $U_1' \cup U_2'$ is an MUDC of $A_1 \cup A_2 \cup A_{cp}$.)*

After describing the properties of the least interactive connection, we describe how these structures may be connected to encode G_B.

Figure 6 illustrates how edge structures are connected to a variable structure with connection patches. The edges may go up or down from the variable structure.

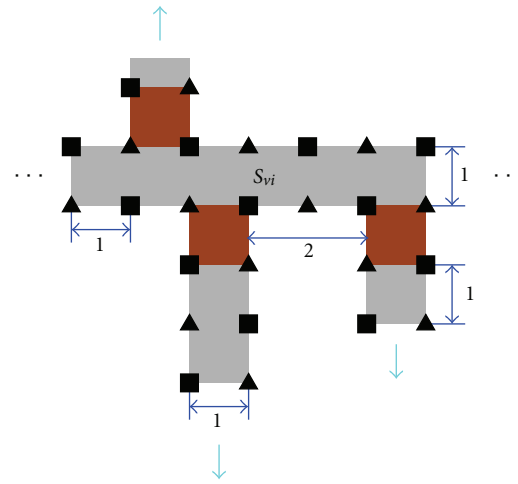


Figure 6: The structure represents a variable v_i with two edges going down and one edge going up. Note that the distance of two nearby ports on the same side must be at least 2 to ensure the least interactive connection.

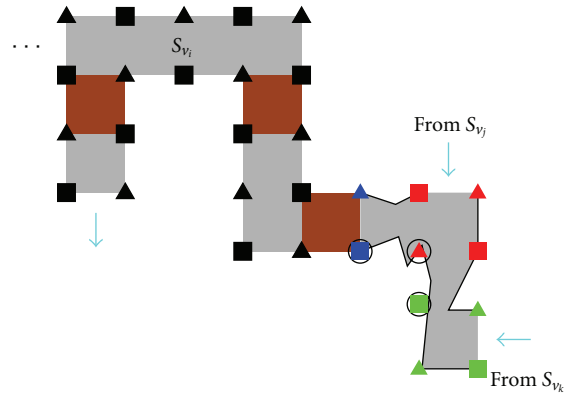


Figure 7: The variable structure S_{v_i} is connected to the positive partition of the 3-way connector, which represents the clause containing two positive literals, v_i and v_k, and one negative literal, v_j. Note that the 3-way connector has two positive partitions and one negative partition. The header nodes are indicated by squares or triangles surrounded by circles.

With n-way connectors described above, we can transform an edge and its ends in G_B by connecting a variable structure to a partition of an n-way connector via an edge structure. For example, suppose a clause is $(v_i + v_j + v_k)$, then we need a 3-way connector with two positive partitions and one negative partition. As shown in Figure 7, we connect the edge from S_{v_i} to one positive partition, the edge from S_{v_j} to the negative partition, and the edge from S_{v_k} to the other positive partition.

Since all structures are connected via 1×1 squares, the positions of structures must match, that is, all structures can be placed in a 2D space such that their ports are on a 2D grid with resolution 1×1. Obviously, such a placement is possible for variable and edge structures. Furthermore, as indicated in

Tables 1 and 2, the relative positions of the ports of each n-way connector are integers, so it is also possible to make such a placement for n-way connectors.

Note that, in addition to positions, polarities of ports must also match. (Precondition 26(ii) needs to be satisfied.) However, for example, it is possible that the polarities may not match when an edge structure is connected to a connector as shown in Figure 8(a).

Therefore, referring to Figure 9, we introduce a structure called a *polarity inverter* and denoted as $S_p = (A_p, N_p)$ to invert the polarity. S_p contains a main structure, 9×1 rectangles, and two buffers, 1×1 squares. (The purpose of the buffers is to ensure S_e satisfies the requirement that nonconnected port nodes do not cover any point, except the vertices, of the connection patches, that is, precondition 27(ii).) Similar to edge structures, a polarity inverter use its endpoint pairs, that is, $\{n_i, n_{i+1}\}$ and $\{n_j, n_{j+1}\}$ in Figure 9, as ports.

The distance between two nearby column-pairs of the main structure is 9/10. Thus, the polarities are inverted compared with a normal edge of the same length. With the polarities inverted, the polarity requirements for vertices of a connection patch can be satisfied as shown in Figure 8(b).

Of course, the structure $S_p = (A_p, N_p)$ also satisfies the following lemma. The proof is similar to variable structures and is given in Appendix F.

Lemma 14. *The structure $S_p = (A_p, N_p)$ shown in Figure 9 is well aligned and well behaved.*

Definition 15. For a given variable structure S_v, an n-way connector S_c is called an *associated connector* of S_v if S_c is connected to S_v via an edge structure. Furthermore, the partition P_i of S_c where S_v is connected to is called an *associated partition* of S_v. Similarly, the edges, polarity inverters, and connection patches used to connect S_v and S_c are called *associated connection edges*, *associated polarity inverters*, and *associated connection patches* of S_v, respectively.

For the ith variable, v_i, let

NV_i be the set of nodes in the corresponding variable structures S_{v_i},

NE_i be the set of nodes in all the associated edge structures of S_{v_i},

NI_i be the set of nodes in all the associated polarity inverters of S_{v_i},

NC_i be the set of nodes in all the associated n-way connectors of S_{v_i},

NP_i be the set of nodes in all the associated partitions of S_{v_i},

AV_i be the shaded region of S_{v_i},

AE_i be the union of the shaded region from all the associated edge structures of S_{v_i},

AI_i be the union of the shaded region from all the associated polarity inverters of S_{v_i},

AC_i be the union of the shaded region from all the associated n-way connectors of S_{v_i},

ACP_i be the union of the shaded region from all the associated connection patches of S_{v_i}.

Definition 16. Let $N_i = NV_i \cup NE_i \cup NI_i \cup NC_i$, $A_i = AV_i \cup AE_i \cup AI_i \cup AC_i \cup ACP_i$, and $T_i = (A_i, N_i)$. We call the composite structure T_i the *territory* of the ith variable. Furthermore, let $N_i' = NV_i \cup NE_i \cup NI_i \cup NP_i$. The nodes in N_i' are the *pieces* of the ith variable. Note that $NP_i \cap NP_j = \emptyset$ if $i \neq j$, and thus $N_i' \cap N_j' = \emptyset$ if $i \neq j$.

Note that T_i represents the ith variable and all clauses it belongs to as shown in Figure 10. It is not difficult to layout each structure on the plane and make variable structures and connectors far enough to prevent unwanted interactions between nearby structures, that is, structures are least interactively connected. Consequently, we have the following lemma which states that, for a given variable, its territory is partially well behaved on the set of its pieces.

Lemma 17. *If all the associated structures of the variable structure S_{v_i} are least interactively connected, the territory T_i is partially well-behaved on N_i'.*

Proof. Since structures are least interactively connected, this lemma can be proved by Lemmas 8, 9(i), 14, and Connection Lemma. The fact that $2 \leq |c| \leq 3$ for each clause c is the key to ensure precondition (ii) of Connection Lemma is satisfied for Connection Lemma being applicable. The complete proof is given in Appendix G.

After introducing the structures and their properties, we can define an equivalent MUDC problem with the geometry, MUDC(B), for a given boolean formula, B, in P3SAT by replacing the variables, clauses, and edges of the bipartite graph G_B with their corresponding structures. Denote that

NV is the set of nodes in all the variable structures,

NE is the set of nodes in all the edge structures,

NI is the set of nodes in all the polarity inverters,

NC is the set of nodes in all the n-way connectors,

AV is the union of the shaded region from all the variable structures,

AE is the union of the shaded region from all the edge structures,

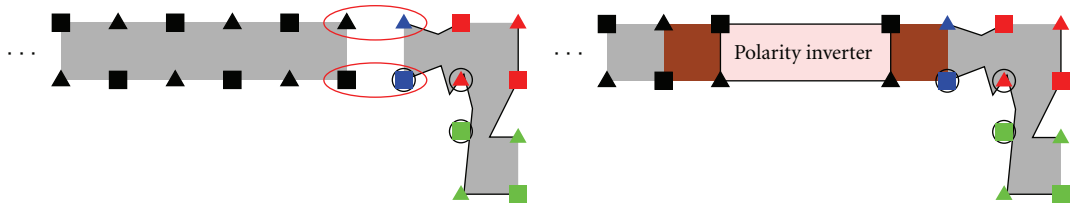
AI is the union of the shaded region from all the polarity inverters,

AC is the union of the shaded region from all the n-way connectors,

ACP is the union of the required connection patches.

Let MUDC(B) $= (A_B, N_B)$ with $A_B = AV \cup AE \cup AI \cup AC \cup ACP$ and $N_B = NV \cup NE \cup NI \cup NC$, and $K = |N_B|/2$. Hence, we have the following claim. Note that it is not difficult to prove that the construction from B to MUDC(B) can be done in polynomial time. □

Claim B is satisfiable if and only if MUDC(B) has a UDC with cardinality K.



(a) The edge's polarities do not match with the connector's (b) The polarities are inverted to match with the connector's

Figure 8: A polarity inverter structure may be needed to invert the polarity of the edge to connect the edge and 3-way connector.

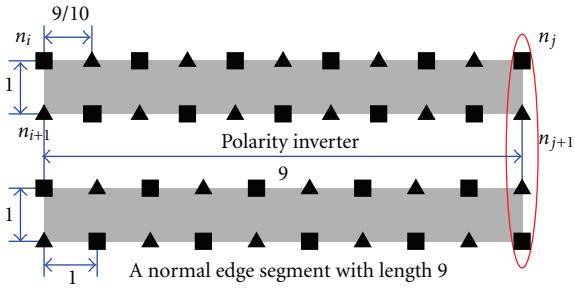


Figure 9: A structure for inverting the polarity of the edge. After distance 11, the polarities are inverted compared with a normal edge.

Note that the key to the backward direction of the proof is Lemma 17; that is, for a given variable, its territory is partially well behaved on the set of its pieces. Lemma 17 will be used to derive, that if MUDC(B) has a UDC with cardinality K, then, for each variable, half of its pieces with same polarity needs to be active to unit-disk cover its territory. Therefore, each variable could be assigned *true* or *false* based on the polarity of its active pieces.

4.3. Proof of the Claim

Proof. $\Rightarrow$ For each N_i', choose the nodes with the polarity which is the same as the assignment of the ith variable in a given satisfying instance of B. Obviously, only $|N|/2 = K$ nodes are picked. By Lemmas 8, 14, and 10, AV, AE, AI, and ACP are covered. Furthermore, since B is satisfiable, at least one header node of each n-way connector is active. By Lemma 9(iii), AC is covered. Thus MUDC(B) has a UDC with cardinality K.

$\Leftarrow$ Let MUDC(B) have a UDC $U \subseteq N$ with $|U| = K$. We will show that this set must look right.

From Lemmas 8, 9(i), and 14, we know that the cardinality of an MUDC of a variable structure, edge structure, n-way connector, or polarity inverter is half the number of the nodes in the structure. Since all structures in MUDC(B) are least interactively connected and $|U| = |N|/2$, it is not difficult to derive that, for the ith variable structure, $U \cap N_i$ is an MUDC of T_i by removing nonassociated edge structures of T_i one by one and Lemma 12.

From Lemma 17 and $N_i' \subseteq N_i$, $U \cap N_i'$ only contains the nodes with the same polarity. Thus, the ith variable could be assigned *true* or *false* based on the polarity of the nodes in $U \cap N_i'$. Finally, since at least one header node of each n-way connector must be active from Lemma 9(ii), the corresponding clause will be *true*. □

5. Extensions of MUDC

MUDC can be easily extended to the following two more general cover problems, which require each location to be unit-disk covered by predefined number of nodes. These problems regard the quality of various services of sensor network applications such as surveillance, object tracking, and fault tolerance.

Problem 18 (Minimum Unit-Disk k-Cover, MUDKC). Given a geometry (A, N) and two positive integers k and K, determine whether there is a subset $U \subseteq N$ with $|U| \leq K$ such that for all $x \in A, |\{u \in U \mid x \in \text{disk}(u)\}| \geq k$; that is, x is unit-disk covered by at least k nodes in U.

Problem 19 (Minimum Unit-Disk Multicover, MUDM). Given a geometry (A, N), a quality of surveillance function $q : A \to \mathbb{Z}^+$, and a positive integer K, determine whether there is a subset $U \subseteq N$ with $|U| \leq K$ such that for all $x \in A, |\{u \in U \mid x \in \text{disk}(u)\}| \geq q(x)$; that is, x is unit-disk covered by at least $q(x)$ nodes in U.

We may also consider connectivity and have the following problem.

Problem 20 (Minimum Connected Unit-Disk Cover, MCUDC). Given a geometry (A, N), a positive number $R_c \in \mathbb{R}^+$, and a positive integer K, determine whether there is a subset $U \subseteq N$ with $|U| \leq K$ such that $A \subseteq \text{disk}(U)$ and the graph $G_c = \{U, E_c\}$ is connected. Here $E_c = \{(n, n') \mid d(n, n') \leq R_c\}$.

Furthermore, under many environmental data sampling applications, instead of full coverage, a predefined percentage of coverage is required for achieving energy efficiency and preciseness of sampling. The objective of the following problem is to find as few nodes as possible to achieve the coverage requirements.

Problem 21 (Minimum Unit-Disk Partial Cover, MUDPC). Given a geometry (A, N), a positive number r with $0 \leq r \leq 1$, and a positive integer K, determine whether there is a subset

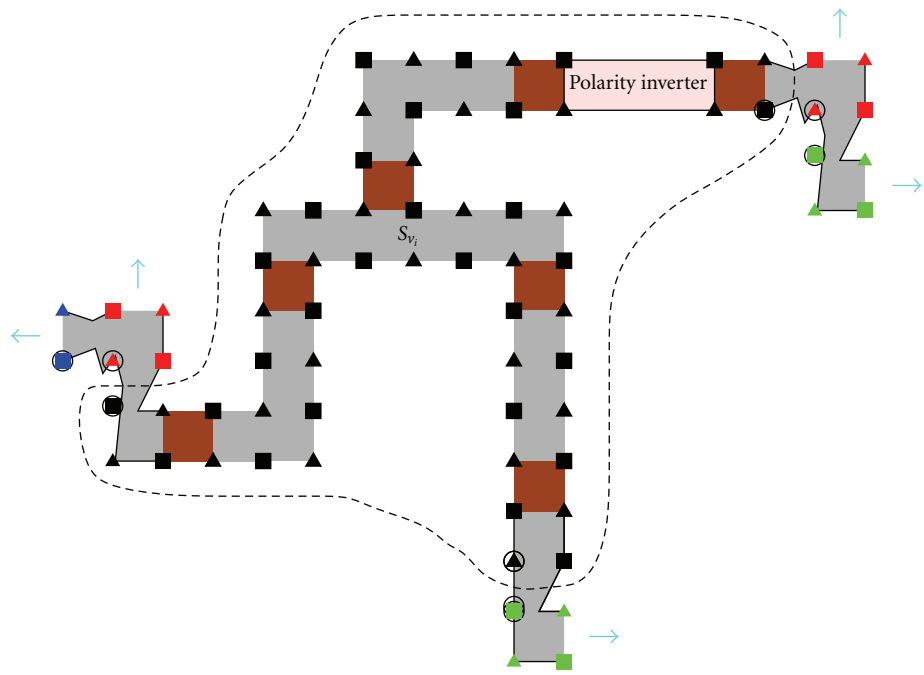


Figure 10: The territory, T_i, of the variable v_i includes variable structure S_{v_i} and all associated edge structures, patches, inverters, and connectors. The nodes enclosed in dashed line are pieces of v_i.

$U \subseteq N$ with $|U| \leq K$ such that $(\text{area}(\text{disk}(U)))/\text{area}(A) \geq r$. Here the function area$(\cdot)$ gives the area of a given region.

By the NP-completeness of MUDC, it is easy to derive the complexity of the above problems.

Corollary 22. *MUDKC, MUDM, MCUDC, and MUDPC are NP-complete.*

Proof. Note that every instance of MUDC can be viewed as an instance of MUDKC, MUDM, or MUDPC simply by letting $k = 1$, $q(x) = 1$ for all $x \in A$, or $r = 1$, respectively. Thus MUDC is just a restricted version of these problems, and their NP-completeness follows by trivial transformations from MUDC.

For the geometry, MUDC(B), described in the NP-completeness proof of MUDC, it is obvious that, for any UDC of A_B, the distance between an active node and its closest active node is less than 2. Thus, it is not difficult to prove that if $R_c \geq 2$, G_c is connected. That is, MUDC is just a restricted version of MCUDC with $R_c \geq 2$, and the NP-completeness of MCUDC follows by a trivial transformation from MUDC. □

6. Arc Sampling Algorithm for Reducing MUDC to MSC

As stated earlier, MUDC may be solved by partitioning the region A into disjoint sectors [20]. Consequently, MUDC is reduced to MSC and many well-known algorithms can be applied, for example, the greedy algorithm is the best approximation algorithm and the approximation factor is well known.

To identify necessary sectors is a key factor to whether the solutions found by the algorithms for the transformed MSC are valid, that is, the solutions are disk covers of the original MUDC. A naive approach for partitioning is to sample A at uniform spacings in a grid pattern; then the sampling points covered by the same set of nodes would be grouped into one sector. With enough resolution, all necessary sectors can be successfully identified at the expense of computation time.

However, to determine a good resolution may be difficult. For example, Figure 11 shows that inappropriately increasing resolution may not necessarily find a valid solution, and Figure 13(a) illustrates that the ratio of successfully finding a valid solution decreases as the node density decreases. Therefore, we propose an arc sampling approach which is inspired by the theorem of the paper [32], that is, A is covered if and only if the perimeter of each node's sensing region is covered. (Several special cases including boundary are also discussed in [32].)

Consider the node n with its neighbors, that is, the nodes with distance not greater than 2 from n. As illustrated in Figure 12, n's perimeter is divided into disjoint arcs by its neighbors' perimeters and the boundary of A. It is obvious that all points of each disjoint arc in A are covered by the same set of nodes. Thus, we can simply choose a point such as the midpoint from each arc in A, for example, χ_2 from arc $\widehat{\alpha_2\alpha_3}$, as a sampling point. Note that if n's perimeter cannot be divided, n's perimeter (and thus A) is not covered [32], as indicated in Lines 7~9 of Algorithm 1. From the earlier mentioned theorem of the paper [32], it is easy to derive that A is covered if and only if all these sampling points are covered. Thus, the solutions of the MSC transformed by this arc sampling approach are always valid.

The arc sampling algorithm is shown in Algorithm 1. Here $P(n) = \{x \mid d(n,x) = 1\}$ is the perimeter of n. The outer loop between Lines 2 and 15 will run $|N|$ times. The average time complexity for a node to find all its perimeter intersections with neighbors, that is, Lines 4∼6, is $O(4\pi d)$. Here $d = |N|/\text{area}(A)$ is the density of nodes. Note that each node has average $4\pi d$ neighbors and thus $8\pi d$ disjoint arcs. Hence, the sorting in Line 10 could be implemented in $O(8\pi d \cdot \log 8\pi d) = O(d \log d)$ time. In addition, the time complexity of finding the midpoints, that is, Lines 11∼14, is $O(8\pi d)$. Thus the overall time complexity of Algorithm 1 is $O(|N|d \log d)$ or $O(\text{area}(A) \cdot d^2 \log d)$. On the other hand, it is not difficult to derive that the time complexity of finding all sampling points is $O(\text{area}(A)/a)$ for the grid sampling approach. Here a is the area of each grid. Together with Figure 13(a), we may conclude that the arc sampling approach will perform more efficiently than the grid sampling approach, particularly with low density of nodes.

We conducted an experiment to compare the grid sampling and the arc sampling approaches. In the experiment, there are 240 nodes deployed uniformly in a square region ranging from 30×30 to 75×75. The radius of the sensing range is 10. The sampling interval of the grid sampling approach is 0.1. After transforming to MSC, the greedy algorithm is used to find the approximated solution. Each result is the average of 100 random deployments.

Figure 13(b) shows the effectiveness of the arc sampling approach. The solutions from both approaches have almost same sizes, that is, same number of nodes. However, Figure 13(a) shows that not every solution obtained from the grid sampling approach is valid. Figure 13(c) illustrates that the arc sampling approach requires less computation time to reduce MUDC to MSC than the grid sampling approach except the densest deployment. Thus, the arc sampling approach is effective and efficient for reducing MUDC to MSC.

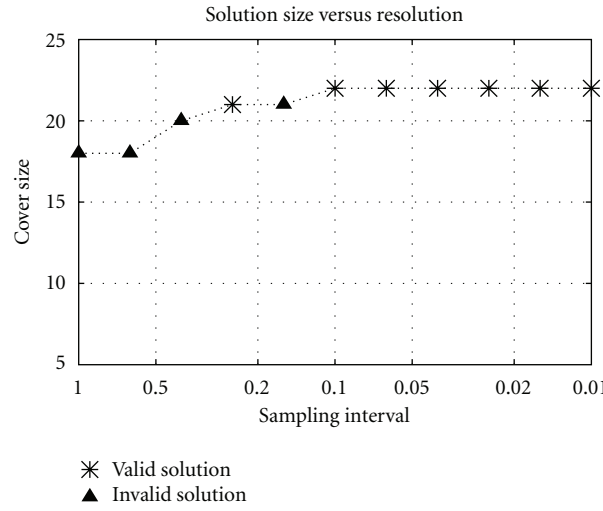


Figure 11: The solution found at different sampling intervals. Increasing resolution may not necessarily find a valid solution.

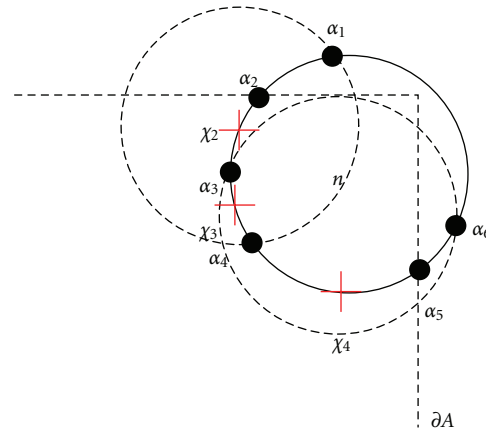


Figure 12: Each arc in A, that is, $\widehat{\alpha_2\alpha_3}$, $\widehat{\alpha_3\alpha_4}$, or $\widehat{\alpha_4\alpha_5}$, is covered by the same set of nodes. Hence, we can simply use a point, for example, the midpoint marked as a cross, from each arc as a sampling point. Here ∂A represents the boundary of A.

7. Decentralized Polynomial Approximation Algorithms

Algorithm 1 and the greedy algorithm may not be suitable for all practical sensor network applications, since it is a centralized algorithm at the cost of potentially excessive communication across the whole network and communication accounts for the majority of energy consumption. The communication power consumption increases with number of nodes and internode distances, so it is not well scalable. Unless the nodes involving in the communication and computation have enough resources, the algorithm may not complete successfully.

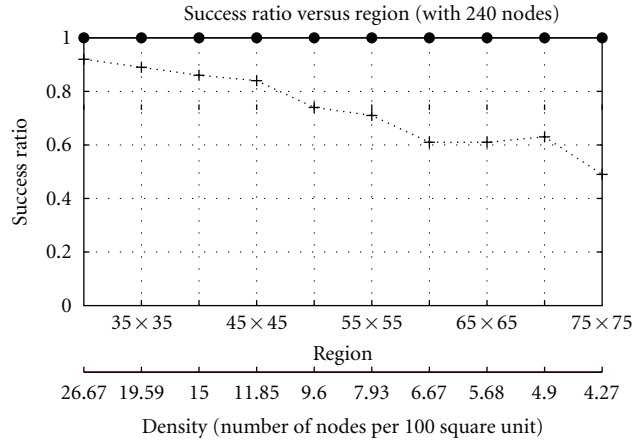
Thus, we present a decentralized algorithm in which nodes only require local information by using the *divide and conquer* technique described in [42] and derive its approximation factor. Furthermore, if the maximum node density is fixed, we may design a constant approximation factor algorithm by using the similar technique. (Note that MUDC remains NP-complete even with fixed maximum node density. It could be easily proved from the fact that the density of MUDC(B) in the NP-completeness proof of MUDC is bounded by a constant.)

7.1. Decentralized Greedy Approximation Algorithm. The proposed algorithm for the instance (A, N) proceeds as follows.

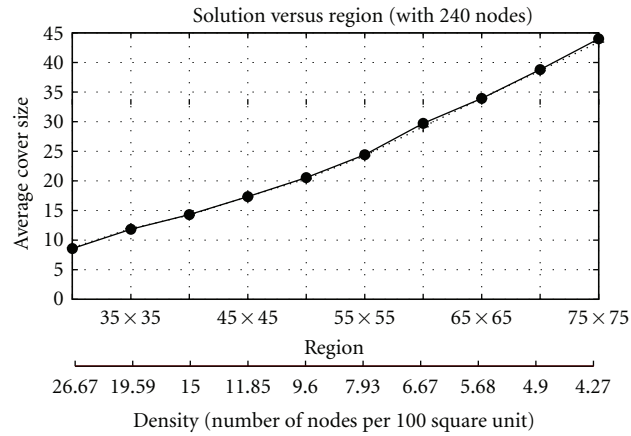
(1) Use Algorithm 1 to determine sampling points.

(2) Divide A into vertical strips with width being the diameter of the sensing region, that is, 2. Each strip is left closed and right open. Number strips from left to right. There are total I strips.

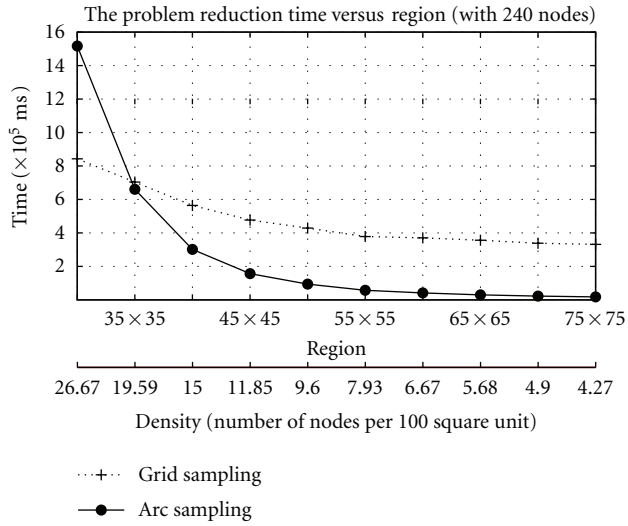
(3) Divide each strip into cells with length being the diameter of the sensing region. Each cell is bottom closed and top open. Number cells from bottom to top. Denote the jth cell of the ith strip as $C_{i,j}$. There are total J cells for each strip.



(a) The ratio of successfully finding a solution with different node density: the greedy algorithm with the arc sampling approach always generates a valid solution



(b) The arc sampling approach is as effective as the grid sampling approach



(c) The arc sampling approach requires less computation time to reduce MUDC to MSC than the grid sampling approach except the densest deployment

Figure 13: The experiment results show that the arc sampling approach is effective and effcient for reducing MUDC to MSC.

(4) Apply the greedy algorithm to each cell, that is, select a node that covers the maximum number of uncovered sampling points in the cell. Denote the solution of $C_{i,j}$ as $\mathrm{SOL}_{C_{i,j}}$.

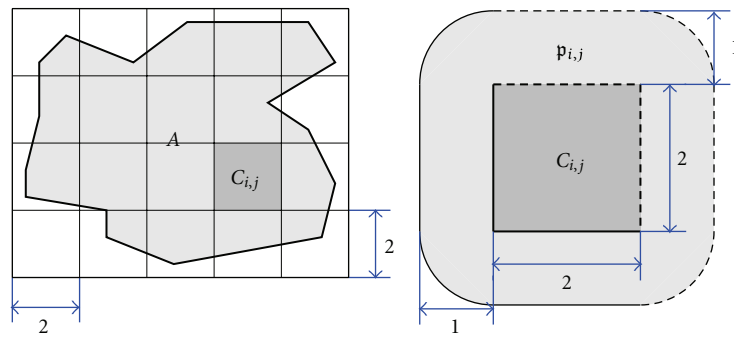
(5) Output the solution $\mathrm{SOL}_A = \bigcup_{\substack{1\le i\le I\\ 1\le j\le J}} \mathrm{SOL}_{C_{i,j}}$.

Figure 14(a) illustrates that A is divided into 2×2 cells. Note that each sampling point is located in exactly one cell. This algorithm requires that geometric information of A and cells are known a priori to each node and each node's location can be determined after deployment.

For each cell $C_{i,j}$, we define its *repository* $\mathfrak{p}_{i,j} = \{x \mid \exists y \in C_{i,j}, d(x, y) \le 1\}$. As illustrated in Figure 14(b), $\mathfrak{p}_{i,j}$ is the region containing all nodes that may cover the sampling points in $C_{i,j}$. Hence, in Step 4, the greedy algorithm is applied to the nodes in each cell's repository and can be implemented in $O(n_{rp}^2 \log n_{rp})$, where n_{rp} is the maximum number of nodes in a repository. Furthermore, Figure 14(b) also illustrates that a node does not need to communicate with others further than $(2 + 2\sqrt{2})$ times of the sensing radius. Thus, this approach is more scalable than the centralized greedy algorithm.

Theorem 23. *The above algorithm has an approximation factor* $4O(\log m)$. *(Though* $4O(\log m)$ *can be written as* $O(\log m)$ *by definition, we explicitly write it out to emphasize the approximation factor of the decentralized algorithm is four times the approximation factor of the centralized algorithm.) Here* m *is the maximum number of sampling points covered by a single node.*

Proof. The theorem is the result of *the shifting lemma* in [42]. The proof proceeds as follows.



(a) (b)

Figure 14: Divide and conquer: (a) The monitored region A (enclosed in heavy line) is divided into 2×2 cells. (b) $C_{i,j}$ (enclosed in heavy line) and its repository $\mathfrak{p}_{i,j}$ (enclosed in thin line)—the solid lines and dashed lines represent closed and open boundaries. Note that the distance between any two points in $\mathfrak{p}_{i,j}$ does not exceed $(2 + 2\sqrt{2})$.

For $A' \subseteq A$, denote that $\mathrm{OPT}_{A'}$ is the optimum solution to cover the sampling points in A'. That is, $\mathrm{OPT}_{C_{i,j}}$ is the optimum solution for $C_{i,j}$, $\mathrm{OPT}_{\bigcup_j C_{i,j}}$ is the optimum solution for the ith strip, and so forth. Thus, from Step 4, $|\mathrm{SOL}_{C_{i,j}}| \leq \mathrm{O}(\log m_{i,j}) \cdot |\mathrm{OPT}_{C_{i,j}}|$. Here $m_{i,j}$ is the maximum number of sampling points in $C_{i,j}$ covered by a single node.

Consider the ith strip, and define the following disjoint subsets of $\mathrm{OPT}_{\bigcup_j C_{i,j}}$:

$\mathrm{OPT}^{(j)}$ be the set of nodes that only cover the sampling points in $C_{i,j}$,

$\mathrm{OPT}^{(j,j+1)}$ be the set of nodes that cover both the sampling points in $C_{i,j}$ and $C_{i,j+1}$.

Note that since the length of cells is the diameter of the sensing region, the union of the above disjoint subsets is $\mathrm{OPT}_{\bigcup_j C_{i,j}}$. Hence, $|\mathrm{OPT}_{\bigcup_j C_{i,j}}| = \sum_{1\leq j\leq J} |\mathrm{OPT}^{(j)}| + \sum_{1\leq j\leq J-1} |\mathrm{OPT}^{(j,j+1)}|$. Besides, it is obvious that $\mathrm{OPT}^{(j-1,j)} \cup \mathrm{OPT}^{(j)} \cup \mathrm{OPT}^{(j,j+1)}$ covers all sampling points in $C_{i,j}$. (Here $\mathrm{OPT}^{(0,1)} = \emptyset$ and $\mathrm{OPT}^{(J,J+1)} = \emptyset$.) Thus, $|\mathrm{OPT}_{C_{i,j}}| \leq |\mathrm{OPT}^{(j-1,j)} \cup \mathrm{OPT}^{(j)} \cup \mathrm{OPT}^{(j,j+1)}| = |\mathrm{OPT}^{(j-1,j)}| + |\mathrm{OPT}^{(j)}| + |\mathrm{OPT}^{(j,j+1)}|$. Therefore, it can easily be derived that

$$\begin{aligned} \sum_{1\leq j\leq J} \left|\mathrm{OPT}_{C_{i,j}}\right| &\leq \left|\mathrm{OPT}_{\bigcup_j C_{i,j}}\right| \\ &\quad + \sum_{1\leq j\leq J-1} \left|\mathrm{OPT}^{(j,j+1)}\right| \qquad (1) \\ &\leq 2 \cdot \left|\mathrm{OPT}_{\bigcup_j C_{i,j}}\right|. \end{aligned}$$

□

Similarly, it can easily be derived that

$$\sum_{1\leq i\leq I} \left|\mathrm{OPT}_{\bigcup_j C_{i,j}}\right| \leq 2 \cdot |\mathrm{OPT}_A|, \qquad (2)$$

and then

$$\sum_{\substack{1\leq i\leq I \\ 1\leq j\leq J}} \left|\mathrm{OPT}_{C_{i,j}}\right| \leq 4 \cdot |\mathrm{OPT}_A|. \qquad (3)$$

Consequently,

$$\begin{aligned} |\mathrm{SOL}_A| &\leq \sum_{\substack{1\leq i\leq I \\ 1\leq j\leq J}} \left|\mathrm{SOL}_{C_{i,j}}\right| \\ &\leq \sum_{\substack{1\leq i\leq I \\ 1\leq j\leq J}} \mathrm{O}\left(\log m_{i,j}\right) \left|\mathrm{OPT}_{C_{i,j}}\right| \\ &\leq \mathrm{O}(\log m) \cdot \sum_{\substack{1\leq i\leq I \\ 1\leq j\leq J}} \left|\mathrm{OPT}_{C_{i,j}}\right| \\ &\leq 4\mathrm{O}(\log m) \cdot |\mathrm{OPT}_A|. \end{aligned} \qquad (4)$$

Note that, obviously, $\max_{i,j}\{m_{i,j}\} \leq m$.

7.2. Constant Approximation Factor Algorithm with Fixed Maximum Density. When the maximum node density, denoted as d, is fixed, the similar divide and conquer technique can be used to derive a constant approximation factor algorithm.

The algorithm is almost the same as the previous one except Step 4, which will be modified as follows:

(4) Apply an exhaustive search for optimum solution to each cell. Denote the solution of $C_{i,j}$ as $\mathrm{SOL}_{C_{i,j}}$.

Theorem 24. *The above algorithm has a constant approximation factor 4.*

Proof. Note that the number of nodes in each cell's repository is at most $\lceil \mathrm{area}(\mathfrak{p}_{i,j}) \cdot d \rceil = \lceil (12+\pi)d \rceil$. (Refer to Figure 14(b); the area of each repository is $(12 + \pi)$.) Thus, the time complexity of the exhaustive search is at most $2^{\lceil (12+\pi)d \rceil}$ for each cell. Since d is fixed, an optimum solution for each cell can be found with a constant time complexity. Since $|\mathrm{SOL}_{C_{i,j}}| = |\mathrm{OPT}_{C_{i,j}}|$, from (3), we have

$$|\mathrm{SOL}_A| \leq \sum_{\substack{1\leq i\leq I \\ 1\leq j\leq J}} \left|\mathrm{SOL}_{C_{i,j}}\right| \leq 4 \cdot |\mathrm{OPT}_A|. \qquad (5)$$

□

7.3. Performance Evaluation. We conducted various simulations to evaluate the proposed algorithms. In Figure 15,

Input: (A, N)
Output: a set Ξ of sampling points
(1) $\Xi \leftarrow \emptyset$
(2) **for all** $n_i \in N$ **do**
(3) $\Gamma \leftarrow P(n_i) \cap \partial A$
(4) **for all** $n_j \in N$ such that $d(n_i, n_j) \leq 2$ **do**
(5) $\Gamma \leftarrow \Gamma \cup (P(n_i) \cap P(n_j))$
(6) **end for**
(7) **if** $\Gamma = \emptyset$ **then**
(8) **exit**/* A is not covered by N. */
(9) **end if**
(10) List $L \leftarrow$ the points in Γ sorted by their azimuth angles on the polar coordinate system with reference to n_i
(11) **for all** $\alpha_i \in L$ such that $1 \leq i < |\Gamma|$ **do**
(12) $\Xi \leftarrow \Xi \cup \{\chi \mid \chi$ is the midpoint of $\widehat{\alpha_i \alpha_{i+1}}$ and $\chi \in A\}$
(13) **end for**
(14) $\Xi \leftarrow \Xi \cup \{\chi \mid \chi$ is the midpoint of $\widehat{\alpha_{|\Gamma|} \alpha_1}$ and $\chi \in A\}$
(15) **end for**

Algorithm 1: Arc sampling algorithm.

nodes are deployed uniformly within a 30×30 square region. Figures 15(a) and 15(b) show the solution size and execution time for various sensing ranges in which there are 25 nodes. The optimum solution OPT is found by exhaustive searching. GRD denotes the solution by using Algorithm 1 and the greedy algorithm. deGRD and deOPT represent the algorithms described in Sections 7.1 and 7.2 respectively. The cover size decreases as the sensing radius increases, since each node can cover a larger region. GRD and deOPT have similar performance in terms of cover size and execution time. Furthermore, deGRD generates the largest cover size, on average 48% more than OPT, 22.7% more than GRD, or 21% more than deOPT, but requires the least execution time, on average 0.01% of OPT, 22.7% of GRD, or 17.8% of deOPT.

Figures 15(c) and 15(d) indicate the solution size and execution time for various number of nodes in which the sensing radius is fixed at 10. The cover size does not change significantly as the number of nodes increases, since the sensing region of each node does not change. GRD and deOPT have similar cover size, but deOPT requires more execution time than GRD. Similarly, deGRD generates the largest cover size, on average 32% more than OPT, 17% more than GRD, or 14% more than deOPT, in the least time, on average 0.017% of OPT, 16.7% of GRD, or 9.5% of deOPT.

We also considered the scenario in which nodes are deployed in a Gaussian distribution with the peak located at the center of A and the variance 15, and the results are illustrated in Figure 16. Here A is a 30×30 square region. In Figures 16(a) and 16(b), the number of nodes is 25 and the sensing radius varies between 8.5 and 12. deGRD generates the largest cover size, on average 50% more than OPT, 18.7% more than GRD, or 26% more than deOPT, but requires the least execution time, on average 0.019% of OPT, 21.7% of GRD, or 10% of deOPT.

Furthermore, in Figures 16(c) and 16(d), the sensing radius is fixed at 10 and the number of nodes varies between 12 and 30. Similarly, deGRD generates the largest cover size, on average 40% more than OPT, 13.1% more than GRD, or 20.6% more than deOPT, in the least time, on average 0.03% of OPT, 20.8% of GRD, or 8.2% of deOPT. For most of cases, deOPT has smaller cover sizes than GRD in this scenario.

8. Conclusion

In this paper, we consider the complexity of MUDC, the Minimum Unit-Disk Cover problem. This problem has applications in extending the sensor network lifetime by selecting minimum number of nodes to fully cover a geometric connected region of interest and putting the remaining nodes in power saving mode. MUDC is a restricted version of MSC where the sensing region of each node is a unit-disk and the monitored region is geometric connected, a well-adopted network model in many works of the literature.

To prove the hardness of MUDC, we construct various structures to represent variables and edges of a given P3SAT instance's bipartite graph G_B. With the well-aligned and partially well-behaved properties of these structures, we illustrate that the structures can be unit-disk covered with half of nodes. Furthermore, we introduce the n-way connectors to represent clauses, which can be unit-disk covered with half of its nodes if and only if the corresponding clauses have *true* assignments. Finally, we discuss how complex structures can be constructed by connecting simpler structures while still preserving these properties, that is, via the least interactive connection. Thus, we prove that P3SAT can be directly reduced to MUDC in polynomial time, and obtain the NP-completeness proof of MUDC.

We also discuss several optimum problems with various coverage constraints introduced by different sensing applications. These problems are extensions of MUDC, and their NP-completeness proofs are presented as a corollary.

We propose the arc sampling algorithm which may effectively and efficiently reduce MUDC to MSC, and many well-known algorithms can be applied to find approximated solutions. We also propose a decentralized algorithm with a guaranteed performance. The algorithm requires only local communication, that is, a node does not need to communicate with others further than $(2+2\sqrt{2})$ times of the sensing radius. Thus, this approach is scalable. Furthermore, we present an algorithm with a constant approximation factor 4 if the maximum node density is fixed. Finally, we provide simulation results to evaluate the proposed algorithms and the optimum algorithm in uniform and Gaussian deployment networks. The results show that deOPT may have smaller cover size than GRD at the cost of more execution time. In addition, deGRD generates the largest cover size in the least time.

Appendices

A. Proof of Lemma 8

In this appendix, we present the proof for Lemma 8.

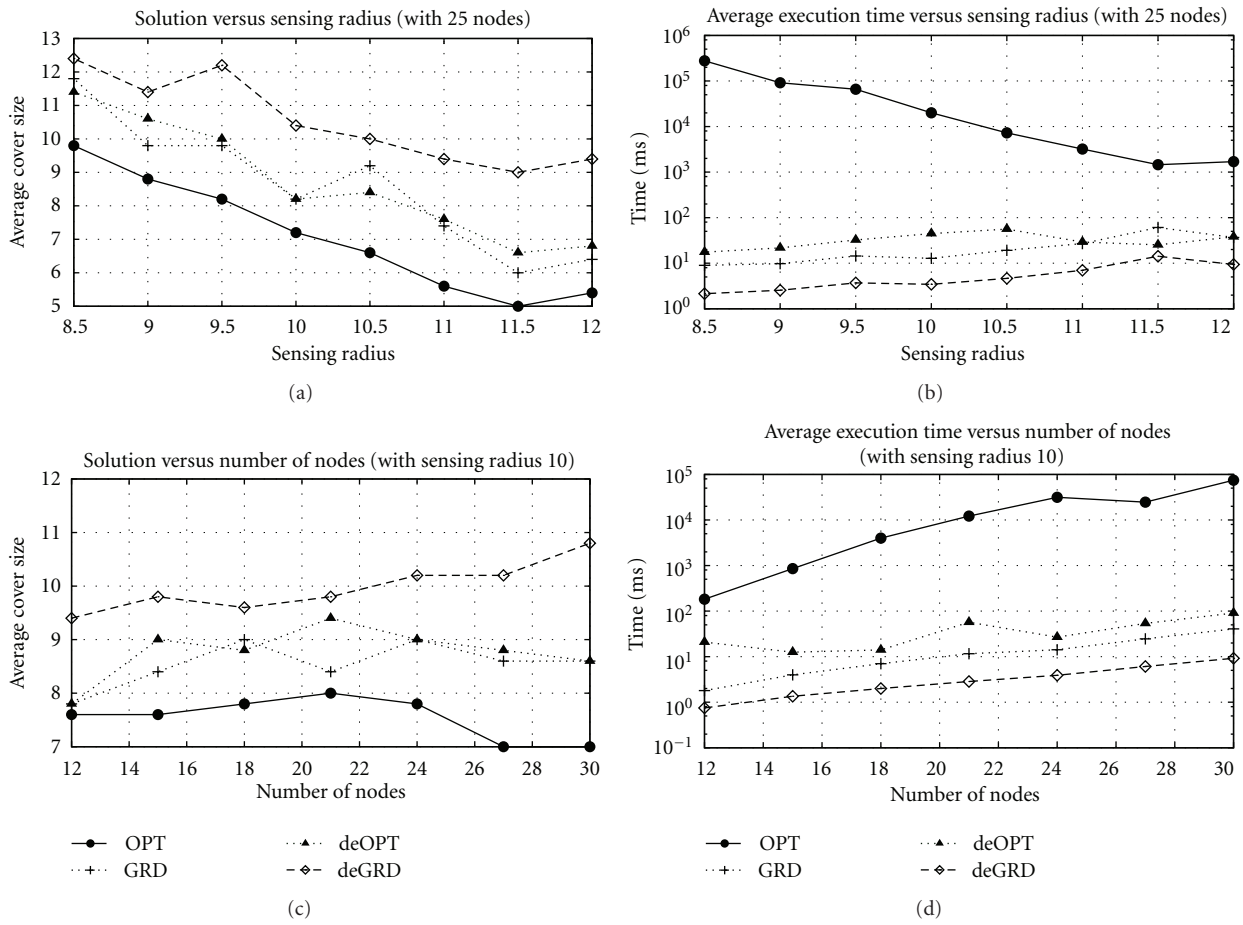


Figure 15: Performance evaluation: the nodes are deployed uniformly within a 30×30 square region.

A.1. Variable Structures. Obviously, the structure shown in Figure 1 has the same number of positive and negative polar nodes and, thus, precondition 7(i) is satisfied.

We call the pair of opposite polar nodes in the ith column the *column-pair* i. From Figure 17, it is not difficult to prove that each column-pair must have at least one active node to fully cover A_v. Since there are $d_v + 1$ columns, $|U| \geq d_v + 1 = |N_v|/2$ for a UDC U of A_c; that is, the precondition 7(ii) is satisfied. Besides, if we can pick exactly one node from each column-pair and the resulting set, say U', can-unit-disk cover A_v, then U' is an MUDC since $|U'| = d_v + 1$.

It is easy to prove that, if the picked nodes from each column-pair do not have the same polarity, A_v cannot be unit-disk covered by these $d_v + 1$ picked nodes. Suppose that the picked nodes of the ith column and the $(i+1)$th column have opposite polarities. From the Figure 18, A_v cannot be covered.

Thus, the only possibility to cover A_v with $d_v + 1$ nodes is to pick the nodes with the same polarity from each column-pair, which can be easily proved by induction. Figures 19(a) and 19(b) show the base cases of the induction and Figure 19(c) illustrates the induction step. Therefore, precondition 7(iii) is satisfied and S is well behaved. Note that the induction step works for both positive and negative cases and also proves that S is well aligned.

A.2. Edge Structures. As illustrated in Figure 20, the structure $S_e = (A_e, N_e)$ is basically a composite structure from numbers of variable structures connected via connection patches. By Lemma 29 described in Appendix D, the variable structures are least interactively connectable at any two nearby nodes on the side of border. Thus, this lemma can be proved by induction on the composing variable structures with Lemma 10 and Connection Lemma.

B. Proof of Lemma 9

In this appendix, we complete the proof of Lemma 9.

B.1. 2-Way Connectors. Figures 3(a) and 4(a) illustrate the labels of nodes and vertices of A_c for the 2-way connector. Table 1 lists the positions of nodes and vertices relative to n_1. Note that $\mathcal{P} = \{P_1, P_2\}$, $P_1 = \{n_4, n_5, n_6, n_7\}$, $P_2 = \{n_1, n_2, n_3, n_8\}$, and the header nodes $h_1 = n_7$ and $h_2 = n_8$. Obviously, $|P_i^+| = |P_i^-|$ for $i = 1$ and 2, and, thus, precondition 7(i) is satisfied.

As shown in Figure 21(d), only two active nodes n_4 and n_7 from P_1 cannot unit-disk cover A_c even all nodes in P_2 are active, which implies that only one node, n_4 or n_7, from P_1 being active cannot unit-disk cover A_c. Similarly from

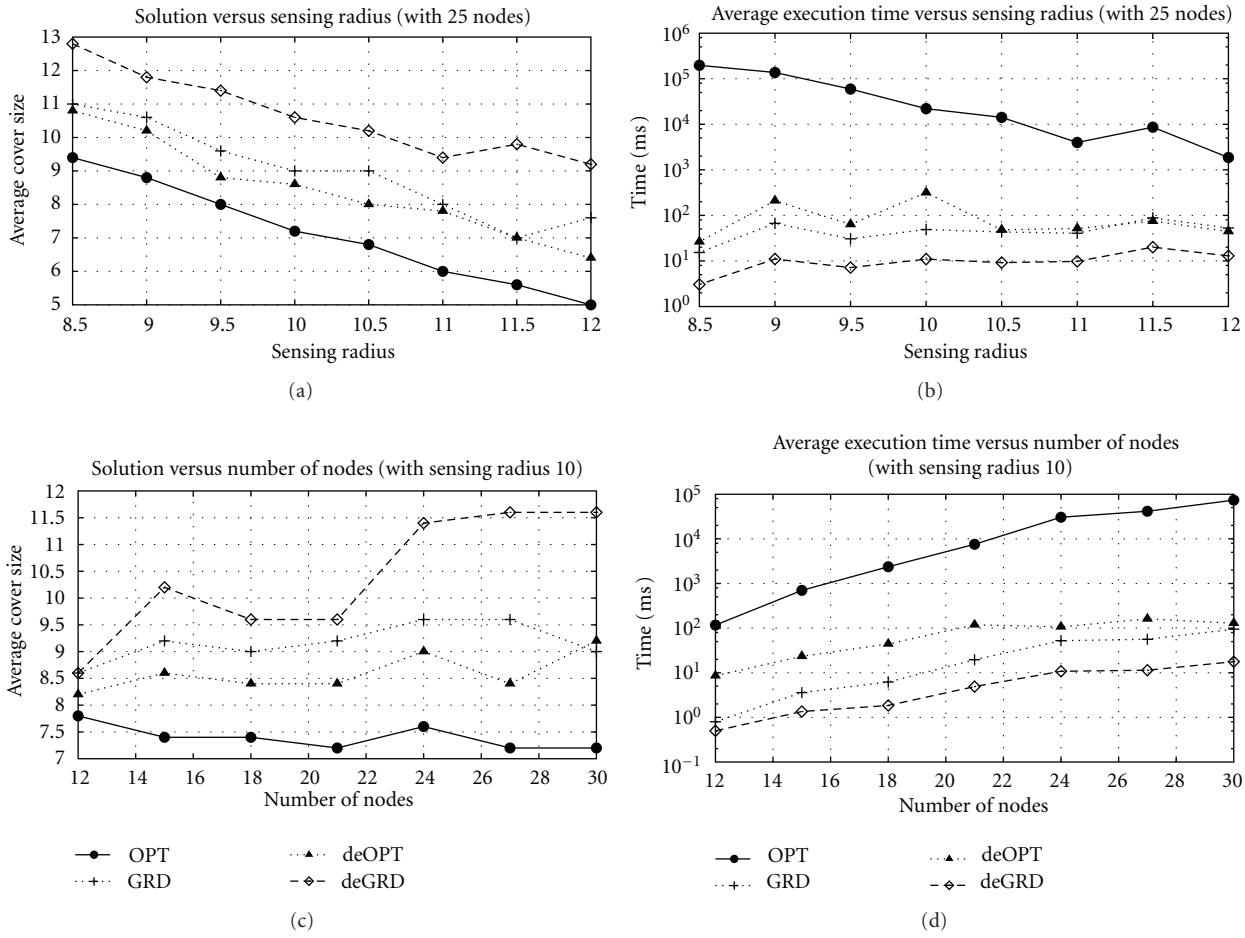


FIGURE 16: Performance evaluation: the nodes are deployed in a Gaussian distribution with the peak located at the center of A and the variance 15. Here A is a 30×30 square region.

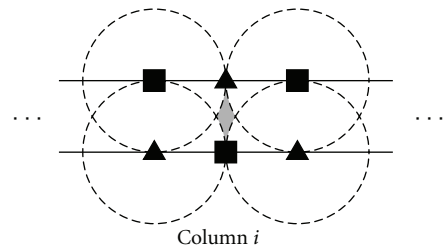


FIGURE 17: The shaded region cannot be covered while none of nodes from the column-pair i is active.

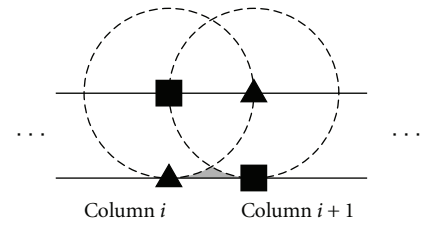


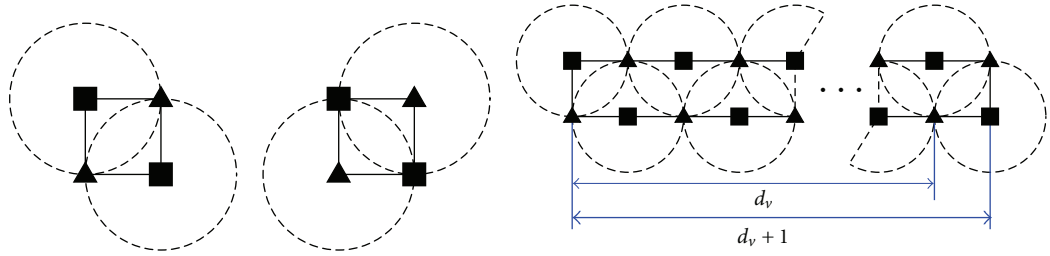
FIGURE 18: The shaded region cannot be covered by two nearby opposite polar nodes.

Figure 21(a), only one node, n_5 or n_6, from P_1 being active cannot unit-disk cover A_c. Thus, if $U \subseteq N_c$ and $|U \cap P_1| < |P_1|/2 = 2$, U is not a UDC of A_c. That is, if U is a UDC of A_c, $|U \cap P_1| \geq |P_1|/2$.

Furthermore, Figure 21 lists all possible cases in which only two opposite polar nodes from P_1 being active cannot unit-disk cover A_c even all nodes in P_2 are active.

As shown in Figure 22(d), only two active nodes n_1 and n_2 from P_2 cannot unit-disk cover A_c even all nodes in P_1 are active, which implies that only one node, n_1 or n_2, from P_2 being active cannot unit-disk cover A_c. Similarly from Figure 22(a), only one node, n_3 or n_8, from P_2 being active cannot unit-disk cover A_c. Thus, if $U \subseteq N_c$ and $|U \cap P_2| < |P_2|/2 = 2$, U is not a UDC of A_c. That is, if U is a UDC of A_c, $|U \cap P_2| \geq |P_2|/2$. Thus, together with the result from P_1, precondition 7(ii) is satisfied.

Furthermore, Figure 22 lists all possible cases in which only two opposite polar nodes from P_2 being active cannot unit-disk cover A_c even all nodes in P_1 are active.



(a) The base case with only positive polar nodes being active

(b) The base case with only negative polar nodes being active

(c) Induction step for the positive polarity case

FIGURE 19: Induction proof for that a d_v long variable structure can be unit-disk covered by $d_v + 1$ nodes with the same polarity.

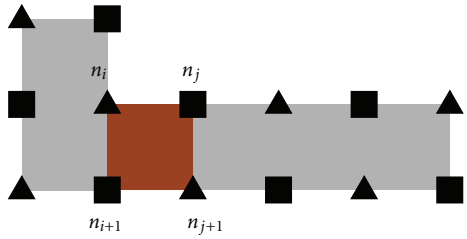


FIGURE 20: An edge is a composite structure of variable structures. Note that a variable structure can be rotated by 90° and still satisfies Lemmas 8 and 29. Here (n_i, n_{i+1}) and (n_j, n_{j+1}) are the connected ports of the connection patch.

Figure 23 illustrates possible cases in which A_c can be unit-disk covered with two same polar nodes from P_1 and two same polar nodes from P_2 being active. Together with Figures 21 and 22, precondition 7(iii) is satisfied. With satisfaction of preconditions 7(i), 7(ii), and 7(iii) for P_1 and P_2, property (i) is satisfied. Furthermore, in Figure 23, at least one of the header nodes must be active, so property (iii) is also satisfied.

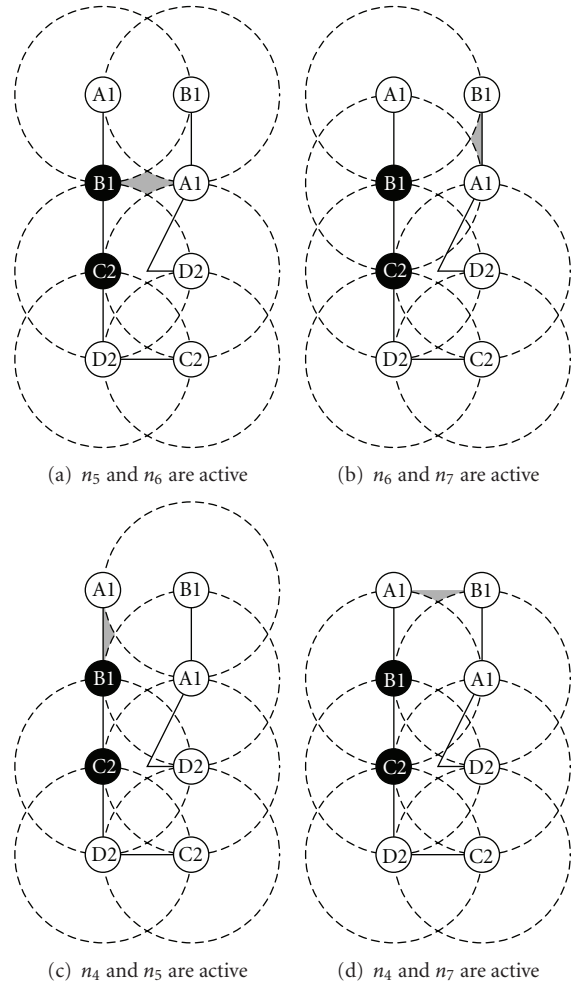
Figure 24 shows that if no header node is active, A_c cannot be unit-disk covered. That is, property (ii) is satisfied. Therefore, we prove that Lemma 9 holds for the 2-way connector shown in Figure 3(a).

B.2. 3-Way Connectors. Figures 3(b) and 4(b) illustrate the labels of nodes and vertices of A_c for the 3-way connector. Table 2 lists the positions of nodes and vertices relative to n_1. Note that $\mathcal{P} = \{P_1, P_2, P_3\}$, $P_1 = \{n_7, n_8\}$, $P_2 = \{n_4, n_5, n_6, n_9\}$, $P_3 = \{n_1, n_2, n_3, n_{10}\}$, and the header nodes $h_1 = n_8$, $h_2 = n_9$, and $h_3 = n_{10}$. Obviously, $|P_i^+| = |P_i^-|$ for $i = 1, 2$, and 3, and, thus, precondition 7(i) is satisfied.

Figure 25 shows that A_c cannot be fully unit-disk covered without any node in P_1 being active, even if all nodes in P_2 and P_3 are active. Thus, if U is a UDC of A_c, $|U \cap P_1| \geq |P_1|/2 = 1$.

As shown in Figure 26(d), only two active nodes n_4 and n_9 from P_2 cannot unit-disk cover A_c even if all nodes in P_1 and P_3 are active, which implies that only one node, n_4 or n_9, from P_2 being active cannot unit-disk cover A_c. Similarly from Figure 26(a), only one node, n_5 or n_6, from P_2 being active cannot unit-disk cover A_c. Thus, if $U \subseteq N_c$ and $|U \cap P_2| < |P_2|/2 = 2$, U is not a UDC of A_c. That is, if U is a UDC of A_c, $|U \cap P_2| \geq |P_2|/2$.

Furthermore, Figure 26 lists all possible cases in which only two opposite polar nodes from P_2 being active cannot unit-disk cover A_c even all nodes in P_1 and P_3 are active.



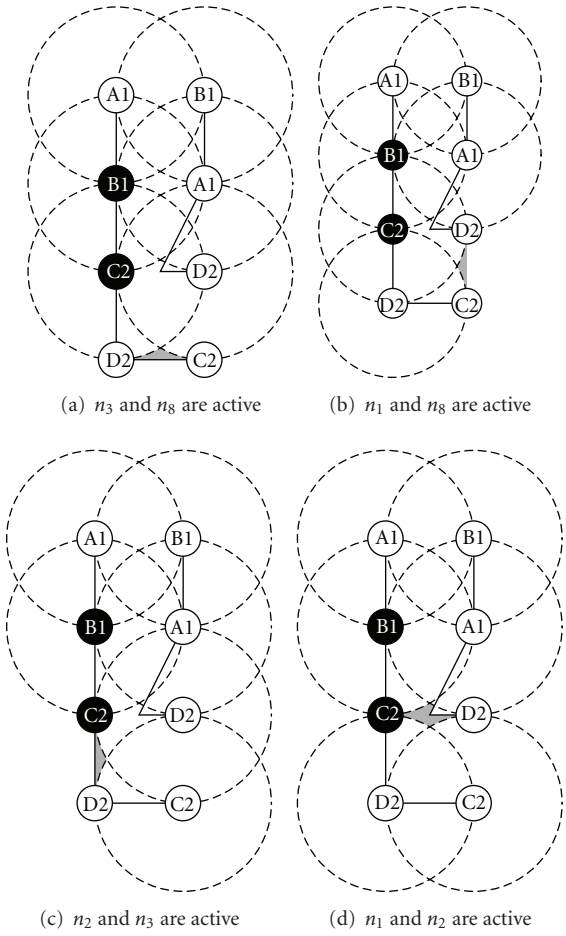
(a) n_5 and n_6 are active

(b) n_6 and n_7 are active

(c) n_4 and n_5 are active

(d) n_4 and n_7 are active

FIGURE 21: A_c cannot be fully unit-disk covered by two opposite polar nodes from P_1, even all the nodes in P_2 are active. The shaded regions indicate the uncovered regions.



(a) n_3 and n_8 are active (b) n_1 and n_8 are active

(c) n_2 and n_3 are active (d) n_1 and n_2 are active

FIGURE 22: A_c cannot be fully unit-disk covered by two opposite polar nodes from P_2, even if all the nodes in P_1 are active. The shaded regions indicate the uncovered regions.

As shown in Figure 27(d), only two active nodes n_1 and n_2 from P_3 cannot unit-disk cover A_c even all nodes in P_1 and P_2 are active, which implies that only one node, n_1 or n_2, from P_3 being active cannot unit-disk cover A_c. Similarly from Figure 27(a), only one node, n_3 or n_{10}, from P_3 being active cannot unit-disk cover A_c. Thus, if $U \subseteq N_c$ and $|U \cap P_3| < |P_3|/2 = 2$, U is not a UDC of A_c. That is, if U is a UDC of A_c, $|U \cap P_3| \geq |P_3|/2$. Thus, together with the results from P_1 and P_2, precondition 7(ii) is satisfied.

Furthermore, Figure 27 lists all possible cases in which only two opposite polar nodes from P_3 being active cannot unit-disk cover A_c even all nodes in P_1 and P_2 are active.

Figure 28 illustrates possible cases in which A_c can be unit-disk covered with one node from P_1, two same polar nodes from P_2, and two same polar nodes from P_3 being active. Together with Figures 26 and 27, precondition 7(iii) is satisfied. With satisfaction of preconditions 7(i), 7(ii), and 7(iii) for each port, property (i) is satisfied. Furthermore, in Figure 28, one of the header nodes must be active, so property (iii) is also satisfied.

Figure 29 shows that if no header node is active, A_c cannot be unit-disk covered. That is, property (ii) is satisfied.

Therefore, we prove that Lemma 9 holds for the 3-way connector shown in Figure 3(b).

C. Formal Definitions about Structure Connection

In this appendix, we define how structures may be connected together to form a complex structure.

Definition 25. The structure, $S = (A, N)$, is *connectable*, if, refer to Figure 5, there exists a pair of nodes, n_i and n_{i+1}, such that

(i) $d(n_i, n_{i+1}) = 1$, (hence, n_i and n_{i+1} can be the vertices of a connection patch defined in Definition 26.)

(ii) n_i and n_{i+1} have opposite polarities.

Furthermore, we call S connectable at n_i and n_{i+1} and the set $N_p = \{n_i, n_{i+1}\}$ a *port* of S.

Definition 26. Consider two connectable structure, $S_1 = (A_1, N_1)$ and $S_2 = (A_2, N_2)$. Suppose S_1 and S_2 are connected together via T 1×1 squares, $A_{cp,1}, A_{cp,2}, \ldots, A_{cp,T}$, called *connection patches*. Each $A_{cp,t}$, $1 \leq t \leq T$, is attached to ports $N_{p_1,t}$ of S_1 and $N_{p_2,t}$ of S_2. These ports are positioned at vertices of each connection patch as shown in Figure 5 and called *connected ports*. S_1 and S_2 are *well connected* if the following hold:

(i) $A_1 \cap A_2 = \varnothing$ and $N_1 \cap N_2 = \varnothing$

(ii) for $1 \leq t \leq T$, if $n \in N_{p_1,t}$ and $n' \in N_{p_2,t}$ on the same edge of $A_{cp,t}$, n and n' have opposite polarities, for example, (n_i, n_j) and (n_{i+1}, n_{j+1}) in Figure 5. Furthermore, n and n' are each other's *connection counterpart*.

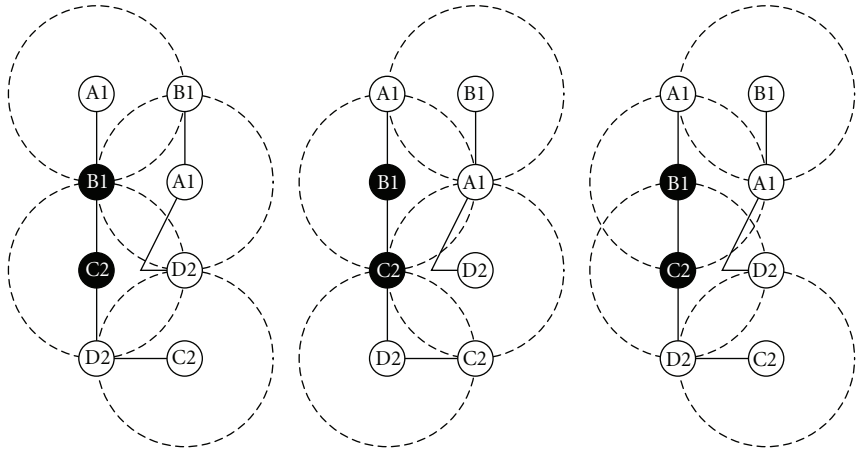
In this NP-completeness proof, we would like to show that two structures are connected in such a way that the nodes, except connected ports, of one structure cannot cover any point of the other structure for preserving the partially well-behaved property. Thus, we have the following definitions.

Definition 27. The structure, $S = (A, N)$, is *least interactively connectable*, if there exists a port, $N_p = \{n_i, n_{i+1}\}$, such that the following preconditions hold.

(i) There exists a point $x \in A$ that is not at the location of n_i or n_{i+1} and can only be unit-disk covered by n_i or n_{i+1}. (This precondition requires that at least one node from each connected port needs to be active.)

(ii) For all $n \in (N - N_p)$, $\text{disk}(n) \cap A_{cp} \subset N_p$. Here A_{cp} is the connection patch attached to N_p. (Nonconnected port nodes and the connection patch are so far that nonconnected port nodes cannot cover any point, except the vertices, of the connection patch. Thus, whether a connection patch can be fully covered only depends on its connected ports.)

Furthermore, we call N_p *the least interactively connectable port*.



(a) $n_4; n_6 \in P_1$ and $n_2; n_8 \in P_2$ are active (b) $n_5; n_7 \in P_1$ and $n_1; n_3 \in P_2$ are active (c) $n_5; n_7 \in P_1$ and $n_2; n_8 \in P_2$ are active

FIGURE 23: With at least one of the header nodes being active, A_c can be fully unit-disk covered by the nodes with the same polarity from each port.

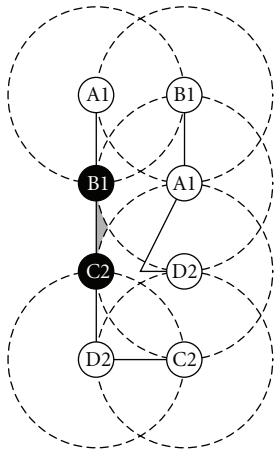


FIGURE 24: Without any header node being active, A_c cannot be fully unit-disk covered, even all the rest of nodes are active.

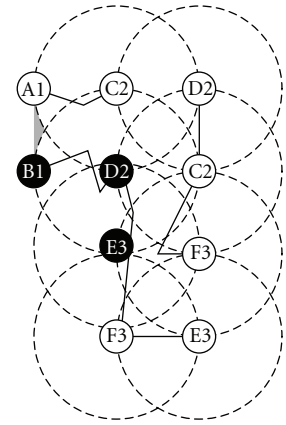


FIGURE 25: A_c cannot be fully unit-disk covered without any node in P_1 being active, even if all the nodes in P_2 and P_3 are active. The shaded region indicates the uncovered region.

Definition 28. The least interactively connectable structures $S_1 = (A_1, N_1)$ and $S_2 = (A_2, N_2)$ are well connected via the connection patches $A_{cp,1}, A_{cp,2}, \ldots, A_{cp,T}$. For $1 \le t \le T$, $A_{cp,t}$ is attached to the least interactively connectable ports $N_{p_1,t}$ of S_1 and $N_{p_2,t}$ of S_2. We call S_1 and S_2 *least interactively connected* if the following preconditions hold.

(i) for all $n \in (N_1 - \bigcup_{1\le t\le T} N_{p_1,t})$, for all $x \in A_2$, $d(n,x) > 1$ and for all $n \in (N_2 - \bigcup_{1\le t\le T} N_{p_2,t})$, for all $x \in A_1, d(n,x) > 1$. (The distance between any nonconnected port nodes of one structure and any point of the other structure is greater than 1. Thus, the nodes, except connected ports, of one structure cannot cover any point of the other structure.)

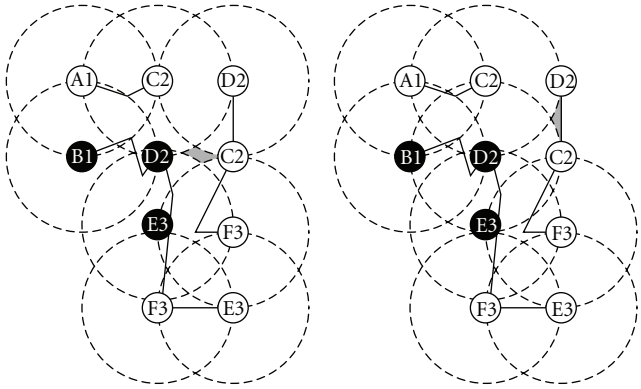
(ii) for $1 \le t \le T$, for all $n \in N_{p_1,t}, \text{disk}(n) \cap A_2 \subset N_{p_2,t}$ and for all $n \in N_{p_2,t}, \text{disk}(n) \cap A_1 \subset N_{p_1,t}$. (The connected ports of one structure cannot cover any point, except their connection counterparts, of the other structure.)

D. The Least Interactive Connectability of Structures

Lemma 29. *The structure $S_v = (A_v, N_v)$ shown in Figure 1 is least interactively connectable at any two nearby nodes on the side of border.*

Proof. Suppose n_i and n_{i+1} are two nearby nodes as shown in Figure 1. It is not difficult to prove that $\{n_i, n_{i+1}\}$ is a port.

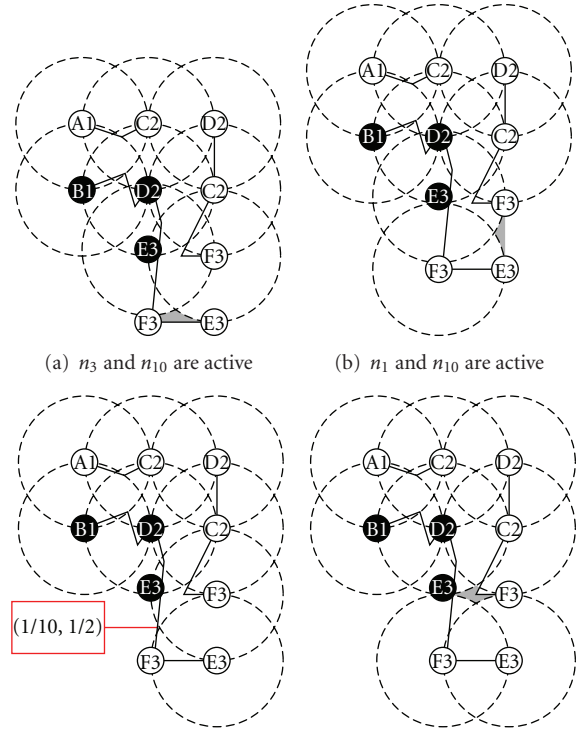
Furthermore, as shown in Figure 1, x, the midpoint of n_i and n_{i+1}, is the point satisfying precondition 27(i). Finally, it is obvious that for all $n \in (N_v - \{n_i, n_{i+1}\})$, $d(n, n_i) \ge 1$ and $d(n, n_{i+1}) \ge 1$. Hence, it is easy to derive



(a) n_5 and n_6 are active (b) n_6 and n_9 are active

(c) n_4 and n_5 are active (d) n_4 and n_9 are active

FIGURE 26: A_c cannot be fully unit-disk covered by two opposite polar nodes from P_2, even if all the nodes in P_1 and P_3 are active. The shaded regions indicate the uncovered regions.



(a) n_3 and n_{10} are active (b) n_1 and n_{10} are active

(c) n_2 and n_3 are active. Note that the tiny uncovered region is located between n_1 and n_{10}. For example, $(1/10, 1/2)$ cannot be covered

(d) n_1 and n_2 are active

FIGURE 27: A_c cannot be fully unit-disk covered by two opposite polar nodes from P_2, even if all the nodes in P_1 and P_3 are active. The shaded regions indicate the uncovered regions.

that precondition 27(ii) is satisfied. Therefore, S_v is least interactively connectable at n_i and n_{i+1}. □

Lemma 30. *The structure $S_e = (A_e, N_e)$ shown in Figure 2 is least interactively connectable at the endpoint pairs indicated by the arrows.*

Proof. It is obvious that the endpoint pairs are also ports, and for each endpoint pair, for example, $\{n_i, n_{i+1}\}$, for all $n \in (N_e - \{n_i, n_{i+1}\}), d(n, n_i) \geq 1$ and $d(n, n_{i+1}) \geq 1$. Hence, it is easy to derive that precondition 27(ii) is satisfied at each endpoint pair. Besides, similar to variable structures, the midpoint of each endpoint pair can only be unit-disk covered by the endpoint pair. Thus, S_e is least interactively connectable at the endpoint pairs. □

Lemma 31. *Each n-way connector of Figure 3 is least interactively connectable at the pairs of nodes indicated by the arrows.*

Proof. It is obvious that the pairs indicated by the arrows are also ports. Furthermore, for each port, it is obvious that the midpoint of the port is the point satisfying precondition 27(i). Moreover, without loss of generality, it is straightforward to prove from Table 2 that, for the port $\{n_7, n_8\}$ of the 3-way connector, for all $n \in (N_c - \{n_7, n_8\})$, $d(n, n_7) \geq 1$ and $d(n, n_8) \geq 1$. Hence, it is easy to derive that precondition 27(ii) is satisfied at $\{n_7, n_8\}$. Similar argument may apply to other ports and the 2-way connector. □

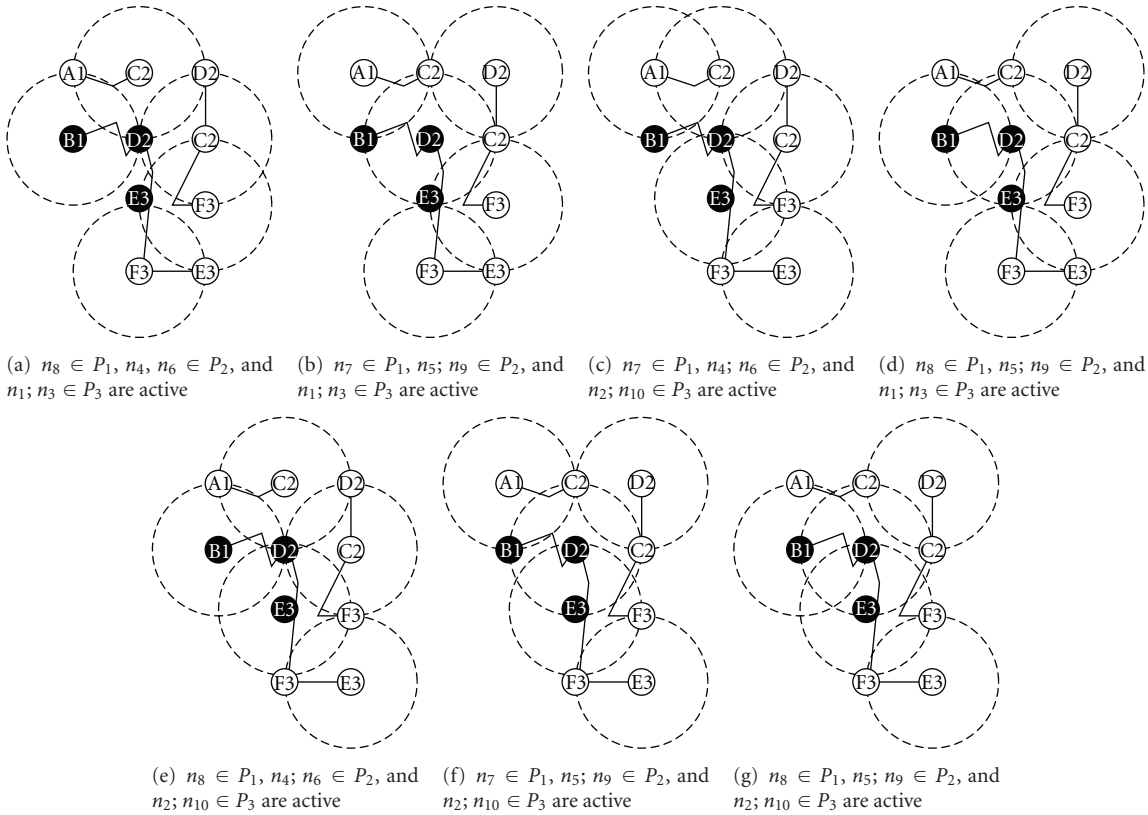
Lemma 32. *The structure $S_p = (A_p, N_p)$ shown in Figure 9 is least interactively connectable at its endpoint pairs.*

Proof. It is obvious that the endpoint pairs are also ports, and for each endpoint pair, for example, $\{n_i, n_{i+1}\}$ for all $n \in (N_e - \{n_i, n_{i+1}\}), d(n, n_i) \geq 1$ and $d(n, n_{i+1}) \geq 1$. Hence, it is easy to derive that precondition 27(ii) is satisfied at each endpoint pair. Besides, similar to variable structures, the midpoint of each endpoint pair can only be unit-disk covered by the endpoint pair. Thus, S_e is least interactively connectable at the endpoint pairs. □

E. Proof of Lemmas about the Least Interactive Connection

E.1. Proof of Lemma 12. First, consider the following lemma.

Lemma 33. *Suppose $S = (A_1, N_1) + (A_2, N_2)$. If $U \subseteq (N_1 \cup N_2)$ is a UDC of $A_1 \cup A_2$, then $U \cap N_1$ and $U \cap N_2$ are UDCs of A_1 and A_2, respectively.*



(a) $n_8 \in P_1$, n_4, $n_6 \in P_2$, and n_1; $n_3 \in P_3$ are active

(b) $n_7 \in P_1$, n_5; $n_9 \in P_2$, and n_1; $n_3 \in P_3$ are active

(c) $n_7 \in P_1$, n_4; $n_6 \in P_2$, and n_2; $n_{10} \in P_3$ are active

(d) $n_8 \in P_1$, n_5; $n_9 \in P_2$, and n_1; $n_3 \in P_3$ are active

(e) $n_8 \in P_1$, n_4; $n_6 \in P_2$, and n_2; $n_{10} \in P_3$ are active

(f) $n_7 \in P_1$, n_5; $n_9 \in P_2$, and n_2; $n_{10} \in P_3$ are active

(g) $n_8 \in P_1$, n_5; $n_9 \in P_2$, and n_2; $n_{10} \in P_3$ are active

FIGURE 28: With at least one of the header nodes being active, A_c can be fully unit-disk covered by the nodes with the same polarity from each port

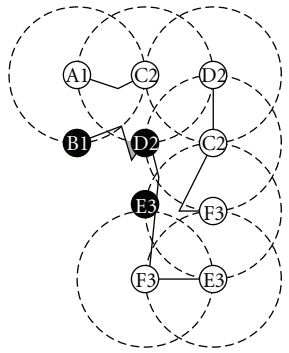


FIGURE 29: Without any header node being active, A_c cannot be fully unit-disk covered, even if all the rest of nodes are active.

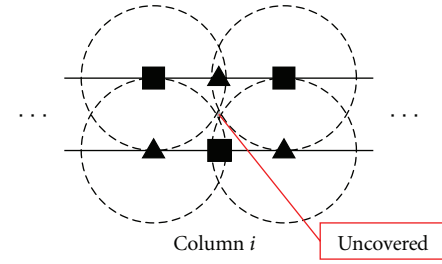


FIGURE 30: The tiny shaded region as indicated in the middle of the column-pair i cannot be covered while none of nodes from the column-pair i is active.

Proof. Suppose $U \cap N_1$ is not a UDC of A_1, then there exists a point $x^\dagger \in A_1$ such that $x^\dagger$ cannot be covered by any node in $U \cap N_1$ but some node, say n, in $U \cap N_2$.

From Definition 28(i), we know the distance between any node in N_2, except connected ports, and any point in A_1 is greater than 1. Thus, none of nodes in N_2, except connected ports, can cover any point in A_1. Hence, n must belong to some connected port. Without loss of generality, let n be n_j of Figure 5. From Definition 28(ii), the only point in A_1 which can be covered by n_j is at the location of n_i, and it implies that $x^\dagger$ can only be at the location of n_i.

Let $x \in A_1$ be the point that is not at the location of n_i or n_{i+1} and can only be covered by n_i or n_{i+1}, as stated in Definition 27(i). From the above argument, x cannot be covered by any node of N_2. Thus, at least one node from $\{n_i, n_{i+1}\}$ must be active in U. Hence, there is a contradiction since either n_i or n_{i+1} can cover $x^\dagger$, that is, the location of n_i. Therefore, $U \cap N_1$ is a UDC of A_1 and similar argument can also derive that $U \cap N_2$ is a UDC of A_2. □

With the help of Lemma 33, we can prove Lemma 12.

If $U \subseteq (N_1 \cup N_2)$ is a UDC of A, U is a UDC of $A_1 \cup A_2$. By Lemma 33, $U \cap N_1$ and $U \cap N_2$ are UDCs of A_1 and A_2, respectively, which implies that $|U \cap N_1| \geq l_1$ and $|U \cap N_2| \geq l_2$.

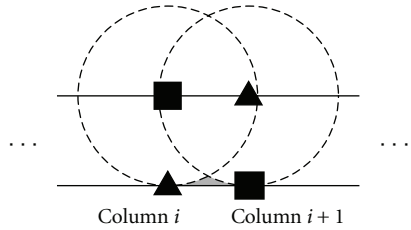
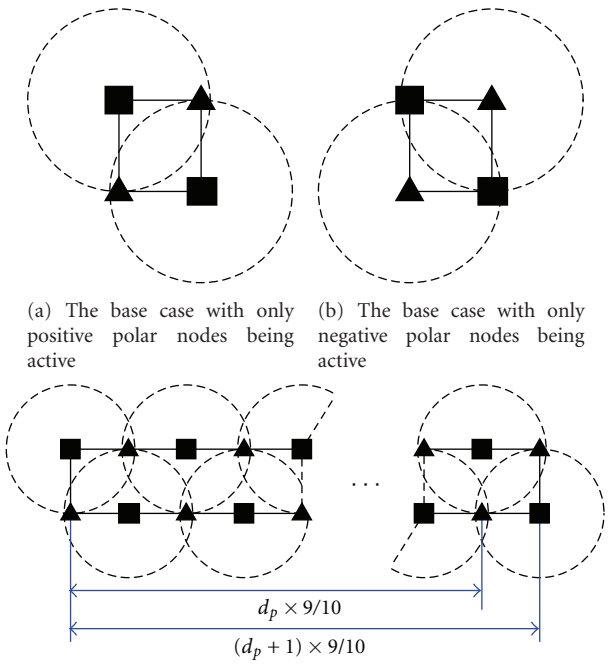


FIGURE 31: The shaded region cannot be covered by two nearby opposite polar nodes.



(a) The base case with only positive polar nodes being active

(b) The base case with only negative polar nodes being active

(c) Induction step for the positive polarity case

FIGURE 32: Induction proof for that the $d_p \times (9/10)$ long main structure can be unit-disk covered by $d_p + 1$ nodes with the same polarity. Note that for the main structure of a polarity inverter, $d_p = 10$.

From Definition 26(i), $N_1 \cap N_2 = \emptyset$ and, hence, $|U| = |U \cap N_1| + |U \cap N_2| = l_1 + l_2$. Thus, $|U \cap N_1| = l_1$ and $|U \cap N_2| = l_2$.

E.2. Proof of Connection Lemma. Before proving Connection Lemma, consider the following lemma.

Lemma 34. *If $S = (A_1, N_1) + (A_2, N_2)$ via one connection patch, then the following propositions will hold. (We note that this lemma may be extended to more than one connection patches with possible minor modification on Definition 28(ii); however we do not use this property to prove Connection Lemma and therefore ignore it.)*

(a) *If $U \subseteq (N_1 \cup N_2)$ is an MUDC of $A_1 \cup A_2$, then $U \cap N_1$ and $U \cap N$ are MUDCs of A_1 and A_2, respectively.*

(b) *If $U_1 \subseteq N_1$ and $U_2 \subseteq N_2$ are MUDCs of A_1 and A_2, respectively, then $U_1 \cup U_2$ is an MUDC of $A_1 \cup A_2$.*

Proof. Let $U \subseteq (N_1 \cup N_2)$ be an MUDC of $A_1 \cup A_2$. By Lemma 33, $U \cap N_1$ and $U \cap N_2$ are UDCs of A_1 and A_2 respectively.

Suppose $U \cap N_1$ is not an MUDC of A_1. Then there exists a UDC $U' \subseteq N_1$ of A_1 and $|U'| < |U \cap N_1|$. Since U' is a UDC of A_1 and $U \cap N_2$ is a UDC of A_2, $U' \cup (U \cap N_2)$ is a UDC of $A_1 \cup A_2$. Since $N_1 \cap N_2 = \emptyset$ by Definition 26(i), $|U' \cup (U \cap N_2)| = |U'| + |U \cap N_2| < |U \cap N_1| + |U \cap N_2| = |(U \cap N_1) \cup (U \cap N_2)| = |U|$, which contradicts the assumption that U is an MUDC of $A_1 \cup A_2$. Similar argument can apply to $U \cap N_2$. Hence, we have proved proposition (a) of this lemma.

Now assume $U_1 \subseteq N_1$ and $U_2 \subseteq N_2$ are MUDCs of A_1 and A_2, respectively, and, obviously, $U_1 \cup U_2$ is a UDC of $A_1 \cup A_2$. Suppose there exists an MUDC U'' of $A_1 \cup A_2$ and $|U''| < |U_1 \cup U_2|$. By proposition (a), $U'' \cap N_1$ and $U'' \cap N_2$ are MUDC of A_1 and A_2, respectively. Again, since $N_1 \cap N_2 = \emptyset$, $|U''| = |U'' \cap N_1| + |U'' \cap N_2| < |U_1 \cup U_2| = |U_1| + |U_2|$. Hence, $|U'' \cap N_1| < |U_1|$ or $|U'' \cap N_2| < |U_2|$, which contradicts the assumption that U_1 and U_2 are MUDCs of A_1 and A_2, respectively. Hence, $U_1 \cup U_2$ is an MUDC of $A_1 \cup A_2$. □

With the help of Lemmas 33 and 34, we can prove Lemma 13.

Let A_{cp} be the connection patch. The key point to the proof is precondition (ii). That is, A_{cp} can be automatically covered by the active nodes of ports without the help of nodes not in U_1' and U_2', if $U_1' \cap N_1'$ and $U_2' \cap N_2'$ have the same polarity. Thus, the MUDC of $A_1 \cup A_2 \cup A_{cp}$ can only be $U_1' \cup U_2'$.

Note that $N_1' \cap N_2' = \emptyset$ by Definition 26(i). Thus, precondition 7(i) is satisfied since S_1 and S_2 are partially well behaved on N_1' and N_2', respectively.

Moreover, let $\widetilde{U} \subseteq (N_1 \cup N_2)$ be a UDC of $A_1 \cup A_2 \cup A_{cp}$. By Lemma 33, it is easy to derive that $\widetilde{U} \cap N_1$ and $\widetilde{U} \cap N_2$ are UDCs of A_1 and A_2, respectively. Then by precondition (i) and Definition 7(ii), $|\widetilde{U} \cap N_1'| \geq |N_1'|/2$ and $|\widetilde{U} \cap N_2'| \geq |N_2'|/2$. Since $N_1' \cap N_2' = \emptyset$, $|\widetilde{U} \cap (N_1' \cup N_2')| = |\widetilde{U} \cap N_1'| + |\widetilde{U} \cap N_2'| \geq (|N_1'| + |N_2'|)/2 = |N_1' \cup N_2'|/2$. That is, precondition 7(ii) is satisfied.

Now, suppose $U' \subseteq (N_1 \cup N_2)$ is an MUDC of $A_1 \cup A_2$ and $U \subseteq N_1 \cup N_2$ is an MUDC of $A_1 \cup A_2 \cup A_{cp}$. Obviously, $|U| \geq |U'|$. Since $\widetilde{U}$ is a UDC of $A_1 \cup A_2 \cup A_{cp}$, $|\widetilde{U}| \geq |U| \geq |U'|$. Consequently, by showing there exists an MUDC U' of $A_1 \cup A_2$ such that U' is a UDC of $A_1 \cup A_2 \cup A_{cp}$, we can prove that $|U| = |U'|$, which implies that U is also an MUDC of $A_1 \cup A_2$.

Note that precondition (ii) states that there exists an MUDC of A_1, $U_1' \subseteq N_1$, and an MUDC of A_2, $U_2' \subseteq N_2$, such that $U_1' \cap N_1'$ and $U_2' \cap N_2'$ have the same polarity. Let $U' = U_1' \cup U_2'$ be such a candidate. Hence, by Lemma 34(b), U' is an MUDC of $A_1 \cup A_2$.

Since $U_1' \cap N_1'$ and $U_2' \cap N_2'$ have the same polarity, $U' \cap (N_1' \cup N_2') = (U_1' \cap N_1') \cup (U_2' \cap N_2') = N_1'^{+} \cup N_2'^{+}$ or $N_1'^{-} \cup N_2'^{-}$. Referring to Figure 5, by Definitions 25(ii) and 26(ii), either the pair (n_i, n_{j+1}) or (n_{i+1}, n_j) is active. Obviously, either pair can unit-disk cover A_{cp} by Lemma 10.

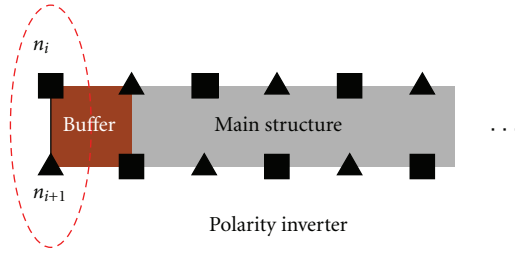


FIGURE 33: Connection Lemma is applied to the main structure and the left end $S_l = (A_l, N_l)$ (enclosed in dashed line). A_l is the line segment $\overline{n_i\, n_{i+1}}$ and $N_l = \{n_i, n_{i+1}\}$.

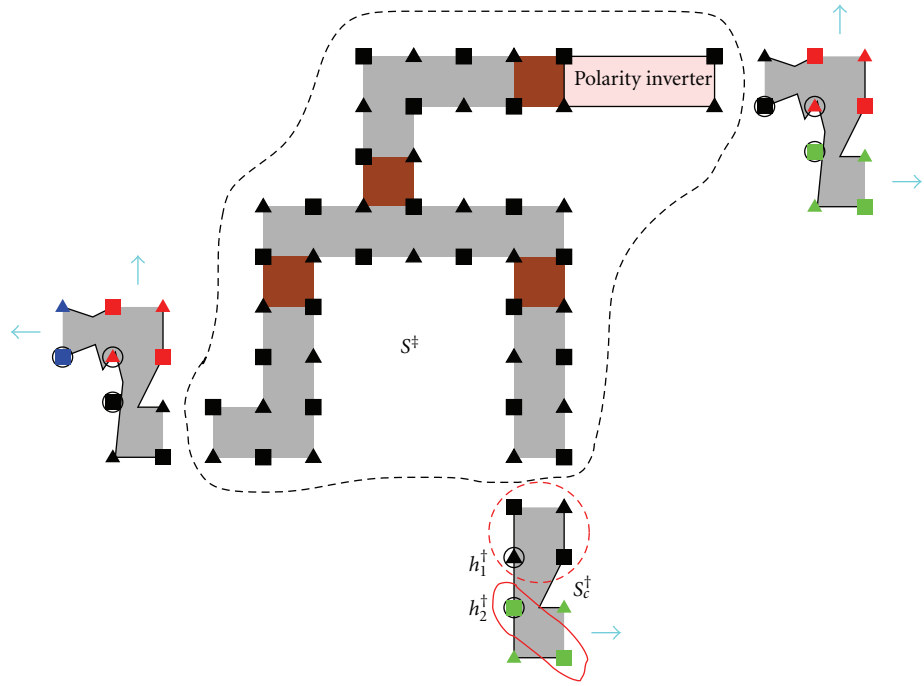


FIGURE 34: The structure enclosed in dashed line is $S^{\ddagger}$. The set of nodes enclosed in dashed circle is $P_1^{\dagger}$, and the set of nodes enclosed in solid line is $P_2^{\dagger p}$. By proving $S^{\ddagger} + S_c^{\dagger}$ is partially well-behaved on $N^{\ddagger} \cup P_1^{\dagger}$, this lemma will be proved.

Therefore, U' is a UDC of $A_1 \cup A_2 \cup A_{cp}$, which implies that, for any MUDC U of $A_1 \cup A_2 \cup A_{cp}$, $|U| = |U'|$ and U is also an MUDC of $A_1 \cup A_2$.

Now we need to prove that, for any MUDC U of $A_1 \cup A_2 \cup A_{cp}$, $U \cap (N_1' \cup N_2')$ only contains the nodes with the same polarity. Since U is also an MUDC of $A_1 \cup A_2$, by Lemma 34(a), $U \cap N_1$, denoted as U_1, is an MUDC of A_1 and $U \cap N_2$, denoted as U_2, is also an MUDC of A_2. By Definition 7(iii), $U_1 \cap N_1'$ only contains the nodes with the same polarity and so does $U_2 \cap N_2'$. Thus, we need to prove that $U \cap (N_1' \cup N_2')$ cannot be $N_1'^{+} \cup N_2'^{-}$ or $N_1'^{-} \cup N_2'^{+}$. Referring to Figure 5, without loss of generality, let $U \cap (N_1' \cup N_2') = N_1'^{+} \cup N_2'^{-}$ with $n_i \in N_1'^{+}$ and $n_j \in N_2'^{-}$ being active. Note that it is obvious that $A_{cp} \not\subseteq \text{disk}(\{n_i, n_j\})$. (In other words, A_{cp} cannot be covered by n_i and n_j.) That is, $\text{disk}(\{n_i, n_j\}) \cap A_{cp} \subsetneq A_{cp}$.

Denote $\overline{N_1} = N_1 - \{n_i, n_{i+1}\}$. Obviously, $N_1'^{+} \subset (\overline{N_1} \cup \{n_i\})$, and, hence, $\text{disk}(N_1'^{+}) \cap A_{cp} \subseteq \text{disk}(\overline{N_1} \cup \{n_i\}) \cap A_{cp}$. From Definition 27(ii), for all $n \in \overline{N_1}$, $\text{disk}(n) \cap A_{cp} \subset \{n_i, n_{i+1}\}$. Thus, $\text{disk}(\overline{N_1}) \cap A_{cp} \subset \{n_i, n_{i+1}\} \subset \text{disk}(n_i) \cap A_{cp}$, which implies $\text{disk}(\overline{N_1} \cup \{n_i\}) \cap A_{cp} = \text{disk}(n_i) \cap A_{cp}$. Hence, $\text{disk}(N_1'^{+}) \cap A_{cp} \subseteq \text{disk}(n_i) \cap A_{cp}$. Similarly, $\text{disk}(N_2'^{-}) \cap A_{cp} \subseteq \text{disk}(n_j) \cap A_{cp}$.

Therefore, $\text{disk}(N_1'^{+} \cup N_2'^{-}) \cap A_{cp} = (\text{disk}(N_1'^{+}) \cap A_{cp}) \cup (\text{disk}(N_2'^{-}) \cap A_{cp}) \subseteq (\text{disk}(n_i) \cap A_{cp}) \cup (\text{disk}(n_j) \cap A_{cp}) = \text{disk}(\{n_i, n_j\}) \cap A_{cp} \subsetneq A_{cp}$. That is, $A_{cp} \not\subseteq \text{disk}(N_1'^{+} \cup N_2'^{-})$. Thus, simply nodes from $N_1'^{+}$ and $N_2'^{-}$ cannot unit-disk cover A_{cp}, and at least one more node not in U is needed, for example, $n_{i+1} \in N_1'^{-}$ or $n_{j+1} \in N_2'^{+}$. Hence, U cannot be an MUDC of $A_1 \cup A_2 \cup A_{cp}$.

Therefore, we can conclude that if $U \subseteq (N_1 \cup N_2)$ is an MUDC of $A_1 \cup A_2 \cup A_{cp}$, $U \cap (N_1' \cup N_2')$ contains only the nodes with the same polarity. That is, precondition 7(iii) is satisfied and, thus, S is partially well behaved on $N_1' \cup N_2'$.

F. Proof of Lemma 14

Similar to the proof of variable structures in Lemma 8, the well-aligned and well-behaved properties of the main

structure can be proved via Figures 30, 31, and 32. Consequently, with the help of Lemma 10 and Connection Lemma, the well-aligned and well-behaved properties of S_p can be proved. Note that, in this case, Connection Lemma is applied to the main structure and both "end", $S_l = (A_l, N_l)$ and $S_r = (A_r, N_r)$. Referring to Figure 33, A_l is the line segment $\overline{n_i n_{i+1}}$ and $N_l = \{n_i, n_{i+1}\}$. Besides, the left buffer is the connection patch for connecting S_l and the main structure. Similar idea can apply to the right end, that is, S_r.

G. Proof of Lemma 17

For the ith variable v_i, consider the composite structure $S^{\ddagger} = (A^{\ddagger}, N^{\ddagger})$ with $A^{\ddagger} = AV_i \cup AE_i \cup AI_i \cup ACP_i^*$ and $N^{\ddagger} = NV_i \cup NE_i \cup NI_i$. Here ACP_i^* is the union of the shaded region from all the associated connection patches, except the ones attached to the associated connectors. An example of $S^{\ddagger}$ is illustrated in Figure 34. In other words, the composite structure of $S^{\ddagger}$ and all associated connectors is the territory T_i. It is obvious that $S^{\ddagger}$ is well behaved by Lemmas 8, 14, and Connection Lemma. After that, we connect each associated n-way connector to $S^{\ddagger}$ iteratively. At each iteration, we can prove the partially well behaved property of the composite structure. When all associated connectors are connected, this lemma is proved.

Without loss of generality, consider an MUDC of $A^{\ddagger}$, $N^{\ddagger +}$. Since $2 \le |c| \le 3$ for each clause c, the clauses are represented by 2-way or 3-way connectors. Note that an P3SAT instance has a graph structure, G_B, in which there is at most one edge between two nodes. That is, the variables in a clause are different. Thus, for each associated connector of T_i, there is another partition, that is, not enclosed in dashed line in Figure 10, which is not an associated partition. Without loss of generality, suppose $S_c^{\dagger} = (A_c^{\dagger}, N_c^{\dagger} \langle \mathcal{P}^{\dagger}, H^{\dagger} \rangle)$ is an associated 2-way connector of S_{v_i}. Here $\mathcal{P}^{\dagger} = \{P_1^{\dagger}, P_2^{\dagger}\}$ and $H^{\dagger} = \{h_1^{\dagger}, h_2^{\dagger}\}$. Referring to Figure 34, let $P_1^{\dagger}$ be the associated partition and $P_2^{\dagger p} = \{n \in P_2^{\dagger} \mid n \text{ has the polarity of } h_2^{\dagger}\}$. We want to prove $S^{\ddagger} + S_c^{\dagger}$ is partially well behaved on $N^{\ddagger} \cup P_1^{\dagger}$.

From Lemma 9(i), Lemma 9(iii), and Definition 7, it is easy to derive that $P_1^{\dagger +} \cup P_2^{\dagger p}$ is an MUDC of $A_c^{\dagger}$. Thus, $N^{\ddagger +}$ and $P_1^{\dagger +} \cup P_2^{\dagger p}$ are the MUDCs of $A^{\ddagger}$ and $A_c^{\dagger}$, respectively, and precondition (ii) of Connection Lemma is satisfied. (In this case, $A_1 = A^{\ddagger}$, $A_2 = A_c^{\dagger}$, $N_1' = N^{\ddagger}$, $N_2' = P_1^{\dagger}$, $U_1' = N^{\ddagger +}$, and $U_2' = P_1^{\dagger +} \cup P_2^{\dagger p}$.) Besides, as mentioned earlier, the variables in a clause are different. That is, $S^{\ddagger}$ and $S_c^{\dagger}$ are connected via one connection patch. Thus, Connection Lemma is applicable. By Lemma 9(i) and Connection Lemma, $S^{\ddagger} + S_c^{\dagger}$ is partially well behaved on $N^{\ddagger} \cup P_1^{\dagger}$. This procedure can be iteratively applied to the rest of connectors and, hence, the lemma is proved.

Acknowledgment

This research was supported by National Science Council (NSC), Taiwan, under Grant NSC 99-2221-E-194-021. The author gratefully acknowledges this support.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] ANSI/IEEE, Standard 802.11, "Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) specifications," 1999.

[3] J. A. Stine and G. D. Veciana, "Improving energy efficiency of centrally controlled wireless data networks," *Wireless Networks*, vol. 8, no. 6, pp. 681–700, 2002.

[4] A. Srinivas and E. Modiano, "Minimum energy disjoint path routing in wireless Ad-hoc networks," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pp. 122–133, ACM Press, San Diego, Calif, USA, 2003.

[5] A. Sankar and Z. Liu, "Maximum lifetime routing in wireless Ad-hoc networks," in *Proceedings of the IEEE Infocom*, Hong Kong, China, March 2004.

[6] S. Singh, C. S. Raghavendra, and J. Stepanek, "Power-aware broadcasting in mobile Ad Hoc networks," in *Proceedings of the 10th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, Osaka, Japan, September 1999.

[7] J. Wu, B. Wu, and I. Stojmenovic, "Power-aware broadcasting and activity scheduling in Ad Hoc wireless networks using connected dominating sets," *Wireless Communications and Mobile Computing*, vol. 3, no. 4, pp. 425–438, 2003.

[8] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Multicasting in energy-limited Ad-hoc wireless networks," in *Proceedings of the IEEE Military Communications Conference*, pp. 723–729, October 1998.

[9] W. Liang, "Approximate minimum-energy multicasting in wireless Ad Hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 4, pp. 377–387, 2006.

[10] L. Hu, "Topology control for multihop packet radio networks," *IEEE Transactions on Communications*, vol. 41, no. 10, pp. 1474–1481, 1993.

[11] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, "Distributed topology control for power efficient operation in multihop wireless Ad Hoc networks," in *Proceedings of the IEEE Infocom*, pp. 1388–1397, Anchorage, AK, USA, April 2001.

[12] R. C. Shah and J. M. Rabaey, "Energy aware routing for low energy Ad Hoc sensor networks," in *Proceedings of the IEEE Wireless Communication and Networking Conference*, Orlando, Fla, USA, March 2002.

[13] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609–619, 2004.

[14] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70–80, 2002.

[15] H. Nama and N. Mandayam, "Sensor networks over information fields: optimal energy and node distributions," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 1842–1847, March 2005.

[16] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002.

[17] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless Ad-hoc sensor networks," in *Proceedings of the IEEE Infocom*, pp. 1380–1387, Anchorage, Alaska, USA, April 2001.

[18] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, pp. 1734–1743, San Franciso, Calif, USA, April 2003.

[19] Y.-C. Tseng, S.-P. Kuo, H.-W. Lee, and C.-F. Huang, "Location tracking in a wireless sensor network by mobile agents and its data fusion strategies," in *Proceedings of the 2nd International Symposium on Information Processing in Sensor Networks*, pp. 625–641, Palo Alto, Calif, USA, April 2003.

[20] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications*, vol. 2, pp. 472–476, Helsinki, Finland, June 2001.

[21] S. Funke, A. Kesselman, F. Kuhn, Z. Lotker, and M. Segal, "Improved approximation algorithms for connected sensor cover," *Wireless Networks*, vol. 13, no. 2, pp. 153–164, 2007.

[22] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, chapter A3, W.H. Freeman and Company, 1979.

[23] D. S. Johnson, "Approximation algorithms for combinatorial problems," *Journal of Computer and System Sciences*, vol. 9, no. 3, pp. 256–278, 1974.

[24] L. Lovász, "On the ratio of optimal integral and fractional covers," *Discrete Mathematics*, vol. 13, no. 4, pp. 383–390, 1975.

[25] V. Chvátal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.

[26] U. Feige, "A threshold of in n for approximating set cover," *Journal of the ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[27] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in wireless Ad-Hoc sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 139–150, ACM Press, Rome, Italy, 2001.

[28] S. Meguerdichian, S. Slijepcevic, V. Karayan, and M. Potkonjak, "Localized algorithms in wireless Ad-Hoc networks: location discovery and sensor exposure," in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 106–116, ACM Press, Long Beach, CA, USA, 2001.

[29] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, pp. 129–143, Philadelphia, PA, USA, October 2004.

[30] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 32–41, ACM Press, Atlanta, Ga, USA, 2002.

[31] B. Carbunar, A. Grama, J. Vitek, and O. Carbunar, "Redundancy and coverage detection in sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 1, pp. 94–128, 2006.

[32] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," in *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, pp. 115–121, ACM Press, San Diego, Calif, USA, 2003.

[33] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *The Wireless Ad Hoc and Sensor Networks*, vol. 1, pp. 89–124, 2005.

[34] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pp. 28–39, ACM Press, Los Angeles, Calif, USA, 2003.

[35] H. Gupta, Z. Zhou, S. R. Das, and Q. Gu, "Connected sensor cover: self-organization of sensor networks for efficient query execution," *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, pp. 55–67, 2006.

[36] A. V. S. Kumar, S. Arya, and H. Ramesh, "Hardness of set cover with intersection 1," in *Proceedings of the 27th International Colloquium on Automata, Languages and Programming*, pp. 624–635, Springer, London, UK, 2000.

[37] R. J. Fowler, M. Paterson, and S. L. Tanimoto, "Optimal packing and covering in the plane are NP complete," *Information Pro-cessing Letters*, vol. 12, no. 3, pp. 133–137, 1981.

[38] N. Megiddo and K. J. Supowit, "On the complexity of some common geometric location problems," *SIAM Journal on Computing*, vol. 13, no. 1, pp. 182–196, 1984.

[39] M. V. Marathe, V. Radhakrishnan, I. Harry, B. Hunt, and S. S. Ravi, "Hierarchically specified unit disk graphs," *Theoretical Computer Science*, vol. 174, no. 1-2, pp. 23–65, 1997.

[40] H. Brönnimann and M. T. Goodrich, "Almost optimal set covers in finite VC-dimension," *Discrete & Computational Geometry*, vol. 14, no. 4, pp. 463–479, 1995.

[41] T. F. Gonzalez, "Covering a set of points in multidimensional space," *Information Processing Letters*, vol. 40, no. 4, pp. 181–188, 1991.

[42] D. S. Hochbaum and W. Maass, "Approximation schemes for covering and packing problems in image processing and Vlsi," *Journal of the ACM*, vol. 32, no. 1, pp. 130–136, 1985.

[43] M. Franceschetti, M. Cook, and J. Bruck, "A geometric theorem for network design," *IEEE Transactions on Computers*, vol. 53, no. 4, pp. 483–489, 2004.

[44] M. Franceschetti, M. Cook, and J. Bruck, "A Geometric Theorem for Approximate Disk Covering Algorithms," California Institute of Technology, Technical Report ETR035, 2001 http://www.paradise.caltech.edu/papers/etr035.pdf.

[45] D. Lichtenstein, "Planar formulae and their uses," *SIAM Journal on Computing*, vol. 11, no. 2, pp. 329–343, 1982.

[46] M. R. Garey and D. S. Johnson, *Computers and Intractability: AGuide to the Theory of NP-Completeness*, chapter A9, W.H. Freeman and Company, 1979.