Sensor Deployment and Target Localization Based on Virtual Forces

Yi Zou , Krishnendu Chakrabarty

IEEE INFOCOM 2003

Outline

- Introduction
- Virtual Force Algorithm
- Target Localization
- Simulation Result
- Conclusion

Introduction(1/2)

- Distributed sensor networks (DSNs)
  - Are important for a number of strategic applications.
    - Ex. Coordinated target detection, surveillance ,and localization
  - The effectiveness of the DSNs is determined to a large extent by the coverage provided by the sensor deployment.

Introduction(2/2)

- Random deployment
  - Is desirable, if no a priori knowledge of the terrain are available.
  - overly clustered?
- How to improve the coverage?

Virtual Force Algorithm(1/10)

- Main steps
  - 1. random deployment of sensors
  - 2. cluster head execute the VFA to decides the new position of each sensor
  - One-time movement is carried out to deploy the sensors at these positions

Virtual Force Algorithm(2/10)

- Virtual force
  - Attractive forces
    - The two sensor nodes are too far apart from each other
    - Preferential coverage
  - Repulsive forces
    - The two sensor nodes are placed too close to each other
    - Obstacles

Virtual Force Algorithm(2/9)

$$\vec{F}_i = \sum_{j=1, j \neq i}^{k} \vec{F}_{ij} + \vec{F}_{iR} + \vec{F}_{iA}$$

$\vec{F}_{iA}$ be the total (attractive) force on s_i due to preferential converage area

$\vec{F}_{iR}$ be the total (repulsive) force on s_i due to obstacles

$\vec{F}_{ij}$ be the force exerted on s_i by another sensor s_j

Virtual Force Algorithm(3/9)

$$\overrightarrow{F_{ij}} = \begin{cases} (w_A(d_{ij} - d_{th}), \alpha_{ij}) & \text{if } d_{ij} > d_{th} \\ 0, & \text{if } d_{ij} = d_{th} \\ (w_R \frac{1}{d_{ij}}, \alpha_{ij} + \pi), & \text{if otherwise} \end{cases}$$

d_{ij} is the Euclidean distance between sensor s_i and s_j,

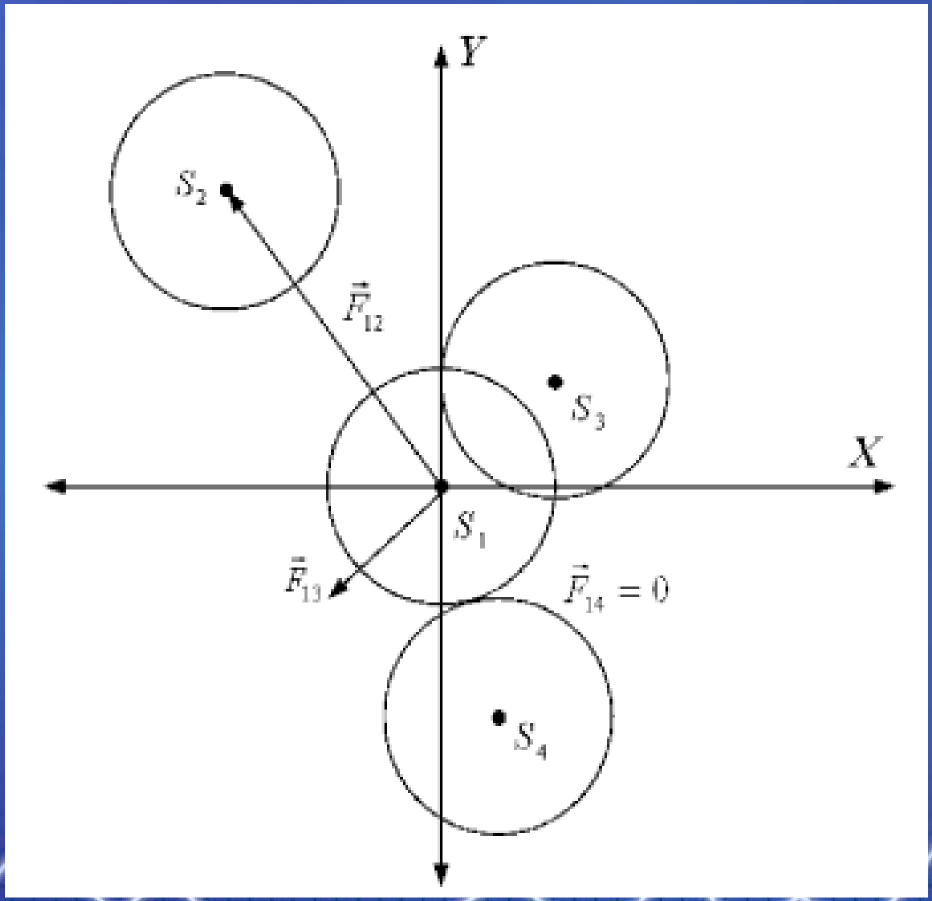
d_{th} is the threshold on the distance between s_i and s_j,

α_{ij} is the orientation (angle) of a line segment form s_i to s_j and

$w_A(w_R)$ is a measure of the attractive (repulsive) force

- The threshold distance d_{th} controls how close sensors get to each other.

Virtual Force Algorithm(4/9)



Virtual Force Algorithm(5/9)

- Binary sensor model
  - the coverage $c_{xy}(s_i)$ of an grid point P by sensor s_i

$$c_{xy}(s_i) = \begin{cases} 1, & \text{if } d(s_i, P) < r \\ 0, & \text{otherwise.} \end{cases}$$

Virtual Force Algorithm(6/9)

- Probabilistic sensor model

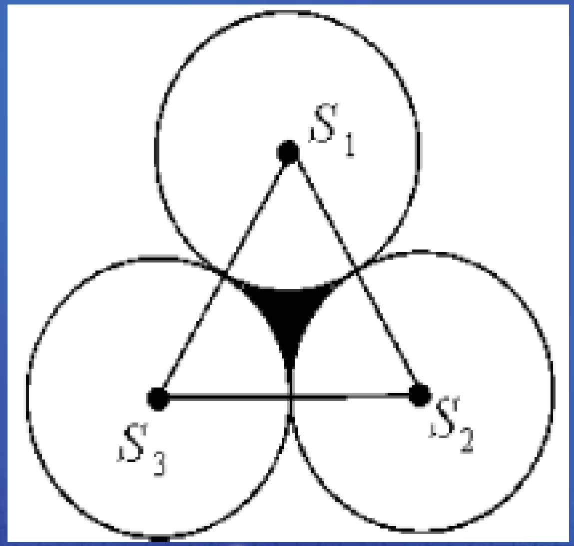
$$c_{xy}(s_i)=\begin{cases}0, & \text{if } r+r_e \le d(s_i,P) \\ e^{-\alpha a^{\beta}}, & \text{if } r-r_e < d(s_i,P) < r+r_e \\ 1, & \text{if } r-r_e \ge d(s_i,P)\end{cases}$$

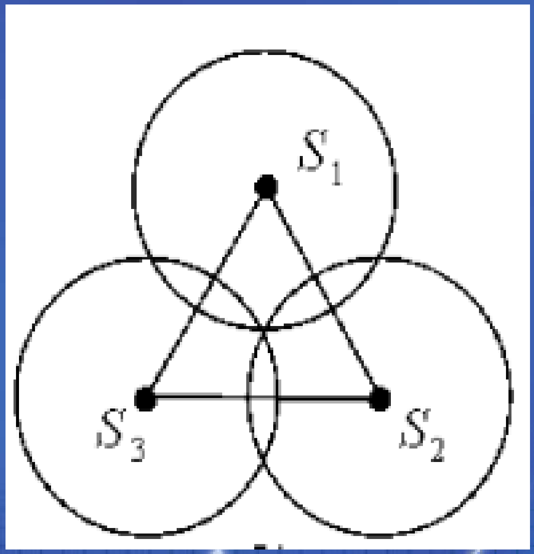
$r_e (r_e < r)$ is a measure of the uncertainty in sensor detection

$a = d(s_i,P)-(r-r_e)$

Virtual Force Algorithm(7/9)

- If $r_e \approx 0$ use the binary model, we attempt to make d_{ij} as close to 2r as possible
- Drawback





Virtual Force Algorithm(8/9)

- If $r_e > 0$,use the probabilistic sensor model

$$c_{x,y}(s_i, s_j) = 1 - (1 - c_{x,y}(s_i))(1 - c_{x,y}(s_j))$$

$$c_{x,y}(S_{ov}) = 1 - \prod_{s_i \in S_{ov}} (1 - c_{x,y}(s_i))$$

$$\min_{x,y}\{c_{x,y}(S_{ov})\} \geq c_{th}$$

Procedure *Virtual_Force_Algorithm* (Grid, $\{s_1, s_2, \cdots, s_k\}$)

Set $loops = 0$;
Set $MaxLoops$ =**MAX_LOOPS**;
While $(loops < MaxLoops)$
  /* coverage evaluation */
  For $P(x, y)$ in Grid, $x \in [1, width], y \in [1, height]$
    For $s_i \in \{s_1, s_2, \cdots, s_k\}$
      Calculate $c_{xy}(s_i, P)$ from the sensor model
      using $(d(s_i, P), c_{th}, d_{th}, \alpha, \beta)$;
    End
    If coverage requirements are met
      Break from **While** loop;
    End
  End
  /* virtual forces among sensors */
  For $s_i \in \{s_1, s_2, \cdots, s_k\}$
    Calculate $\vec{F}_{ij}$ using $d(s_i, s_j)$, d_{th}, w_A, w_R;
    Calculate $\vec{F}_{iA}$ using $d(s_i, PA_1, \cdots, PA_{n_P})$, d_{th};
    Calculate $\vec{F}_{iR}$ using $d(s_i, OA_1, \cdots, OA_{n_O})$, d_{th};
    $\vec{F}_i = \sum \vec{F}_{ij} + \vec{F}_{iR} + \vec{F}_{iA}, j \in [1, k], j \neq i$;
  End
  /* move sensors virtually */
  For $s_i \in \{s_1, s_2, \cdots, s_k\}$
    $\vec{F}_i(s_i)$ virtually moves s_i to its next position;
  End
  Set $loops = loops + 1$;
End

Target Localization(1/6)

- Two-step communication protocol
  - yes/no message
  - Detailed information
    ex. Detection strength, imagery
- Advantages
  - Save power and bandwidth

Target Localization(2/6)

- Detection probability table
  - Contains entries for all possible detection reports from those sensors that can detect a target at this gird point.

EXAMPLE PROBABILITY TABLE.

i	$d_1d_2d_3$	$p\_table_{xy}(i)$	i	$d_1d_2d_3$	$p\_table_{xy}(i)$
0	000	0	1	001	0
2	010	0.18	3	011	0.24
4	100	0	5	101	0
6	110	0.24	7	111	0.33

Target Localization(3/6)

- Score-based Ranking

$$SCORE_{xy}(t) = p\_table_{xy}(i(t)) \times w_{xy}(t)$$

$$w_{xy}(t) = \frac{k_{rep,xy}(t)}{k_{rep}(t)}$$

- $k_{rep}(t)$ is the set of sensors that have reported the detection of an object
- $k_{rep,xy}(t)$ is the set of sensors that can detect point P(x,y) and have also reported the detection of an object

Target Localization(4/6)

- Selection of sensors to query

$$S_q(t) : d(S_q(t), P_{MS}) = \min\{d(s_i, P_{MS})\}$$

- The sensors selected corresponding to he ones that have the shortest distance to those grid points with the highest scores

Target Localization(5/6)

- Evaluation of energy savings

$$
\begin{aligned}
E_1^*(t) &= k_{rep}(t)(E_t + E_r)T_1 \\
E_2^*(t) &= (k_q(t)E_r + E_t)T_2 \\
E_3^*(t) &= k_q(t)(E_t + E_r)T_3 \\
E_4^*(t) &= E_s T_s \\
E(t)^* &= E_1(t)^* + E_2(t)^* + E_3(t)^* + E_4(t)^* \\
E^* &= \sum_{t=t_{start}}^{t_{end}} E(t)^*
\end{aligned}
$$

$$
\begin{aligned}
E_1^*(t) &= k_{rep}(t)(E_t + E_r)T_1 \\
E_2^*(t) &= (k_q(t)E_r + E_t)T_2 \\
E_3^*(t) &= k_q(t)(E_t + E_r)T_3 \\
E_4^*(t) &= E_s T_s \\
E(t)^* &= E_1(t)^* + E_2(t)^* + E_3(t)^* + E_4(t)^* \\
E^* &= \sum_{t=t_{start}}^{t_{end}} E(t)^*
\end{aligned}
$$

$$
\Delta E = E - E^* = C \sum_{t=t_{start}}^{t_{end}} (k_{rep}(t) - k_q(t))
$$

Procedure

Localization (Grid, $\{s_1, s_2, \cdots, s_k\}$, TargetTrace)

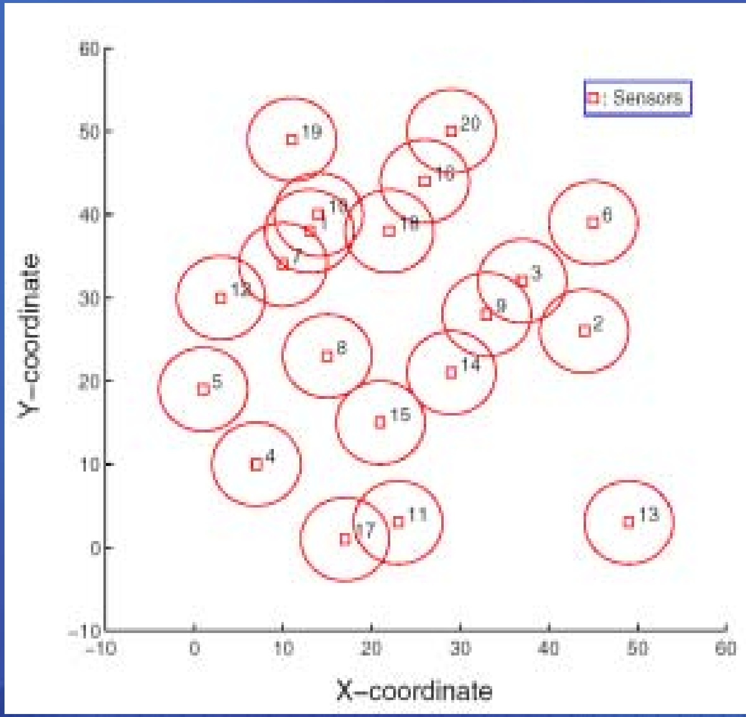
/* k_{max} is the maximum number of sensors that are allowed for querying. p_{rep} is the threshold level for a sensor to report to the cluster head of an event. $TargetTrace$ starts from t_{start} and it ends at t_{end}. The simulation time unit is 1. */

Set $t = t_{start}$;
While $(t \leq t_{end})$
    /* current target location */
    Set $Target = TargetTrace(t)$;
    /* calculate the scores */
    Calculate $S_{rep}(t)$ from $\{s_1, s_2, \cdots, s_k\}, Target(t), p_{rep}$;
    Set $k_{rep}(t) = |S_{rep}(t)|$;
    For $P(x, y)$ in Grid, $x \in [1, width], y \in [1, height]$
        Calculate $S_{rep,xy}(t)$ from $S_{rep}(t)$ and $P(x, y)$;
        Calculate the index $i(t)$ of p_table_{xy}
            from $S_{rep}(t)$ and $S_{rep,xy}(t)$;
        Set $k_{rep,xy}(t) = |S_{rep,xy}(t)|$;
        Set $w_{xy}(t) = \frac{k_{rep,xy}(t)}{k_{xy}(t)}$;
        Set $SCORE_{xy}(t) = p\_table_{xy}(i(t)) \times w_{xy}(t)$;
    End
    /* select sensors for querying */
    Calculate $S_q(t)$ from $SCORE_{xy}(t)$ and k_{max},
        $x \in [1, width], y \in [1, height]$;
    /* next time instant */
    Set $t = t + 1$;
End

Fig. 13. Pseudocode of the localization algorithm.

Simulation Results(1/4)

- Binary sensor detection model



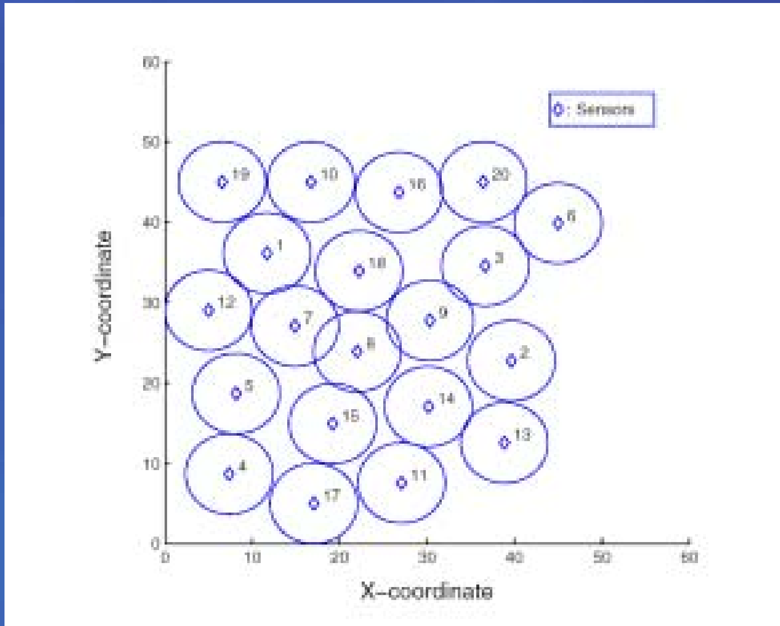
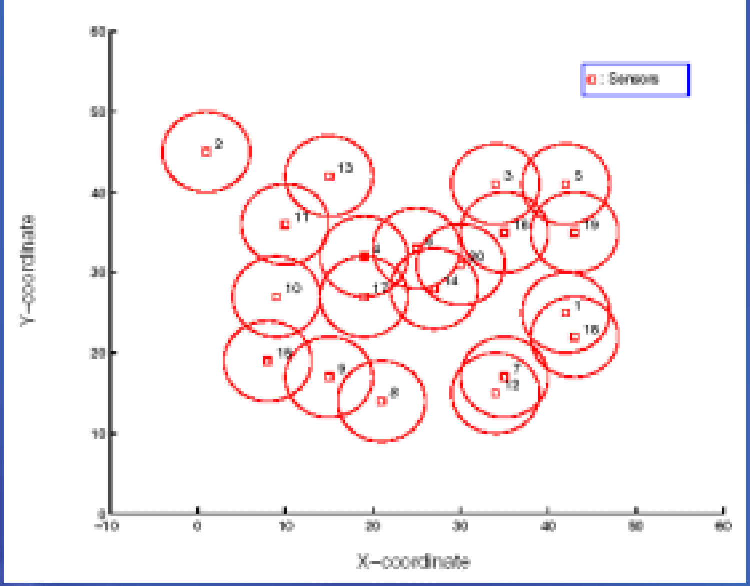


Fig. 15. Sensor positions after the execution of the VFA algorithm (binary sensor detection model).

Simulation Results(2/4)

- Probabilistic sensor detection model



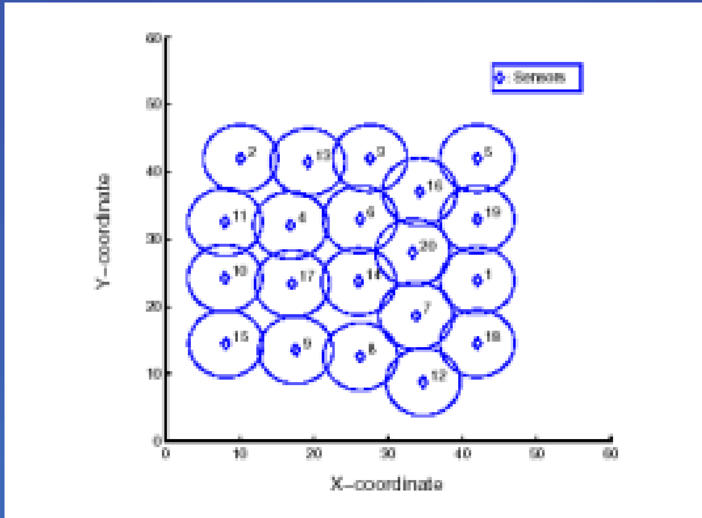


Fig. 18. Sensor positions after the execution of the VFA algorithm (probabilistic sensor detection model).

Simulation Results(3/4)

- Sensor Field with a preferential area and on obstacle

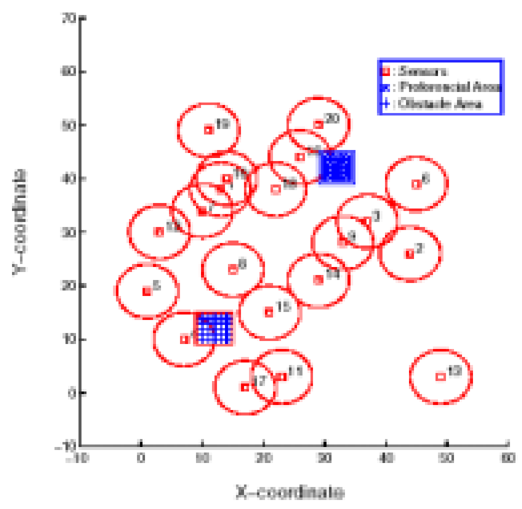


Fig. 20. Initial sensor positions after random placement with obstacles and preferred areas.

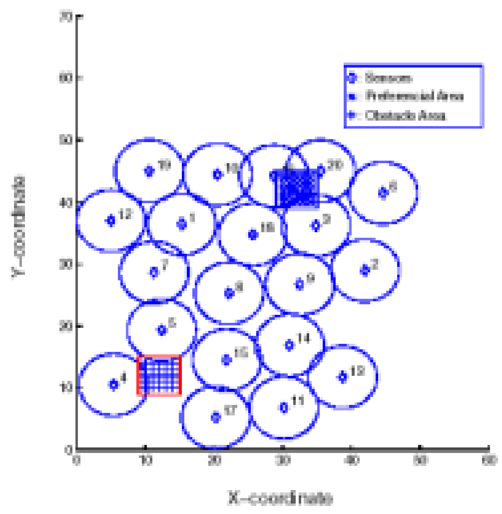


Fig. 21. Sensor positions after the execution of the VFA algorithm with obstacles and preferred areas.

Simulation Results(4/4)

- Probability-based target Localization

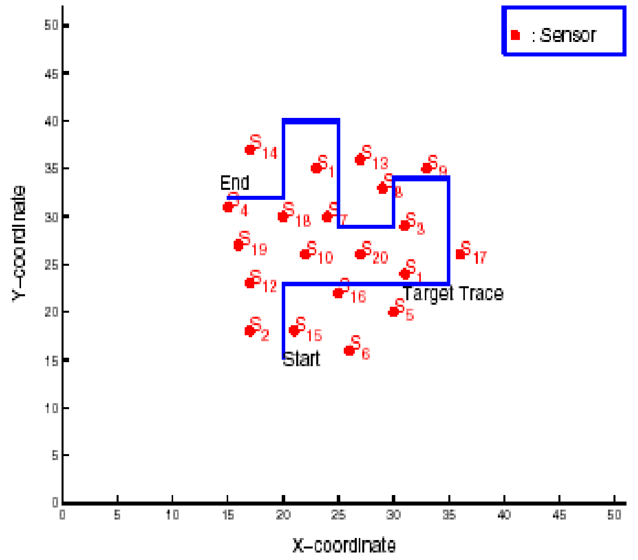


Fig. 23. Sensor field with sensors deployed by the VFA algorithm and target movement trace.

TABLE III

SENSORS SELECTED FOR QUERYING BY THE CLUSTER HEAD.

t	$S_{rep(t)}$	$S_q(t)$	$\Delta E(t)$
01	s_2, s_6, s_{15}	s_2, s_{15}	C
02	$s_2, s_6, s_{12}, s_{15}, s_{16}$	s_2, s_{12}	$3C$
03	$s_2, s_6, s_{12}, s_{15}, s_{16}$	s_2, s_{12}	$3C$
04	$s_2, s_6, s_{12}, s_{15}, s_{16}$	s_2, s_{12}	$3C$
...	...	...	...
41	s_3, s_8, s_9, s_{13}	s_9, s_3	$2C$
42	$s_3, s_7, s_8, s_9, s_{11}, s_{13}$	s_9, s_3	$4C$
43	$s_3, s_7, s_8, s_9, s_{11}, s_{13}, s_{20}$	s_9, s_3	$5C$
44	$s_3, s_7, s_8, s_9, s_{11}, s_{13}, s_{20}$	s_9, s_3	$5C$
...	...	...	...
79	$s_4, s_7, s_{10}, s_{11}, s_{14}, s_{18}, s_{19}$	s_7, s_{11}	$5C$
80	$s_4, s_7, s_{10}, s_{11}, s_{14}, s_{18}, s_{19}$	s_7, s_{11}	$5C$
81	$s_4, s_{11}, s_{14}, s_{18}, s_{19}$	s_{11}, s_{18}	$3C$
82	$s_4, s_{14}, s_{18}, s_{19}$	s_{18}, s_{14}	$2C$

Conclusion(1/2)

- Advantages
  - Negligible computation time and a one-time repositioning of the sensors
  - The desired sensor field coverage and model parameters can provided as input to the VFA algorithm
  - The localization algorithm can reduce the energy consumption for target detection and location

Conclusion(2/2)

- Future work
  - Make route plan for repositioning the sensors
  - Detect multiple objects
  - Consider about sensors nodes failures
  - Examine continuous coordination systems