Supplemental Materials

Xiangmao Chang, Rui Tan, Guoliang Xing, Zhaohui Yuan, Chenyang Lu, Yixin Chen, Yixian Yang

This document includes the supplemental materials for the paper titled "Sensor Placement Algorithms for Fusion-based Surveillance Networks".

Appendix A
Hardness of Problem 2

In this appendix, we discuss the hardness of Problem 2 presented in Section 4.2. Specifically, we discuss the non-linearity and non-convexity of the problem. We note that non-linear and non-convex optimization problems often have exponential complexities and the general solution to them does not exist [1]. In the constraint of Problem 2, the detection probability P_{D_j}, which is given by (3), has a complex non-linear relationship with the number of sensors and their positions. Therefore, Problem 2 is a non-linear optimization problem. We now discuss the non-convexity of Problem 2. Denote the placement with n sensors as $\mathbf{S}_n$. Thus, $|\mathbf{S}_n| = n$. By replacing P_{D_j} in the constraint (*i.e.*, $\min_{1 \le j \le K}\{P_{D_j}\} \ge \beta$) with (3) and after manipulation, we have

$$\min_{1 \le j \le K} \sum_{i=1}^{n_j} W(d_i) \ge \sigma^2(\chi^{-1}_{n_j}(1-\alpha) - \chi^{-1}_{n_j}(1-\beta)).$$

where n_j is the number of sensors in the fusion range of spot j. Denote $f(\mathbf{S}_n) = \min\limits_{1 \le j \le K} \sum_{i=1}^{n_j} W(d_i) - \sigma^2(\chi^{-1}_{n_j}(1-\alpha) - \chi^{-1}_{n_j}(1-\beta))$. Suppose $\mathbf{S}^*_n$ maximizes $f(\mathbf{S}_n)$, if $f(\mathbf{S}^*_n) \ge 0$, $\mathbf{S}^*_n$ is a candidate solution. By iterating n, we can find the optimal solution to Problem 2, *i.e.*, $\mathbf{S} = \arg\min_{\mathbf{S}^*_n} n$. If the sub-problem, *i.e.*, maximizing $f(\mathbf{S}_n)$, is non-convex, Problem 2 is non-convex. We now prove the non-convexity of $f(\mathbf{S}_n)$ by exemplification. In our example, we randomly choose two surveillance spots in a 10×10 field, with the coordinates of $(8.21, 0.15)$ and $(0.43, 1.69)$, respectively. We then randomly place 3 or 4 sensors in the field. When $n = 3$, the coordinates of sensors are $(6.49, 4.51)$, $(7.32, 5.47)$ and $(6.48, 2.96)$, respectively. When $n = 4$, the coordinates of sensors are $(7.45, 3.68)$, $(1.89, 6.26)$, $(6.87, 7.80)$ and $(1.84, 0.81)$, respectively. Fig. A.1 plots $f(\mathbf{S}_n)$ versus the X-coordinate of the first sensor. We can see from the figure that $f(\mathbf{S}_n)$ is a non-convex function with respect to sensor position. In summary, Problem 2 is a non-linear and non-convex optimization problem.

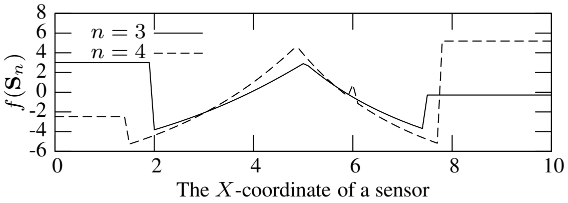


Fig. A.1. $f(\mathbf{S}_n)$ is non-convex. Settings: $W_0 = 400$, $d_0 = 1$, $k = 2$, $\sigma^2 = 1$, $\alpha = 0.01$, $\beta = 0.9$.

Algorithm 1 The procedure of finding global optimal solution

```
Input: $\alpha$, $\beta$, surveillance field $A$, a set of surveillance spots $\mathbf{T}$
Output: Sensor placement $\mathbf{S}$ where $|\mathbf{S}|$ is minimized
$N = 1$
repeat
    use the CSA solver to find $N$ sensor locations in $A$ that maximize $\min_{1 \le j \le K}\{P_{D_j}\}$
    compute $\eta_j$ for each $t_j \in \mathbf{T}$ by $\eta_j = \frac{\sigma^2 \mathcal{X}^{-1}_{n_j}(1-\alpha)}{n_j}$
    compute $P_{D_j}$ for each $t_j \in \mathbf{T}$ by (3)
    $N = N + 1$
until $\min_{1 \le j \le K}\{P_{D_j}\} \ge \beta$
return $\mathbf{S}$
```

Appendix B
Global Optimal Placement

A straightforward optimal solution for Problem 2 is to incrementally search for the optimal sensor placements with different number of sensors under the constraint $\min_{1 \le j \le K}\{P_{D_j}\} \ge \beta$. The details are shown in Algorithm 1. It begins with $N = 1$ and iterates for incremental N. In each iteration, the minimum detection probability among all surveillance spots, *i.e.*, $\min_{1 \le j \le K}\{P_{D_j}\}$, is

- *X. Chang and Y. Yang are with Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China; Y. Yang is also with Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China.*
- *R. Tan and G. Xing are with Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA.*
- *Z. Yuan is with School of Software, Huadong Jiao Tong University, Nanchang, Jiangxi 310013, China.*
- *C. Lu and Y. Chen are with Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130, USA.*

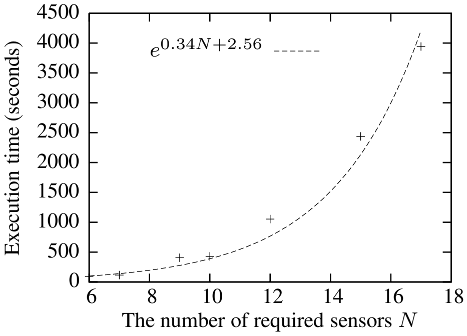


Fig. B.1. The execution time vs. the number of sensors in the optimal placement.

maximized. Once the constraint $\min_{1 \le j \le K}\{P_{D_j}\} \ge \beta$ is satisfied, the global optimal solution is found.

The optimization step that maximizes the minimum detection probability (Line 3 in Algorithm 1) is implemented by a non-linear programming solver based on the Constrained Simulated Annealing (CSA) algorithm [2]. CSA extends conventional Simulated Annealing to look for the global optimal solution of a constrained optimization problem with discrete variables. CSA allows the objective function and constraint functions to be specified in a procedure instead of in a closed-form. Theoretically, CSA is a global optimal algorithm that converges asymptotically to a constrained global optimum with probability one (Theorem 1 of [2]).

In theory, the complexity of CSA, like other stochastic search algorithms, increases exponentially with respect to the number of variables [2]. Therefore, the non-linear programming solver has an exponential time complexity with respect to the number of sensors. Hence, for a large-scale placement problem, Algorithm 1 becomes prohibitively expensive.

Fig. B.1 shows the execution time of Algorithm 1 versus the number of sensors in the optimal solution. The dotted curve is the linear regression of the execution times with different numbers of sensors. We can see that the execution time increases drastically with the number of sensors. For instance, if 100 sensors are to be placed, the projected execution time of the global optimal algorithm is about e^{36} seconds, *i.e.*, 5×10^{10} days.

Appendix C
Divide-and-Conquer Sensor Placement Algorithm

The pseudo codes of the divide-and-conquer sensor placement algorithm and the refinement process proposed in Section 5.2 are shown in Algorithm 2 and Algorithm 3, respectively.

In Algorithm 2, we treat the surveillance spots one by one. Specifically, for spot t_j, we place the fewest additional sensors within the fusion region of t_j to cover t_j and the neighboring spots in its impact region. The optimization (Line 5) is implemented by the CSA solver.

Algorithm 2 The divide-and-conquer sensor placement algorithm

```
Input: $\alpha$, $\beta$, impact region set $\{A_j | 1 \le j \le K\}$, and
    the set of surveillance spots in each impact region
    $\{\mathbf{T}_j | 1 \le j \le K\}$
Output: Local optimal sensor placement $\mathbf{S}$
1:  $\mathbf{S} = \emptyset$
2:  for $j = 1$ to $K$ do
3:      $n = 0$
4:      repeat
5:          place additional $n$ sensors in $C_j$ (denoted by
            set $\Delta$) to maximize $\min_{t_h \in \mathbf{T}_j}\{P_{D_h}\}$ under placement $\mathbf{S} \cup \Delta$
6:          compute $\eta_h$ for each $t_h \in \mathbf{T}_j$ under placement
            $\mathbf{S} \cup \Delta$
7:          compute $P_{D_h}$ for each $t_h \in \mathbf{T}_j$ under placement
            $\mathbf{S} \cup \Delta$
8:          $n = n + 1$
9:      until $\min_{t_h \in \mathbf{T}_j}\{P_{D_h}\} \ge \beta$
10:     $\mathbf{S} = \mathbf{S} \cup \Delta$
11: end for
12: return $\mathbf{S}$
```

In each round of Algorithm 3 (from Line 2 to 20), all surveillance spots are processed one by one. The algorithm terminates if the current round cannot further reduce the number of sensors in the placement. When the j^{th} spot t_j is processed (from Line 4 to 19), the algorithm first removes all dedicated sensors of t_j, and computes a new local placement $\mathbf{S}'$ using the CSA solver (from Line 8 to 15). If $\mathbf{S}'$ uses fewer sensors than the original placement $\mathbf{S}$, we replace the original placement with $\mathbf{S}'$ (from Line 17 to 19). Note that we do not remove any shared sensors in the placement computed by Algorithm 2 as otherwise the coverage of neighboring surveillance spots may be affected.

We now discuss the convergence of Algorithm 3. As only the new placement with fewer sensors (from Line 17 to 19) in each iteration is acceptable, obviously, the size of the placement computed in each round keeps decreasing. Denote N^* and N_0 as the sizes of the global optimal solution (*i.e.*, the output of Algorithm 1) and the solution of Algorithm 2, respectively. The upper bound of the number of rounds of Algorithm 3 is thus $N_0 - N^*$.

Appendix D
Analysis of Model Parameters

In this appendix, we discuss how to set the parameters of the sensing and data fusion models.

D.1 Optimal Upper Bound of Fusion Radius

In this section, we analyze the upper bound of fusion radius R, and give the optimal value regarding sensor density. Suppose a surveillance spot t is covered by n

Algorithm 3 The sensor placement refinement algorithm

```
Input: $\alpha$, $\beta$, impact region set $\{A_j | 1 \le j \le K\}$, the set of
    surveillance spots in each impact region $\{\mathbf{T}_j | 1 \le j \le K\}$,
    sensor placement $\mathbf{S}$ computed by Algorithm 2
Output: new sensor placement $\mathbf{S}$
repeat
  total sensor number $N = |\mathbf{S}|$
  for $j = 1$ to $K$ do
    find dedicated sensor set $D_j$ of $t_j$ in $\mathbf{S}$
    if $D_j = \emptyset$ then
      skip this iteration for $t_j$ and continue
    else
      $\mathbf{S}' = \mathbf{S} \setminus D_j$ /* remove dedicated sensors of $t_j$ */
      $n = 1$
      repeat
        place additional $n$ sensors in $C_j$ (denoted by $\Delta$) to maximize
            $\min_{t_h \in \mathbf{T}_j} \{P_{D_h}\}$ under placement $\mathbf{S}' \cup \Delta$
        compute $\eta_h$ and $P_{D_h}$ for each $t_h \in \mathbf{T}_j$ under
            placement $\mathbf{S}' \cup \Delta$
        $n = n + 1$
      until $\min_{t_h \in \mathbf{T}_j} \{P_{D_h}\} \ge \beta$
      $\mathbf{S}' = \mathbf{S}' \cup \Delta$
    end if
    if $|\mathbf{S}'| < |\mathbf{S}|$ then
      $\mathbf{S} = \mathbf{S}'$
    end if
  end for
until $|\mathbf{S}| = N$
return $\mathbf{S}$
```

sensors in the fusion region of t. According to (3), the detection probability of t satisfies:

$$P_D = 1 - \mathcal{X}_n \left(\frac{n\eta - \sum_{i=1}^{n} W(d_i)}{\sigma^2} \right) \ge \beta.$$

By replacing η with the optimal detection threshold $\eta = \frac{\sigma^2 \mathcal{X}_n^{-1}(1-\alpha)}{n}$ obtained in Section 4.2 and solving the sum of energies $\sum_{i=1}^{n} W(d_i)$ from the previous inequality, we get:

$$\frac{\sum_{i=1}^{n} W(d_i)}{\sigma^2} \ge \mathcal{X}_n^{-1}(1-\alpha) - \mathcal{X}_n^{-1}(1-\beta). \qquad \text{(D.1)}$$

Inequality (D.1) means that given n sensors, the sum of energies received by sensors is lower bounded if the corresponding spot is covered. As the signal energy attenuates with the distance from the target, in the worst case, every sensor is placed at the edge of the fusion region. Therefore, the minimum sum of energies is

$$\min \left\{ \sum_{i=1}^{n} W(d_i) \right\} = nW(R). \qquad \text{(D.2)}$$

By replacing the sum of energies in (D.1) with its minimum value given by (D.2) and solving the fusion radius R, we get:

$$R \le W^{-1} \left(\frac{\sigma^2 \left(\mathcal{X}_n^{-1}(1-\alpha) - \mathcal{X}_n^{-1}(1-\beta) \right)}{n} \right), \qquad \text{(D.3)}$$

where $W^{-1}(\cdot)$ is the inverse function of the signal attenuation model $W(d)$. Denote the right hand of (D.3) as R_u, which is the upper bound of the fusion radius. Inequality (D.3) shows that, given n sensors, if the fusion radius does not exceed R_u, the surveillance spot t is always covered no matter how the sensors are placed in the fusion region of t. If we set $R = R_u$, the sensor density is $\rho = n/(\pi R_u^2)$, which is the minimum sensor density that guarantees the coverage of the surveillance spot no matter how the sensors are placed. Because both ρ and R_u depend on n, it's hard to derive the marginal relationship between ρ and R_u with an analytical expression. However, in practice, we can find the optimal R_u that minimizes ρ by numerical approaches. For instance, Fig. D.1 shows several numerical results under the signal attenuation model defined by (1) under different settings. From the figure, we can see that the sensor density ρ is a convex function of R_u under different settings and thus we can find the optimal R_u, *e.g.*, $R_u = 7.76$ is the optimal value for the setting $\sigma^2 = 1$. Furthermore, for a certain fusion radius (*e.g.*, $R_u = 10$), the sensor density for low SNR setting (*e.g.*, $\sigma^2 = 5$) is greater than the sensor density for high SNR setting (*e.g.*, $\sigma^2 = 1$). This is reasonable as more sensors are needed to achieve the same sensing quality when the SNR is low.

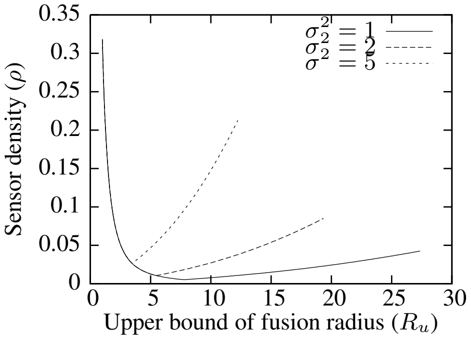


Fig. D.1. Sensor density vs. upper bound of fusion radius. Settings: $W_0 = 400$, $d_0 = 1$, $k = 2$, $\alpha = 0.01$, $\beta = 0.9$.

D.2 Analysis of Impact Region Radius

In this section, we discuss the optmal setting of the impact region radius. As discussed in Section 5.1, the key idea of our divide-and-conquer approach is to reduce the total number of sensors by taking advantage of the shared sensors in the overlapping fusion regions of surveillance spots. Moreover, the global optimization problem is divided into the sub-problems of covering the spots within the impact region of each spot. If the distance between any two spots is less than $2R$, the two spots can have shared sensors. Therefore, in previous sections, we let the *impact region radius* be $2R$. In this section, we discuss the setting of impact region radius in detail, which provides more insights into the divide-and-conquer approach.

We first discuss the impact of placing an additional sensor for the coverage of a surveillance spot. The necessary and sufficient condition of the coverage of a surveillance spot is given by (D.1). If an additional sensor is placed into the fusion region of the spot, both sides of inequality (D.1) increase. We now explore the sufficient condition for maintaining inequality (D.1). We let $f(n|\alpha,\beta) = \mathcal{X}_n^{-1}(1-\alpha) - \mathcal{X}_n^{-1}(1-\beta)$ and $g(n|\alpha,\beta) = f(n+1|\alpha,\beta) - f(n|\alpha,\beta)$, where $n \geq 1$. Our extensive numerical experiments show that $g(n|\alpha,\beta)$ is a monotonically decreasing function if $\alpha < \beta$. Note that mission-critical applications typically require low false alarm rates (*e.g.*, $< 5\%$) and high detection probabilities (*e.g.*, $> 90\%$). Therefore, the maximum value of $g(n|\alpha,\beta)$ is $g(1|\alpha,\beta)$. When an additional sensor is placed in the fusion region of the spot, the minimum increase to the left-hand side of (D.1) is $\frac{W(R)}{\sigma^2}$. Accordingly, if $W(R) \geq \sigma^2 \cdot g(1|\alpha,\beta)$, inequality (D.1) still holds when an additional sensor is placed, and therefore the coverage of the spot will not be breached.

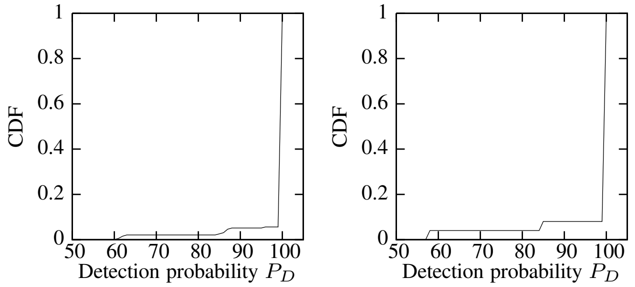
With the above sufficient condition for maintaining coverage when additional sensors are placed, we now discuss the setting of impact region radius. Our discussion has the following two cases:

Case 1): $W(R) < \sigma^2 \cdot g(1|\alpha,\beta)$. Suppose the impact region radius is smaller than $2R$, placing sensors to cover the currently treated spot can breach the coverage of the spots out of the impact region. Therefore, the best choice of impact region radius is $2R$.

Case 2): $W(R) \geq \sigma^2 \cdot g(1|\alpha,\beta)$. In this case, the impact region radius can be smaller than $2R$. However, as the sensor placement is not known *a priori*, it is difficult to derive the optimal impact region radius that minimizes the network density. In this paper, we evaluate the optimal impact region radius through numerical experiments, which are presented in Appendix F.1.

D.3 Handling Target Location Error

The problem formulation in this paper only ensures the detection performance at the surveillance spots. When the target does not appear at the surveillance spots exactly, the detection performance of the computed sensor placement may not meet the user requirement. We now briefly discuss an approach to handle this issue. Suppose the distance between the surveillance spot and the real position of the target, referred to as *target location error*, is upper-bounded by ϵ. By replacing $W(d)$ with $W(d+\epsilon)$, Eq. (3) computes the lower bound of detection probability. Therefore, the resulted sensor placement can meet the detection performance requirement when the target location error does not exceed ϵ.



(a) 196 regular spots

(b) 25 random spots

Fig. E.1. The CDF of the detection probabilities at regular or random surveillance spots.

Appendix E
More Trace-driven Simulation Results

In the trace-driven simulations presented in Section 6, we evaluate the performance of sensor placement for real vehicle trajectories. In order to extensively evaluate the performance of our sensor placement algorithm, in this appendix, we choose regularly or randomly distributed surveillance spots in a $30 \times 30\,\mathrm{m}^2$ area. We evaluate the performance of the cluster-based sensor placement algorithm in two sets of simulations. First, 67 sensors are placed to cover 196 surveillance spots regularly distributed at 14×14 grid points. Second, 10 sensors are placed to cover 25 surveillance spots randomly scattered in the field. We then evaluate the effectiveness of the resulted sensor placements using the real data traces via the approach discussed in Section 6. Fig. 1(a) and Fig. 1(b) show the CDF of the detection probability under the two sensor placements, respectively. From the figures, we can see that over 90% surveillance spots are covered in both two placements, which satisfies the required lower bound of detection probability. In Section 6, we have discussed the reasons for the remaining 10% surveillance spots that do not reach the requirement on detection probability.

Appendix F
More Numerical Evaluation Results

In Section 7, we have evaluated the impact of surveillance spot clustering on our algorithms and compared the divide-and-conquer placement algorithm with the global optimal algorithm as well as a greedy algorithm. In this appendix, we present more evaluation results on the impacts of fusion radius, impact region radius and decay factor. The evaluation settings can be found in Section 7.

F.1 Fusion Radius and Impact Radius

We first evaluate the impact of fusion radius. Total 45 surveillance spots scatter randomly in the field. The cluster-based divide-and-conquer placement algorithm is used. We change the fusion radius R from 3 to 12. Fig. F.1 plots the number of sensors required in the

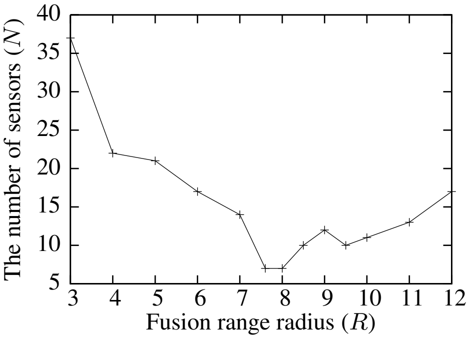


Fig. F.1. The number of sensors vs. fusion region radius.

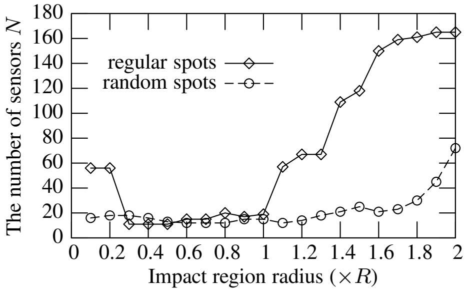


Fig. F.2. The number of sensors vs. impact region radius.

solution computed by the placement algorithm versus the fusion radius. From the figure, we can see that the number of required sensors drops rapidly from 37 to 7 when fusion radius R increases from 3 to 7.6, and gradually increases to 17 when R becomes larger. In Appendix D.1, we discussed the optimal *upper bound* of fusion radius that minimizes the network density. The simulation result in this appendix is consistent with the numerical results shown in Fig. D.1 in Appendix D.1, which means that the analysis in Appendix D.1 is tight enough to be applicable to computing the optimal fusion radius.

We now evaluate the impact of impact region radius. As discussed in Appendix D.2, if $W(R) \geq \sigma^2 \cdot g(1|\alpha, \beta)$, it is difficult to derive the optimal impact region radius that minimizes the network density. In this appendix, we conduct numerical experiments to explore the optimal impact region radius. Note that we use the sensor placement algorithm without surveillance spot clustering. In the experiments, 255 or 196 surveillance spots are chosen regularly or randomly in the $30 \times 30\,\mathrm{m}^2$ area, respectively. Fig. F.2 plots the number of placed sensors versus the impact region radius. We can see from the figure that, for both the cases of regular and random surveillance spots, the number of placed sensors has a convex relationship with the impact region radius. The intuition behind this result is as follows. If the impact region radius is too small (*e.g.*, less than $0.2R$), the overlapping fusion region of two adjacent surveillance spots may not be exploited to place shared sensors. If the impact region radius is close to $2R$, many shared sensors may be placed into the small area of the shared fusion region of two surveillance spots that are nearly $2R$ meters apart from each other. Both these two cases can lead to the inefficiency of sensor placement. Moreover, we can see from Fig. F.2 that fewer sensors are needed to cover random spots. Intuitively, if spots are randomly distributed, multiple spots can be in the same impact region such that a few sensors can cover them. In contrast, the regular spots have evener spatial distribution than the random spots, resulting more sensors placed.

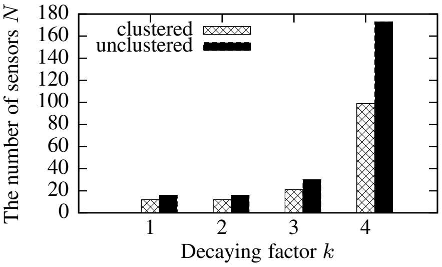


Fig. F.3. The number of sensors vs. decaying factor.

F.2 Impact of Decaying Factor

In this section, we evaluate the impact of the decaying factor k of the signal decay model on our algorithm. In the experiment, 45 surveillance spots are randomly chosen in the $30 \times 30\,\mathrm{m}^2$ field. Fig. F.3 plots the number of placed sensors versus the decaying factor k with and without the QT clustering, respectively. We can see from the figure that the number of placed sensors increases with the decaying factor. This result is consistent with the intuition that more sensors are required to cover the surveillance spots if the sensors receive weak signals from the target due to the fast attenuation of signal energy. Moreover, we can see from the figure that the cluster-based placement algorithm becomes more effective in reducing sensors when the decaying factor is larger.

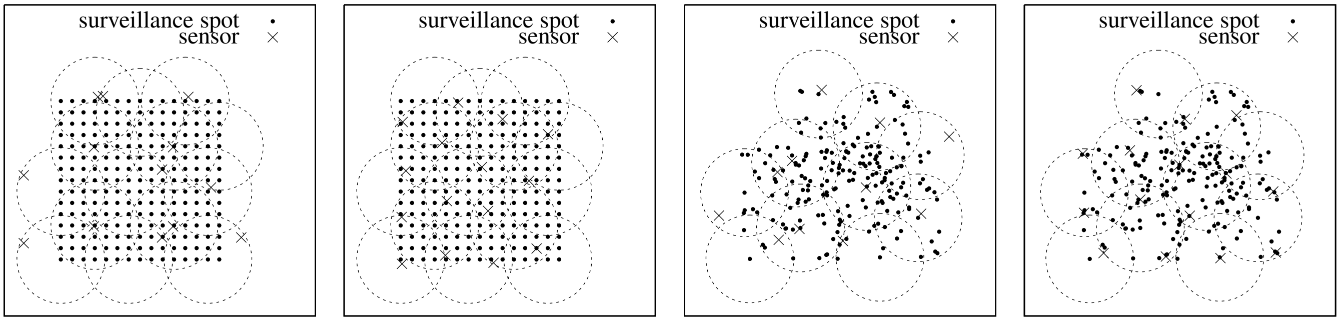
F.3 Visualization of Sensor Placements

In the first set of experiments, total $15 \times 15 = 225$ surveillance spots are regularly distributed, as shown in Fig. 4(a) and Fig. 4(b). The sensor placements computed by the cluster-based divide-and-conquer algorithm and the greedy algorithm presented in Section 7.2 are also shown in the two figures. Total 13 and 15 sensors are placed by the two algorithms, respectively. In the second set of experiments, total 196 surveillance spots randomly scatter in the field, as shown in Fig. 4(c) and Fig. 4(d). Total 11 and 15 sensors are placed, respectively.

References

[1] R. T. Rockafellar, "Lagrange multipliers and optimality," *SIAM review*, vol. 35, no. 2, 1993.

[2] B. W. Wah, Y. Chen, and T. Wang, "Simulated annealing with asymptotic convergence for nonlinear discrete constrained global optimization," *J. Global Optimization*, vol. 39, 2007.



(a) D&C: $K = 225$, $N = 13$

(b) Greedy: $K = 225$, $N = 15$

(c) D&C: $K = 196$, $N = 11$

(d) Greedy: $K = 196$, $N = 15$

Fig. F.4. Sensor placements using the cluster-based divide-and-conquer and the greedy algorithms. The dotted circles are the impact regions of clusters.