Chapter 6
Coverage

The only difference between suicide and martyrdom is press coverage.

CHUCK PALAHNIUK

6.1 Motivation and Overview

A classic type of resource management problem is as follows: Given a certain amount of resource and a set of users, find an assignment of resource to maximize the number of satisfied users. The maximum lifetime coverage is such a classic type of problem in wireless sensor networks.

When a very large number of sensors are randomly deployed into a certain region possibly by an aircraft to monitor a certain set of targets, usually, there are a lot of redundant sensors. A better usage of those redundant sensors is to schedule active/sleep time of sensors to increase the lifetime of the system.

A simple scheduling is to divide sensors into disjoint subsets, each of which fully covers all targets, called a *sensor cover* [18, 80].

SENSOR-COVER-PARTITION: Given n targets $r_1, \ldots, r_n$ and m sensors $s_1, \ldots, s_m$, each covering a subset of targets, find the maximum number of disjoint sensor covers.

This problem is NP-hard. Various heuristics and approximation algorithms have been given in [11, 13, 96]. In general, there is no polynomial-time $(-\varepsilon)\ln n$)-approximation for any $\varepsilon > 0$ unless $NP \subseteq DTIME(n^{O(\log\log n)})$ [48] and there exists polynomial-time $O(\log n)$-approximation [6, 80]. But, there is an open problem in a special case.

Open Problem 6.1.1. *Suppose all sensors are uniform, that is, they have the same sensing radius. It is unknown whether a polynomial-time constant-approximation exists or not.*

D.-Z. Du and P.-J. Wan, *Connected Dominating Set; Theory and Applications*, Springer Optimization and Its Applications 77, DOI 10.1007/978-1-4614-5242-3_6,

When the sensor set and the target set are identical, SENSOR-COVER-PARTITION becomes the following domatic partition problem.

MAX#DS: Given a graph $G = (V, E)$, partition the vertex set V into maximum number of disjoint dominating sets.

In general graph, there is no polynomial-time $(1-\varepsilon)\ln n$-approximation unless $NP \subseteq DTIME(n^{O(\log\log n)})$ and there exists polynomial-time $O(\log n)$-approximation for MAX#DS [48]. However, for unit disk graphs, there is a polynomial-time constant-approximation [86].

For this type of scheduling, the sensor is activated only once, that is, once the sensor is activated, it keeps active until it dies.

Cardei et al. [15] found that it is possible to increase the lifetime if each sensor is allowed to alternate between active and sleeping states. An example can be found in Chap. 1. The model is also better supported by an interesting fact discovered in [64] that putting a sensor alternatively in active and sleeping states in a proper way may double its lifetime since the battery could be recovered in a certain level during sleeping. The formulation of this model is as follows.

MAX-LIFETIME COVERAGE: Given n targets $t_1, \ldots, t_n$ and m sensors $s_1, \ldots, s_m$, each covering a subset of targets, find a family of sensor cover $S_1, \ldots, S_p$ with time lengths $t_1, \ldots, t_p$ in $[0, 1]$, respectively, to maximize $t_1 + \cdots + t_p$ subject to that the total active time of every sensor is at most 1.

This is still an NP-hard problem. Cardei [15] formulated it as a 0-1 integer programming and designed a heuristic without guaranteed theoretical bound. Berman et al. [6, 7] first designed an approximation algorithm for MAX-LIFETIME COVERAGE with theoretical bound. They showed that there exists a polynomial-time approximation for MAX-LIFETIME COVERAGE with performance ratio $O(\log n)$ where n is the number of sensors. By employing Garg–Könemann theorem [55], Berman et al. reduced MAX-LIFETIME COVERAGE to the following:

MINW-SENSOR-COVER: Consider n targets $t_1, \ldots, t_n$ and m sensors $s_1, \ldots, s_m$, each covering a subset of targets. Given a weight function on sensors $c : \{s_1, \ldots, s_m\} \to R^+$, find the minimum total weight sensor cover.

They showed that if MINW-SENSOR-COVER has a polynomial-time ρ-approximation, then MAX-LIFETIME COVERAGE has a polynomial-time $(1+\varepsilon)\rho$-approximation for any $\varepsilon > 0$. Note that MINW-SENSOR-COVER is equivalent to MINW-SENSOR-COVER. Therefore, it has a polynomial-time $(1+\log n)$-approximation. Hence, MAX-LIFETIME SENSOR COVER has a polynomial-time $O(\log n)$-approximation. Actually, the first one who found the application of Garg-Könemann theorem in study of lifetime maximization type of problems is Calnescu et al. [12].

Ding et al. [34] noted that all results in Chap. 5 about MINW-DS can be extended to MINW-SENSOR-COVER in the case that all sensors and targets lie in the Euclidean plane and all sensors have the same covering radius. Therefore, they proved that in this case, MAX-LIFETIME COVERAGE has polynomial-time 3.63-approximation.

Du et al. [37] extended this approach to study the coverage problem with connectivity requirement. They constructed a polynomial-time constant-approximation in geometric case and $O(\log n)$-approximation in general case. However, many maximum lifetime coverage with connectivity requirement are still open. The following is an example.

Open Problem 6.1.2. *Does* MAX#CDS *have a polynomial-time constant-approximation in unit disk graphs?*

6.2 Max-Lifetime Connected Coverage

As described in the previous section, the method of Garg and Könemann [55] plays an important role in design of constant-approximation for various problems on the maximum lifetime coverage. In this section, we introduce it through the work of Du et al. [37].

Du et al. [37] studied a quite general model of wireless sensor networks which was previously studied by Zhang and Li [126]. In this model, each sensor has two modes, active mode and sleep mode, and the active mode has two phases, the full-active phase and the semi-active phase. A full-active sensor can sense, transmit, receive, and relay the data packets. A semi-active sensor cannot sense data packets, but it can transmit, receive, and relay data packets. Usually, a sensor in the full-active phase consumes more energy than in the semi-active phase.

Sensors are often randomly deployed into hostile environment, such as battlefield and inaccessible area with chemical or nuclear pollution, so that recharging batteries of sensors is a mission impossible. Assume the battery of each sensor contains a certain amount of energy, say unit amount. Then the lifetime of each sensor depends on energy consumption.

Du et al. [37] studied the following problem:

> MAX-LIFETIME CONNECTED-COVERAGE with two active phases: Given a set of targets and a set of sensors with two active phases, find an active/sleeping schedule for sensors to maximize the system lifetime where the network system is said to be *alive* if the following conditions are satisfied:
>
> (A1) Every target is monitored by a full-active sensor.
> (A2) All (full-/semi-) active sensors induce a connected subgraph.

They studied this problem with the primal-dual method of Garg and Könemann [55].

Let S be the set of all sensors. Assume all sensors are uniform, that is, they have the same communication radius R_c, the same sensing radius R_s, the same full-active energy consumption u of unit time and the same semi-active energy consumption v of unit time. Also, assume $u \geq v$. A pair p of sets is called an *active sensor set pair* if $p = (p_1, p_2)$ where p_1 is a set of full-active sensors and p_2 is a set of semi-active sensors with $p_1 \cap p_2 = \emptyset$. For any active sensor set pair p, define

$$a_{s,p} = \begin{cases} u & \text{if } s \in p_1, \\ v & \text{if } s \in p_2, \\ 0 & \text{otherwise.} \end{cases}$$

Suppose $\mathcal{C}$ is the collection of all active sensor set pairs satisfying conditions (A1) and (A2). Then MAX-LIFETIME CONNECTED COVERAGE with two active phases can be formulated as the following linear programming:

$$\begin{aligned} \max \quad & \sum_{p \in \mathcal{C}} x_p \\ \text{subject to} \quad & \sum_{p \in \mathcal{C}} a_{s,p} x_p \le 1 \quad \text{for } s \in S \\ & x_p \ge 0 \quad \text{for } p \in \mathcal{C}. \end{aligned}$$

Its dual is as follows.

$$\begin{aligned} \min \quad & \sum_{s \in S} y_s \\ \text{subject to} \quad & \sum_{s \in S} a_{s,p} y_s \ge 1 \quad \text{for } p \in \mathcal{C}, \\ & y_s \ge 0 \quad \text{for } s \in S. \end{aligned}$$

Motivated from the work of Garg and Könemann [55], Du et al. [37] designed the following primal-dual algorithm.

Primal-Dual Algorithm DPWW

Initially, choose $x_p = 0$ for all $p \in \mathcal{C}$ and $y_s = \delta$ for all $s \in S$ where δ is a positive constant which will be determined later.

In each iteration, carry out the following steps until $(y_s, s \in S)$ becomes dual feasible, that is, all constrains in dual linear programming are satisfied:

Step 1. Compute a ρ-approximation solution p^* for

MINW-CSC with two active phases:

$$\min_{p \in \mathcal{C}} \sum_{s \in S} a_{s,p} y_s.$$

Step 2. Compute a solution s^* for

$$\max_{s \in S} a_{s,p^*}.$$

Step 3. Update x_p and y_s as follows:

(B1) x_p does not change for $p \neq p^*$, and

$$x_{p^*} \leftarrow x_{p^*} + \frac{1}{a_{s^*,p^*}}.$$

(B2) y_s does not change for $s \notin p_1^* \cup p_2^*$, and

$$y_s \leftarrow y_s \left(1 + \theta \frac{a_{s,p^*}}{a_{s^*,p^*}}\right)$$

for $s \in p_1^* \cup p_2^*$ where θ is a constant chosen later.

The following lemmas give two important properties at the end of above algorithm.

Lemma 6.2.1. *At the end of Primal-Dual Algorithm DPWW,* $(x_p, p \in \mathcal{C})$ *may not be a primal-feasible solution. However,* $(x_p/\tau, p \in \mathcal{C})$ *is a primal-feasible solution where* $\tau = \frac{(v/u)\ln\frac{1+\theta}{v\delta}}{\ln(1+\theta v/u)}$.

Proof. Note that when y_s gets updated, the following facts must hold:

(a) $(y_s, s \in S)$ is not dual feasible.
(b) $s \in p_1^* \cup p_2^*$.

It follows immediately from (a) that $\sum_{s\in S} a_{s,p^*} y_s < 1$, which together with (b) yields that $y_s < 1/v$ before y_s receives any value change. After y_s is updated, we have

$$y_s < \left(1 + \theta \frac{a_{s,p^*}}{a_{s^*,p^*}}\right)/v \leq (1+\theta)/v.$$

Therefore, at the end of Primal-Dual Algorithm DPWW, $y_s < (1+\theta)/v$.

Now, consider a constraint in the primal linear programming,

$$\sum_{p\in\mathcal{C}} a_{s,p} x_p \leq 1,$$

which may not be satisfied after x_p is updated. If updating x_p increases the value of $\sum_{p\in\mathcal{C}} a_{s,p} x_p$ by adding $\frac{a_{s,p^*}}{a_{s^*,p^*}}$, then the value of y_s is increased by multiplying a factor $1 + \theta \frac{a_{s,p^*}}{a_{s^*,p^*}}$. Note that the value of $\frac{a_{s,p^*}}{a_{s^*,p^*}}$ has only two possibilities, v/u and 1. Suppose $\frac{a_{s,p^*}}{a_{s^*,p^*}}$ takes value v/u for k times and 1 for ℓ times. Then the value of $\sum_{p\in\mathcal{C}} a_{s,p} x_p$ receives an increase in $k(v/u) + \ell$ and

$$(1+\theta v/u)^k (1+\theta)^\ell \leq \frac{1+\theta}{v\delta}$$

since initially $y_s = \delta$. Moreover, initially, $\sum_{p\in\mathcal{C}} a_{s,p}x_p = 0$. Thus, at the end of Primal-Dual Algorithm DPWW, the value of $\sum_{p\in\mathcal{C}} a_{s,p}x_p$ is $k(v/u)+\ell$. The maximum value of $k(v/u)+\ell$ can be obtained from the following linear programming with respect to k and ℓ:

$$\begin{aligned} \max \quad & k(v/u)+\ell \\ \text{subject to} \quad & k\ln(1+\theta v/u)+\ell\ln(1+\theta) \leq \ln\frac{1+\theta}{v\delta} \\ & k \geq 0, \ell \geq 0. \end{aligned}$$

By theory of the linear programming, the maximum value of objective function can always be achieved by some extreme point. For above one, the feasible domain has three extreme points

$$(0,0), \quad \left(0, \frac{\ln\frac{1+\theta}{v\delta}}{\ln(1+\theta)}\right), \quad \left(\frac{\ln\frac{1+\theta}{v\delta}}{\ln(1+\theta v/u)}, 0\right).$$

Their objective function values are

$$0, \quad \frac{\ln\frac{1+\theta}{v\delta}}{\ln(1+\theta)}, \quad \frac{v}{u}\cdot\frac{\ln\frac{1+\theta}{v\delta}}{\ln(1+\theta v/u)},$$

respectively. Note that $\frac{z}{\ln(1+\theta z)}$ is strictly monotone decreasing for $z \leq 1$. Thus,

$$0 < \frac{\ln\frac{1+\theta}{v\delta}}{\ln(1+\theta)} < \frac{v}{u}\cdot\frac{\ln\frac{1+\theta}{v\delta}}{\ln(1+\theta v/u)}.$$

Hence, at the end of Primal-Dual Algorithm DPWW,

$$\sum_{p\in\mathcal{C}} a_{s,p}x_p \leq \tau = \frac{v}{u}\cdot\frac{\ln\frac{1+\theta}{v\delta}}{\ln(1+\theta v/u)}.$$

Therefore,

$$\sum_{p\in\mathcal{C}} a_{s,p}x_p/\tau \leq 1.$$

□

Lemma 6.2.2. *At the end of Primal-Dual Algorithm DPWW,*

$$\sum_{p\in\mathcal{C}} x_p/\tau \geq \frac{\ln(v|S|\delta)^{-1}}{\tau\theta\rho}\cdot \mathrm{opt}_{\mathrm{lcc}}$$

where opt_{lcc} *is the objective function value of optimal solution for* MAX-LIFETIME CONNECTED COVERAGE *with two active phases and* $\tau = (v/u)\log_{1+\theta v/u}\frac{1+\theta}{\delta v}$.

Proof. Denote by $x_p(0)$ the initial value of x_p and by $y_s(0)$ the initial value of y_s. Denote by $x_p(i)$ and $y_s(i)$, respectively, the values of x_p and y_s after the ith iteration. Denote by $s^*(i)$ and $p^*(i)$, respectively, the values of s^* and p^* in the ith iteration. Furthermore, denote $X(i) = \sum_{p\in\mathcal{C}} x_p(i)$ and $Y(i) = \sum_{s\in S} y_s(i)$. Then, for $i \geq 1$, one has

$$\begin{aligned} Y(i) &= \sum_{s\in S} y_s(i-1) + \theta \frac{1}{a_{s^*(i),p^*(i)}} \sum_{s\in S} a_{s,p^*(i)} y_s(i-1) \\ &\leq Y(i-1) + \theta (X(i) - X(i-1))\rho \min_{p\in\mathcal{C}} \sum_{s\in S} a_{s,p} y_s(k-1). \end{aligned}$$

Thus,

$$Y(i) \leq Y(0) + \theta\rho \sum_{k=1}^{i} ((X(k) - X(k-1)) \min_{p\in\mathcal{C}} \sum_{s\in S} a_{s,p} y_s(k-1).$$

By the duality theory of linear programming, opt_{lcc} is also the objective function value of optimal solution for the dual linear programming. Therefore,

$$\text{opt}_{\text{lcc}} = \min_{y_s} \frac{\sum_{s\in S} y_s}{\min_{p\in\mathcal{C}} \sum_{s\in S} a_{s,p} y_s},$$

where the minimization is subject to $y_s \geq 0$ for $s \in S$. Hence,

$$\min_{p\in\mathcal{C}} \sum_{s\in S} a_{s,p} y_s(k-1) \leq \frac{Y(k-1)}{\text{opt}_{\text{lcc}}}.$$

Therefore,

$$Y(i) \leq |S|\delta + \frac{\theta\rho}{\text{opt}} \sum_{k=1}^{i} (X(k) - X(k-1)) Y(k-1).$$

Define

$$w(0) = |S|\delta$$

and

$$w(i) = |S|\delta + \frac{\theta\rho}{\text{opt}} \sum_{k=1}^{i} (X(k) - X(k-1)) w(k-1).$$

It is easy to prove by induction on i that $Y(i) \leq w(i)$. Moreover,

$$\begin{aligned} w(i) &= \left(1 + \frac{\theta\rho}{\text{opt}_{\text{lcc}}}(X(i) - X(i-1))\right) w(i-1) \\ &\leq \mathrm{e}^{\frac{\theta\rho}{\text{opt}_{\text{lcc}}}(X(i)-X(i-1))} w(i-1) \end{aligned}$$

$$\leq \mathrm{e}^{\frac{\theta\rho}{\mathrm{opt}_{\mathrm{lcc}}}X(i)} w(0)$$
$$= \mathrm{e}^{\frac{\theta\rho}{\mathrm{opt}_{\mathrm{lcc}}}X(i)} |S|\delta.$$

Suppose Primal-Dual Algorithm DPWW stops at the mth iteration. Then $Y(m) \geq 1/v$. Hence

$$1/v \leq Y(m) \leq w(m) \leq |S|\delta \mathrm{e}^{\frac{\theta\rho}{\mathrm{opt}_{\mathrm{lcc}}}X(m)}.$$

Therefore,

$$\frac{\mathrm{opt}_{\mathrm{lcc}}}{X(m)/\tau} \leq \frac{\tau\theta\rho}{\ln(v|S|\delta)^{-1}}.$$ □

Theorem 6.2.3 (Du et al. [37]). *If* MINW-CSC *with two active phases has a polynomial-time ρ-approximation, then* MAX-LIFETIME CONNECTED COVERAGE *with two active phases has a polynomial-time $\rho(1+\varepsilon)$-approximation for any $\varepsilon > 0$.*

Proof. Choose $\delta = (1+\theta)((1+\theta)|S|)^{-\theta}/v$. Note that

$$\frac{\ln \frac{1+\theta}{\delta v}}{\ln(\delta v|S|)^{-1}} = \frac{1}{1-\theta},$$

and $(1+\theta v/u)^{u/(v\theta)+1} > e$ implies $\ln(1+\theta v/u) > \frac{v\theta}{u+v\theta}$. Thus,

$$\frac{\tau\theta\rho}{\ln(v|S|\delta)^{-1}} = \frac{\theta\rho}{(1-\theta)\ln(1+\theta v/u)} \leq \rho \cdot \frac{1+\theta v/u}{1-\theta}.$$

Choose θ such that

$$\frac{1+\theta v/u}{1-\theta} < 1+\varepsilon.$$

Then

$$\frac{\mathrm{opt}}{\sum_{p\in\mathcal{C}} x_p/\tau} \leq (1+\varepsilon)\rho.$$

To estimate the running time of Primal-Dual Algorithm DPWW, let p^* be a polynomial time ρ-approximation solution for MINW-CSC with Two Active Phases. Note that every iteration can be carried out in polynomial-time. Therefore, it suffices to estimate the number of iterations. Note that at each iteration, at least one of y_s has its value increased. In the proof of Lemma 6.2.1, it is already proved that at the end of the algorithm, each y_s has its value increased by multiplying at most $\log_{1+\theta v/u} \frac{1+\theta}{\delta v}$. Therefore, the number of iterations is at most

$$|S|\log_{1+\theta v/u} \frac{1+\theta}{\delta v} = \frac{|S|\theta \ln((1+\theta)|S|)}{\ln(1+\theta v/u)} = O(|S|\log|S|),$$

where $\delta v = (1+\theta)((1+\theta)|S|)^{-\theta}$ and θ is fixed as ε is fixed. □

In Chap. 5, it has been shown that there exists a polynomial-time 3.63-approximation for MINW-DS. This result can be extended to the following problem.

> MINW-SENSOR-COVER: Consider a set of targets and a set of sensors lying in the Euclidean plane. Suppose all sensors have the same sensing radius R_s, but may have different weights. The problem is to find the minimum weight subset of sensors for covering all targets.

Therefore, the following holds.

Theorem 6.2.4 (Du et al. [37]). MAX-LIFETIME CONNECTED COVERAGE *with Two Active Phases has polynomial-time* $(7.105+\varepsilon)$*-approximations for any* $\varepsilon > 0$ *when all targets and all sensors lie in the Euclidean plane and all sensors are uniform with* $R_c \geq 2R_s$.

Proof. Let $\mathrm{Opt}_{\mathrm{CSC}}$ be the optimal solution for MINW-CSC with two active phases. Compute a polynomial-time 3.63-approximation solution A for MINW-SENSOR-COVER with weight $y_s u$ for each sensor s. Then

$$\sum_{s\in A} y_s u \leq 3.63 \cdot \mathrm{opt}_{\mathrm{CSC}},$$

where $\mathrm{opt}_{\mathrm{CSC}}$ is the objective function value of $\mathrm{Opt}_{\mathrm{CSC}}$. Since $R_c \geq 2R_s$, every sensor in A is adjacent to some sensor in $\mathrm{Opt}_{\mathrm{CSC}}$. This means that $\mathrm{Opt}_{\mathrm{CSC}} \cup A$ induces a connected subgraph and hence $\mathrm{Opt}_{\mathrm{CSC}}$ contains the set of Steiner nodes in a feasible solution for NODE-WEIGHTED STEINER TREE on the terminal set A. Now, find a polynomial-time 3.475-approximation solution B for NODE-WEIGHTED STEINER TREE with weight $y_s v$ for each sensor s. Then

$$\sum_{s\in B} y_s v \leq 3.475 \cdot \sum_{s\in Opt_{\mathrm{CSC}}} y_s v \leq 3.475 \cdot \mathrm{opt}_{\mathrm{CSC}}.$$

Therefore,

$$\sum_{s\in A} y_s u + \sum_{s\in B} y_s v \leq 7.105 \cdot \mathrm{opt}_{\mathrm{CSC}}.$$

□

6.3 Domatic Partition

So far, the best known constant-approximation for MAX#DS in unit disk graphs is designed also using grid partition, however with a new technique. Let us start to introduce a problem on sensor-cover-partition with a separating line.

> SENSOR-COVER-PARTITION with Separating Line: Let L be a horizontal line. Given a set T of targets above L and a set S of sensors with sensing radius one below L, assume that every target is covered by at least one sensor. The problem is to find the maximum number of disjoint sensor covers. (A sensor cover is a subset of sensors covering all targets.)

Let $\delta(S,T) = \min_{t\in T} |\{s \in S \mid t \in \text{disk}_1(s)\}|$ where $\text{disk}_1(s)$ denotes the disk with radius one and the center s. Call as the *skyline* the part, above line L, of envelope of disks $\text{disk}_1(s)$ for all $s \in S$. Let S' be the set of those sensors s such that $\text{circle}_1(s)$ has a piece appearing in the skyline where $\text{circle}_1(s)$ denotes the circle with radius one and the center s. By Lemma 5.7.1 S' lines up from right to left by following their pieces on the skyline. For any $t \in T$, denote $C_{S'}(t) = \text{disk}_1(t) \cap S'$. The following properties are important.

Lemma 6.3.1. *Let s_1, s_2, s_3 be three sensors in S with $s_1.x \leq s_2.x \leq s_3.x$ where $s_i.x$ denotes the x-coordinate of point s_i. Suppose there exists a target t such that $t \in \text{disk}_1(s_1) \cap \text{disk}_1(s_3)$ but $t \notin \text{disk}_2(s_2)$. Then $\text{up}(L) \cap \text{disk}_1(s_2) \subseteq \text{up}(L) \cap (\text{disk}_1(s_1) \cup \text{disk}_1(s_3))$ where $\text{up}(L)$ denotes the half plane above the horizontal line L and $\text{circle}_1(s_2)$ cannot appear in the skyline.*

Proof. It is trivial in the case that $s_1.x = s_2.x$ or $s_2.x = s3.x$. Thus, we next assume $s_1.x < s_2.x < s_3.x$. For contradiction, suppose there exists a point $p \in \text{up}(L) \cap \text{disk}_1(s_2)$ but $p \notin \text{up}(L) \cap (\text{disk}_1(s_1) \cup \text{disk}_1(s_3)$. Note that $t \in \text{disk}_1(s_1) \cap \text{disk}_1(S_3)$ implies that for any point $q \in \text{up}(L)$ with $q.x = t.x$ and $q.y \leq t.y$, $q \in \text{disk}_1(s_1) \cap \text{disk}_1(S_3)$. Moreover, $t \notin \text{disk}_1(s_2)$ implies that for any $q \in \text{up}(L) \cap \text{disk}_1(s_2)$ with $q.x = t.x$ and $q.y < t.y$ and hence $q \in \text{disk}_1(s_1) \cap \text{disk}_1(S_3)$. It follows that $p.x \neq t.x$. Hence $p.x < t.x$ or $p.x > t.x$. First, consider the case that $p.x < t.x$. In this case, two segments ps_2 and ts_1 must intersect at a point o. Note that $|ps_2| < |ps_1|$ and $|ts_2| > |ts_1|$. Hence, $|ps_2| + |ts_1| < |ps_1| + |ts_2|$. However, by the property of the triangle,

$$|po| + |os_1| \geq |ps_1|$$

and

$$|to| + |os_2| \geq |ts_2|.$$

Therefore

$$|ps_2| + |ts_1| = |po| + |os_2| + |to| + |os_1| \geq |ps_1| + |ts_2|,$$

a contradiction. Similarly, a contradiction can result from the case that $p.x > t.x$.

Note that $\text{circle}_1(s_2) \cap \text{up}(L)$ cannot intersect $\text{up}(L) \cap (\text{circle}_1(s_1) \cup \text{circle}_1(s_3))$. In fact, if they have an intersection point p, then a contradiction can still result from the above argument by noting that the argument still works when $|ps_2| = |ps_1|$. So, $\text{up}(L) \cap \text{disk}_1(s_2)$ is contained strictly inside of $\text{up}(L) \cap (\text{disk}_1(s_1) \cup \text{disk}_1(s_3))$. Hence, $\text{circle}_1(s_2)$ cannot appear in the skyline. □

Lemma 6.3.2. *For any $t \in T$, $C_{S'}(t)$ is a nonempty contiguous subset of the ordered set S'.*

Proof. Suppose $s_1, s_2, s_3 \in S'$ with $s_1.x \leq s_2.x \leq s_3.x$. If $s_1, s_3 \in C_{S'}(t)$ and $s_2.x \notin C_{S'}(t)$, then by Lemma 6.3.1, $s_2 \notin S'$, a contradiction. □

Lemma 6.3.3. *Suppose T' is a subset of targets, satisfying a property that for any two distinct targets $t,t' \in T'$, $C_S(t) \not\subseteq C_S(t')$. Then for any two distinct $t,t' \in T'$, $C_{S'}(t) \cap C_{S'}(t') \neq \emptyset$ implies that $C_{S'}(t)$ contains an endpoint of $C_{S'}(t')$.*

Proof. The lemma holds trivially in the case that $C_{S'}(t)$ is not contained in $C_{S'}(t)$. So, we next assume $C_{S'}(t) \subseteq C_{S'}(t)$. By the assumption on T', there exists $s \in C_S(t) \setminus C_S(t')$. Let s_r and s_l be the right endpoint and the left endpoint of $C_{S'}(t')$. Let $s' \in C_{S'}(t) \cap C_{S'}(t')$. Next, consider two cases.

Case 1. $s_l.x \leq s.x \leq s_r.x$. Note that t' is contained in $\mathrm{disk}_1(s_r)$ and $\mathrm{disk}_1(s_l)$ but not contained in $\mathrm{disk}_1(s)$. By Lemma 6.3.1, $t \in \mathrm{up}(L) \cap \mathrm{disk}_s \subseteq \mathrm{up}(L) \cap (\mathrm{disk}_1(s_l) \cup \mathrm{disk}_1(s_r))$. Therefore, $s_l \in C_{S'}(t)$ or $s_r \in C_{S'}(t)$.

Case 2. $s_1.x > s.x$ or $s_r.x < s.x$. Note that $s_l.x \leq s'.x \leq s_r.x$. For contradiction, suppose t is contained by neither $\mathrm{disk}_1(s_l)$ nor $\mathrm{disk}_1(s_r)$. In the case that $s.x < s_1.x$, t is contained by $\mathrm{disk}_1(s)$ and $\mathrm{disk}_1(s')$, but not contained by $\mathrm{disk}_1(s_l)$. By Lemma 6.3.1, $s_l \notin S'$, a contradiction. Similarly, a contradiction can result from the case that $s_r.x < s.x$. □

Now, it is ready to show the following.

Theorem 6.3.4. *There is a polynomial-time algorithm which can find at least $\delta(S,T)/4$ disjoint sensor covers.*

Proof. Consider the following algorithm.

The DomPart Algorithm.
input: a sensor set S and a target set T.
$j \leftarrow 0$;
$E \leftarrow S$;
while E is a set cover **do begin**
1. $j \leftarrow j+1$;
2. $T' \leftarrow T$;
   while there exist $t,t' \in T'$ such that $C_E(t) \subseteq C_E(t')$
   do $T' \leftarrow T' \setminus \{t'\}$;
3. Let $E' \subseteq E$ contribute the skyline of disks at E;
4. Find a maximal subset T'' of T' such that $C_{E'}(t)$ for $t \in T''$ are disjoint;
5. $A_j = \{\text{two endpoints of } C_{E'}(t) \mid t \in T''\}$;
6. $E \leftarrow E \setminus A_j$;

end-while
output: $A_1, A_2, \ldots, A_j$.

First, we show that each A_i for $i = 1,\ldots,j$ is a sensor cover. In fact, for each $t'' \in T''$, A_i contains two endpoints of $C_{E'}(t'')$ and hence t'' is covered by A_i. For $t' \in T' \setminus T''$, there exists $t'' \in T''$ such that $C_{E'}(t') \cap C_{E'}(t'') \neq \emptyset$. By Lemma 6.3.3, $C_{E'}(t')$ contains an endpoint of $C_{E'}(t'')$ and hence t' is covered by A_i. For $t \in T \setminus T'$, there exists $t' \in T'$ such that $C_E(t') \subseteq C_E(t)$. So, there exists $t'' \in T''$ such that $C_E(t)$ contains an endpoint of $C_{E'}(t'')$ and hence t is covered by A_i.

Next, we show that at the end of the jth iteration, $|C_E(t)| \geq \delta(S,T) - 4j$ for every $t \in T$. To do so, let E_j denote the E at the end of the jth iteration. Suppose this inequality holds at the end of the $(j-1)$th iteration, that is, $|C_{E_{j-1}}(t)| \geq \delta(S,T) - 4(j-1)$ for all $t \in T$. We show that $|C_{E_j}(t)| \geq \delta(S,T) - 4j$ for all $t \in T$.

In the jth iteration, for $t'' \in T''$, two endpoints of $C_{E'}(t'')$ are deleted from E_{j-1} and hence

$$|C_{E_j}(t'')| \geq |C_{E_{j-1}}(t'')| - 2 > \delta(S,T) - 4j.$$

For $t' \in T' \setminus T''$, if $C_{E'}(t')$ contains an endpoint of $C_{E'}(t'')$ for $t'' \in T''$, then by Lemma 6.3.3, $C_{E'}(t'')$ must contain an endpoint of $C_{E'}(t')$. Thus, there are at most two such t'''s because all $C_{E'}(t'')$ for $t'' \in T''$ are disjoint. This means that

$$|C_{E_j}(t')| \geq |C_{E_{j-1}}(t')| - 4 \geq \delta(S,T) - 4j.$$

For $t \in T \setminus T'$, there exists $t' \in T'$ such that $C_{E_{j-1}}(t') \subseteq C_{E_{j-1}}(t)$. This relationship is preserved in the algorithm, that is, $C_{E_j}(t') \subseteq C_{E_J}(t)$. Therefore,

$$|C_{E_j}(t)| \geq |C_{E_j}(t')| \geq \delta(S,T) - 4j.$$

It follows immediately from this inequality that at the end of The DomPart Algorithm, $j \geq \delta(S,T)/4$. $\square$

With Theorem 6.3.4, Pandit et al. [86] constructed an algorithm for MAX#DS in unit disk graphs as follows.

Put input unit disk graph $G = (V,E)$ into a square and partition the square with a grid of cells with diameter one (or say, diagonal length one). A cell is called a *heavy* cell if it contains at least $\delta/14$ nodes where $\delta^{\min}$ is the minimum node degree of G. A cell is *light* if it is not heavy. For each node v in a light cell, $\mathrm{disk}_1(v)$ intersects at most 14 cells, at least one of which contains at least $\delta^{\min}/14$ nodes adjacent to v. Choose such a heavy cell σ and put v to T^σ, say that v belongs to σ. Let $S^\sigma = \sigma \cap V$. Consider S^σ as a sensor set and T^σ as a target set. Then the following lemma gives an important fact.

Lemma 6.3.5. *If for every heavy cell σ, S^σ can be partitioned into k sensor covers for T^σ, then G has k disjoint dominating sets.*

Proof. Choose a sensor cover A^σ for each heavy cell σ. Let A be the union of A^σ for σ over all heavy cells. Then A is a dominating set because each A^σ dominates not only all nodes in T^σ, but also dominates all nodes in S^σ. $\square$

For each heavy cell σ, partition T^σ into four parts $(T^\sigma_{\mathrm{north}}, T^\sigma_{\mathrm{south}}, T^\sigma_{\mathrm{east}}, T^\sigma_{\mathrm{west}})$ where $T^\sigma_{\mathrm{north}}$ consists of nodes lying above the line through the upper bound of σ, $T^\sigma_{\mathrm{south}}$ consists of nodes lying below the line through the lower bound of σ, T^σ_{east} consists of nodes lying in the right of the line through the right bound of σ, and T^σ_{west} consists of nodes lying in the left of the line through the left bound of σ. When two parts are available for a node v in T^σ, v can arbitrarily choose one of them as its home. Corresponding these four parts, partition S^σ also into four parts $(S^\sigma_{\mathrm{north}}, S^\sigma_{\mathrm{south}}, S^\sigma_{\mathrm{east}}, S^\sigma_{\mathrm{west}})$ by independently and randomly distributing each node into these four parts.

Now, solve SENSOR-COVER-PARTITION with separation line on four inputs $(S^\sigma_{\text{north}}, T^\sigma_{\text{north}})$, $(S^\sigma_{\text{south}}, T^\sigma_{\text{south}})$, $(S^\sigma_{\text{east}}, T^\sigma_{\text{east}})$, and $(S^\sigma_{\text{west}}, T^\sigma_{\text{west}})$. Combine those solutions into k disjoint dominating sets of G where

$$k = \min\{\delta(S^\sigma_{\text{south}}, T^\sigma_{\text{south}}), \delta(S^\sigma_{\text{east}}, T^\sigma_{\text{east}}), \delta(S^\sigma_{\text{west}}, T^\sigma_{\text{west}}) \mid \text{all heavy cells } \delta\}.$$

Next, we show that $k \geq \delta^{\min}/112$ with a quite high probability.

Note that for each $t \in T^\sigma$, $|\sigma \cap \text{disk}_1(t)| \geq \delta^{\min}/14$ and the probability of at least one of two nodes in $\sigma \cap \text{disk}_1(t)$ distributed in the part containing t is $3/4$. By Chernoff bound, the probability of at least $\delta^{\min}/56$ nodes in $\sigma \cap \text{disk}_1(t)$ distributed in the part containing t is at least $1 - \mathrm{e}^{-\delta^{\min}/112}$.

Note that for each heavy cell σ, there are at most 20 cells within distance one to σ. So, there are at most 20 light cells which contain a node belonging to σ. Hence, $|T^\sigma| \leq (20/14)\delta^{\min}$. Thus, the probability of the following held is at least $1 - (20/14)\delta^{\min}\mathrm{e}^{-\delta^{\min}/112}$:

$$\min(\delta(S^\sigma_{\text{south}}, T^\sigma_{\text{south}}), \delta(S^\sigma_{\text{east}}, T^\sigma_{\text{east}}), \delta(S^\sigma_{\text{west}}, T^\sigma_{\text{west}})) \geq \delta^{\min}/56.$$

Since the number of heavy cells cannot be bounded by $O(\delta^{\min})$, it is hard to estimate the probability of $k \geq \delta^{\min}/56$. Thus, it requires more efforts on distribution of each element of S^σ in order to establish a solution of the following problem.

Open Problem 6.3.6. *Is there a polynomial-time algorithm which produces $\Omega(\delta^{\min})$ disjoint dominating sets for G with high probability?*

6.4 Min-Weight Dominating Set

Pandit et al. [86] gave an interesting idea to construct approximation algorithms for MINW-DS using algorithm for MAX#DS.

Consider the following LP-relaxation of MINW-DS.

$$\begin{aligned} \min \quad & \sum_{i \in V} c_i x_i \\ \text{subject to} \quad & \sum_{i \in \text{disk}_1(j)} x_i \geq 1 \text{ for all } j \in V \\ & x_i \geq 0 \text{ for all } i \in V. \end{aligned}$$

Let $(x_i^*, i \in V)$ be an optimal solution of this LP. Denote $n = |V|$. Let

$$\bar{x}_i = \begin{cases} 0 & \text{if } x_i^* \leq 1/2n \\ \frac{k}{2n} & \text{if } \frac{k-1}{2n} < x_i^* \leq \frac{k}{2n}. \end{cases}$$

Lemma 6.4.1. *The following holds:*

(1) For $j \in V$, $\sum_{i \in \text{disk}_1(j)} \bar{x}_i \geq 1/2$.
(2) $\sum_{i \in V} c_i \bar{x}_i \leq 2 \cdot \text{opt}_{\text{WDS}}$ *where* opt_{eds} *is the objective function value of an optimal solution for* MINW-DS.

Proof. Since $|V \cap \text{disk}(j)| \leq n$, there are at most n x_i^* are rounded down to 0. Therefore,

$$\sum_{i \in \text{disk}_1(j)} \bar{x}_i \geq 1 - n \cdot \frac{1}{2n} = 1/2.$$

This means that (1) holds. For (2), note that

$$\sum_{i \in V} c_i \bar{x}_i \leq 2 \sum_{i \in \text{disk}_1(j)} c_i x_i^* \leq 2 \cdot \text{opt}_{\text{WDS}}. \qquad \square$$

Construct a set P by making $2n \cdot \bar{x}_j$ copies of node j for each $j \in V$. Suppose each copy of j has the same weight as that of j.

Lemma 6.4.2. $c(P) \leq 4n \cdot \text{opt}_{\text{WDS}}$.

Proof. By Lemma 6.4.1, $c(P) = 2n \cdot \sum_{i \in V} c_i \cdot \bar{x}_i \leq 4n\text{opt}_{\text{WDS}}$. $\square$

Lemma 6.4.3. $\delta(P, V) \geq n$.

Proof. By Lemma 6.4.1, $\sum_{i \in \text{disk}_1(j} \bar{x}_i \geq 1/2$. Thus, for each $j \in V$, $|P \cap \text{disk}_1(j)| = 2n \sum_{i \in \text{disk}_1(j)} \bar{x}_i \geq n$. $\square$

Suppose there is an algorithm which can produce at least $\delta(P, V)/C$ sensor cover packing $A_1, \ldots, A_t$ ($t \geq n/C$) for sensor set P and target set V. Then there exists A_i such that

$$c(A_i) \leq \frac{c(P)}{t} \leq \frac{C \cdot c(P)}{n} \leq \frac{4Cn \cdot \text{opt}_{\text{WDS}}}{n} = 4C \cdot \text{opt}_{\text{WDS}}.$$

This means that the following holds.

Theorem 6.4.4. *If there is a polynomial-time algorithm for* SENSOR-COVER-PARTITION *which can produce* $\delta(P, V)/C$ *sensor covers for sensor set* P *and target set* V*, then there is a polynomial-time* $4C$*-approximation for* MINW-DS.