J Syst Sci Complex (2018) 31: 276–286

Dynamically Optimized Sensor Deployment Based on Game Theory*

ZHANG Fan · ZHENG Zhi · JIAO Lei

DOI: 10.1007/s11424-018-7384-5
Received: 15 November 2017 / Revised: 9 January 2018

Abstract Sensor network deployment is the key for sensors to play an important performance. Based on game theory, first, the authors propose a multi-type sensor target allocation method for the autonomous deployment of sensors, considering exploration cost, target detection value, exploration ability and other factors. Then, aiming at the unfavorable environment, e.g., obstacles and enemy interference, the authors design a method to maintain the connectivity of sensor network, under the conditions of effective detection of the targets. Simulation result shows that the proposed deployment strategy can achieve the dynamic optimization deployment under complex conditions.

Keywords Connectivity, dynamic deployment, game theory, target allocation.

1 Introduction

Sensors are widely used in environmental monitoring, battlefield target detection, etc., and have aroused widespread concern. The deployment of sensors is an important foundation to ensure the effective operation of the sensor network, about which a serious of results have achieved. Ozturk, et al.[1] applied artificial bee colony algorithm (ABC) to dynamic deployment of wireless sensor networks (WSNs). Based on [1], authors in [2] used probability-based node-aware model modeling in hybrid WSNs. Liao, et al.[3] applied glowworm swarm optimization (GSO) to the optimal deployment of WSNs. In [4], authors proposed a network coverage algorithm by improving GSO. In [5], a mobile-sensor-network-node optimization strategy based on differential evolution (DE) was put forward. In [6], artificial immune system (AIS) was used into the redeployment of WSNs. In order to achieve the full coverage of all targets in the

ZHANG Fan · ZHENG Zhi (Corresponding author) · JIAO Lei
School of Automation, Beijing Institute of Technology, Beijing 100081*, China; State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing* 100081*, China.*
Email: yi_cui_zhang@163.com; zz2012@bit.edu.cn; jiaolei_1029@163.com.

*This research was supported by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant No. 61321002, the Program for Changjiang Scholars and Innovative Research Team in University under Grant No. IRT1208, the Changjiang Scholars Program, and the Beijing Outstanding Ph.D. Program Mentor under Grant No. 20131000704.
◇*This paper was recommended for publication by Guest Editor XIN Bin.*

network with directed sensors, authors proposed a target coverage scheme based on immune algorithm[7]. Aziz, et al.[8] tried to settle the coverage optimization problem of WSN by using particle swarm optimization (PSO) for the first time. Howard, et al.[9] introduced potential field-based approach to optimizing deployment of WSNs. In this method, each sensor node is treated as a mobile robot. Through the action of the Potential Field-based Approach, the sensor nodes that are initially gathered together are fully dispersed, thereby expending the coverage area in the network.

As can be seen from the studies above, the main feature of current sensor deployment researches is reflected in the ideal deployment environment, that is, the complex situations appear in the actual deployment missions have not been considered. The deployment environment of sensors is increasingly complex. Especially in combat missions, values vary from target to target, and targets pose different threats to sensors. Due to the limited function of a single sensor, a variety of types of sensors, with different capabilities, are needed to complete the collaboration. As different types of sensors have differences in value and capability, and have limited detection ranges, how to effectively accomplish target allocation among different types of sensors is an important part of ensuring the optimal deployment. In addition, most sensor networks have ideal deployment environment without considering the impact of actual environment on deployment. Some restrictions will affect the connectivity of the sensor network and cause a serious impact on deployment performance, e.g., threatened areas set by the enemy, un-deployable areas in the environment, and so on. Starting from the actual application needs, in this paper, we first establish an objective function, including the value of the detection target, detection cost, detection revenue and other indicators, and allocate targets in sensor network. Then, treating the threat areas set by the enemy and the un-deployable areas as obstacles, design a method to make it possible to guarantee the connectivity of the network and avoid these two types of areas mentioned above while sensors probe targets assigned. This paper uses game theory to obtain the optimal decision and finally, shows the effectiveness of the sensor network deployment strategy and its suitability for dynamic deployment in complex environment through simulation.

2 Game Theory Model

Game theory studies that, under certain rules, parties participating in the game, in accordance with given sequence, predict the behavior of other individuals according to the information they have obtained, and select the optimal strategy that maximizes their own revenue from the strategy set, so that to make the whole situation achieve a balance. The game contains four basic elements: Participating individuals, strategy set, the order of the game and the proceeds of the game.

2.1 Participating Individuals

It is usually assumed that the participators, in a finite number, are rational, that is, each individual will choose the strategy that maximizes its own profits, regardless of considering whether the strategy will damage the interests of others. In this paper, participating individuals are sensors in the sensor network $S = \{s_1, s_2, \cdots, s_n\}$.

2.2 Strategy Set

Each individual has its own strategy set $a_i = \{a_{i1}, a_{i2}, \cdots, a_{in}\}$, and the number of strategy sets can be finite or infinite. During each game, each individual in accordance with the specified order, predicts the strategies of other individuals, and chooses one from their own strategy set to maximize his own profits. The set of all individual strategies $A = \prod_{i=1}^{n} a_i$ is called strategy space. For easy description, take the strategy combination as $a = (a_i, a_{-i})$, among which, a_{-i} represents a decision collection including all individuals expect individual s_i.

In this paper, the next available location of each individual constitutes sensor decision collection. To reduce the amount of computation, with the sensor as the center, select discrete coordinate points within a certain range of R as the current optional strategy, according to the actual physical characteristics of the sensor. And divide R into several parts $R = \{R_1, R_2, \cdots, R_n\}$, based on the number of neighbors, to balance the need of the network connectivity and decentralized deployment more effectively. The more the neighbors, the father discrete space the sensor chooses, on the contrary, the more recent one it chooses, which is shown in Figure 1.

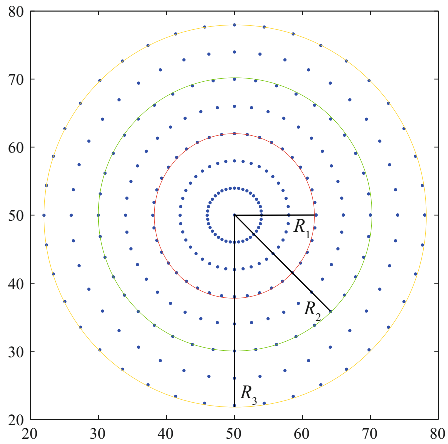


Figure 1 Discretization of the sensor's next decision space

2.3 Revenue Function

Define $f_i = (a_i, a_{-i})$ as a revenue function to describe the profits of individual s_i under the strategy (a_i, a_{-i}). It can be seen that the benefits of an individual is not only related to the strategy of its own choice, but also depends on the strategy of other participants.

2.4 Nash Equilibrium

For individual s_i, if there exists a_i^*, and only if $\forall a_i' \in a_i, f_i(a_i^*, a_{-i}) \geq f_i(a_i', a_{-i})$, we can say a_i^* is the optimal strategy set for it. If $\forall s_i \in S, \forall a_i' \in a_i, f_i(a_i^*, a_{-i}^*) \geq f_i(a_i', a_{-i}^*)$, then the set of optimal strategies for all individuals is called Nash equilibrium. As can be seen, in the case of other sensors are not changed, it is impossible for each to obtain greater benefits by changing its strategy alone.

3 Task Allocation Model

Assuming that the sensor network consists of n sensors $S = \{s_1, s_2, \cdots, s_n\}$, and the mission area has m important targets that require focused detection, recorded as $K = \{k_1, k_2, \cdots, k_m\}$.

3.1 Detection Cost

Suppose s_i detects the target k_j, and the probability of being found by the target is Pk_{ij}, and the value of is v_i^s. Then the cost of assigning target k_j to s_i can be written as

$$C_1 = v_i^{\mathrm{s}} Pk_{ij}. \tag{1}$$

The closer the distance between the sensor s_i and the target k_j, the less energy is consumed, which can be shown as follows:

$$C_2 = \frac{d_{i\_j}}{\max_{j \in K} d_{i\_j}}, \tag{2}$$

d_{i_j} is the distance between sensor s_i and the target k_j.

3.2 Detection Revenue

The value of the target k_j is v_j^k, and the detection efficiency of the sensor s_i to the target k_j is Ps_{ij}, Then the detection revenue of detection target k_j can be calculated as

$$C_3 = v_j^k Ps_{ij}. \tag{3}$$

Task assignment problem is a multi-objective optimization one. According to the indexes above, we can transform it into a single-objective optimization problem through the linear weighting method and design the revenue function of target allocation:

$$\min f = w_1 C_1 + w_2 C_2 - w_3 C_3, \tag{4}$$

$\omega_1, \omega_2, \omega_3$ are positive weight coefficients.

4 Connectivity Maintenance

After dividing task area, each sensor needs to identify and move to the next position. In [10], authors propose that the connectivity of sensor networks depends on the number of neighbors within the communication range of each sensor. If each sensor can maintain connectivity with L key neighbors during deployment, the network can remain connectivity with greater probability. According to the idea mentioned in [11], suppose that the sensors s_i and s_j are neighbors of each other at time t. Define the circle whose radius is $r_c/2$, half the communication radius, as D_{ij}, the center of which is the midpoint connected by s_i and s_j. If at the next time $t+1$, the positions p_i and p_j of s_i and s_j are still within the circle D_{ij}, that is, $p_i(t+1) \in D_{ij}$ and $p_j(t+1) \in D_{ij}$, then s_i and s_j will certainly be able to maintain connectivity. Thus, if $p_i(t+1) \in \cap_{j \in N_i} D_{ij}$, then s_i can continue to be connected with all of the neighbors at the next moment, which have already been connected with it at the current time t, as shown in Figure 2.

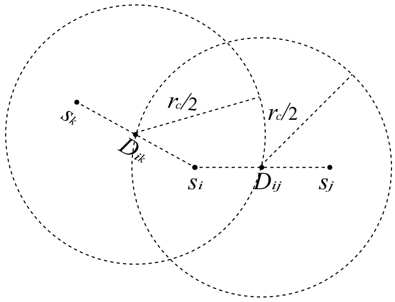


Figure 2 Maintain connectivity

5 Design of Revenue Function

To meet the needs of exploration mission, two types of performance should be reflected in the revenue function. One, sensors should be as close as possible to detection targets. The other, the distribution of the sensors should be as uniform as possible so that the efficiency of detection can be improved. Also, keep the appropriate distance between sensors, which can increase the overall coverage area of the network. In addition, avoid deploying the sensors at both threatened and un-deployable areas.

In this paper, we consider the threatened areas and un-deployable areas, set by the enemy, as obstacles. For ease of analysis, a circle O is used to represent the obstacle. When the sensor detects an obstacle, it will keep a certain distance from the edge point q of the nearest obstacle. To reduce the probability of bumping into the obstacles, we state that the individual benefits value will decrease as the distance between the sensor and the obstacle O decreases. When the distance is very close to 0, the penalty value tends to be infinite, and the revenue value will go to infinitesimal. The penalty function designed as $\sum_{o\in O_i(t)}\frac{1}{d_{io}(t+1)}$, and $d_{io}(t+1) = ||p_i(t+1)-O_0-r_0||$ represents the distance between the position (p_i) of sensor s_i and the edge point q of the obstacle nearest to itself, and O_0, r_0 represents the center and the radius of the obstacle, respectively.

In summary, we can design the revenue function of the deployment process:

$$\begin{aligned}\max f_i(a_i(t), a_{-i}(t)) = & -\mu_1 d_{i\_j}(t+1) + \mu_2 \sum_{j\in N_i(t)} d_{ij}(t+1) \\ & -\mu_3\left[\sum_{j\in N_i(t)}\left(d_{ij}(t+1) - \frac{\sum_{j\in N_i(t)} d_{ij}(t+1)}{N_i(t)}\right)^2\right]^{1/2} \\ & -\mu_4 \sum_{o\in O_i(t)} \frac{1}{d_{io}(t+1)}, \end{aligned} \tag{5}$$

where $\mu_1, \mu_2, \mu_3, \mu_4$ are positive weight coefficients, $N_i(t)$ consists of the neighbors of s_i at the current moment, d_{i_j} is the distance between s_i and its own detection target, d_{ij} is the distance between s_i and its neighbor s_j, Those four parts in the function (5) represent that when sensor s_i is expected to adopt strategy $a_i(t)$, the distance between s_i and its own target k and the distance from all current neighbors at the next moment, the variance of the distance distribution and the penalty value of the distance from detected obstacle O at the current time.

6 Experimental Simulation

We randomly distribute 25 sensors with communication radius of 30, and set 25 targets. The parameters of sensors and targets are as follows:

Table 1 The parameters of sensors and targets

(a) Target number, value and location

Target number	Value	Location
k_1	64	$[38.2, 65.5]$
k_2	16	$[51.9, 11.9]$
k_3	17	$[13.4, 44.3]$
k_4	82	$[29.8, 91.1]$
k_5	83	$[34.5, 65.9]$
k_6	88	$[66.7, 10.8]$
k_7	12	$[70.8, 50.1]$
k_8	65	$[17.4, 48.6]$
k_9	57	$[41.4, 90.3]$
k_{10}	86	$[30.8, 77.4]$
k_{11}	75	$[86.8, 87.1]$
k_{12}	4	$[29.1, 38.4]$
k_{13}	86	$[11.4, 45.4]$
k_{14}	47	$[79.3, 81.4]$
k_{15}	49	$[17.1, 92.4]$
k_{16}	52	$[22.8, 75.9]$
k_{17}	39	$[52.7, 71.6]$
k_{18}	39	$[76.4, 90.9]$
k_{19}	21	$[61.7, 35.8]$
k_{20}	11	$[86.1, 40.4]$
k_{21}	72	$[20.1, 40.6]$
k_{22}	3	$[47.1, 70.1]$
k_{23}	100	$[86.1, 19.3]$
k_{24}	8	$[84.1, 17.4]$
k_{25}	40	$[28.3, 10.6]$

(b) Sensor number, value and initial location

Sensor number	Value	Initial Location
s_1	44	$[8.9, 11.7]$
s_2	24	$[12.8, 10.1]$
s_3	67	$[10.5, 9.9]$
s_4	60	$[14.3, 13.9]$
s_5	76	$[10.8, 10.1]$
s_6	33	$[18.5, 6.6]$
s_7	74	$[11.4, 6.5]$
s_8	3	$[6.3, 6.0]$
s_9	11	$[13.2, 6.0]$
s_{10}	88	$[8.6, 12.5]$
s_{11}	33	$[13.6, 5.7]$
s_{12}	99	$[10.6, 8.4]$
s_{13}	60	$[6.7, 13.2]$
s_{14}	26	$[10.9, 13.1]$
s_{15}	36	$[8.4, 14.0]$
s_{16}	49	$[6.1, 10.9]$
s_{17}	28	$[11.4, 5.6]$
s_{18}	81	$[13.1, 14.5]$
s_{19}	27	$[12.5, 9.9]$
s_{20}	20	$[10.9, 8.7]$
s_{21}	75	$[6.4, 7.9]$
s_{22}	68	$[11.3, 9.4]$
s_{23}	71	$[13.6, 11.9]$
s_{24}	17	$[11.0, 8.9]$
s_{25}	39	$[9.7, 7.8]$

Table 2 Target treat probability to sensor

ID	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}
k_1	0.73	0.40	0.53	0.64	0.99	0.79	0.23	0.46	0.52	0.54	0.26	0.03	0.88
k_2	0.75	0.09	0.33	0.91	0.51	0.85	0.84	0.88	0.26	0.70	0.74	0.53	0.10
k_3	0.06	0.04	0.81	0.64	0.38	0.01	0.45	0.27	0.19	0.92	0.39	0.73	0.93
k_4	0.77	0.39	0.52	0.18	0.54	0.82	0.06	0.71	0.17	0.03	0.68	0.69	0.37
k_5	0.01	0.59	0.18	0.85	0.07	0.28	0.07	0.13	0.18	0.91	0.01	0.54	0.16
k_6	0.34	0.01	0.88	0.97	0.83	0.11	0.78	0.34	0.29	0.57	0.84	0.32	0.01
k_7	0.66	0.30	0.05	0.32	0.25	0.41	0.10	0.52	0.37	0.14	0.79	0.57	0.36
k_8	0.58	0.34	0.53	0.25	0.17	0.16	0.98	0.33	0.63	0.30	0.27	0.89	0.42
k_9	0.79	0.68	0.63	0.56	0.47	0.05	0.49	0.69	0.38	0.24	0.84	0.89	0.42
k_{10}	0.08	0.33	0.88	0.73	0.23	0.76	0.74	0.85	0.46	0.04	0.30	0.30	0.01
k_{11}	0.07	0.12	0.75	0.62	0.30	0.86	0.35	0.91	0.75	0.19	0.14	0.26	0.30
k_{12}	0.49	0.53	0.37	0.02	0.49	0.79	0.07	0.87	0.25	0.29	0.69	0.66	0.27
k_{13}	0.17	0.90	0.84	0.29	0.84	0.29	0.82	0.77	0.68	0.98	0.20	0.30	0.86
k_{14}	0.08	0.28	0.49	0.12	0.82	0.86	0.24	0.74	0.04	0.97	0.18	0.63	0.71
k_{15}	0.21	0.52	0.13	0.14	0.05	0.73	0.05	0.70	0.45	0.50	0.37	0.56	0.22
k_{16}	0.55	0.79	0.46	0.73	0.14	0.06	0.05	0.59	0.79	0.58	0.71	0.09	0.31
k_{17}	0.08	0.30	0.11	0.46	0.25	0.66	0.70	0.67	0.56	0.73	0.42	0.38	0.97
k_{18}	0.07	0.01	0.28	0.53	0.88	0.07	0.42	0.29	0.60	0.99	0.29	0.56	0.94
k_{19}	0.93	0.74	0.40	0.28	0.24	0.09	0.69	0.54	0.75	0.95	0.67	0.05	0.77
k_{20}	0.50	0.28	0.07	0.84	0.37	0.51	0.84	0.92	0.67	0.01	0.33	0.48	0.83
k_{21}	0.98	0.92	0.39	0.15	0.30	0.73	0.37	0.70	0.73	0.89	0.94	0.02	0.05
k_{22}	0.71	0.01	0.76	0.02	0.42	0.28	0.53	0.77	0.36	0.24	0.11	0.43	0.14
k_{23}	0.71	0.47	0.93	0.59	0.12	0.98	0.74	0.57	0.89	0.08	0.91	0.24	0.50
k_{24}	0.29	0.13	0.93	0.64	0.77	0.57	0.39	0.96	0.97	0.89	0.92	0.81	0.67
k_{25}	0.94	0.28	0.31	0.48	0.20	0.93	0.76	0.03	0.01	0.89	0.01	0.43	0.17

ID	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}	s_{20}	s_{21}	s_{22}	s_{23}	s_{24}	s_{25}
k_1	0.39	0.47	0.46	0.67	0.65	0.78	0.83	0.52	0.36	0.38	0.17	0.73
k_2	0.91	0.01	0.25	0.29	0.27	0.53	0.08	0.58	0.54	0.04	0.27	0.67
k_3	0.96	0.93	0.83	0.13	0.85	0.21	0.69	0.22	0.54	0.02	0.67	0.99
k_4	0.76	0.37	0.29	0.96	0.39	0.38	0.34	0.68	0.10	0.87	0.44	0.43
k_5	0.07	0.86	0.11	0.57	0.72	0.60	0.75	0.33	0.48	0.90	0.89	0.70
k_6	0.42	0.87	0.38	0.20	0.28	0.80	0.13	0.68	0.37	0.31	0.28	0.86
k_7	0.17	0.02	0.21	0.26	0.39	0.09	0.72	0.97	0.53	0.76	0.55	0.06
k_8	0.26	0.87	0.50	0.09	0.27	0.76	0.02	0.70	0.76	0.64	0.29	0.18

ID	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}	s_{20}	s_{21}	s_{22}	s_{23}	s_{24}	s_{25}
k_9	0.41	0.87	0.52	0.67	0.56	0.87	0.40	0.59	0.57	0.82	0.94	0.01
k_{10}	0.23	0.10	0.91	0.94	0.37	0.83	0.85	0.17	0.16	0.85	0.75	0.86
k_{11}	0.17	0.95	0.94	0.15	0.12	0.91	0.82	0.84	0.19	0.27	0.57	0.56
k_{12}	0.43	0.41	0.95	0.66	0.15	0.40	0.33	0.49	0.03	0.11	0.43	0.08
k_{13}	0.22	0.91	0.49	0.78	0.70	0.88	0.65	0.25	0.49	0.60	0.81	0.42
k_{14}	0.94	0.53	0.57	0.90	0.26	0.47	0.39	0.20	0.11	0.15	0.62	0.05
k_{15}	0.37	0.24	0.99	0.50	0.83	0.03	0.44	0.91	0.55	0.05	0.36	0.20
k_{16}	0.35	0.78	0.56	0.38	0.17	0.95	0.60	0.77	0.36	0.37	0.05	0.14
k_{17}	0.99	0.60	0.61	0.75	0.67	0.55	0.98	0.80	0.41	0.93	0.74	0.46
k_{18}	0.32	0.20	0.15	0.45	0.24	0.28	0.07	0.12	0.30	0.20	0.99	0.80
k_{19}	0.89	0.37	0.30	0.57	0.64	0.39	0.35	0.33	0.36	0.42	0.58	0.11
k_{20}	0.37	0.06	0.72	0.32	0.33	0.31	0.20	0.33	0.29	0.50	0.20	0.38
k_{21}	0.01	0.98	0.56	0.26	0.46	0.78	0.40	0.22	0.87	0.20	0.76	0.84
k_{22}	0.76	0.51	0.14	0.73	0.70	0.01	0.36	0.99	0.64	0.62	0.99	0.43
k_{23}	0.42	0.94	0.55	0.01	0.35	0.73	0.78	0.22	0.75	0.01	0.32	0.99
k_{24}	0.13	0.21	0.62	0.64	0.30	0.60	0.54	0.27	0.81	0.81	0.74	0.36
k_{25}	0.85	0.68	0.68	0.20	0.74	0.66	0.70	0.34	0.81	0.33	0.75	0.77

Table 3 Sensor detection efficiency on target

ID	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9	k_{10}	k_{11}	k_{12}	k_{13}
s_1	0.66	0.81	0.77	0.62	0.44	0.60	0.16	0.14	0.78	0.74	0.99	0.76	0.91
s_2	0.14	0.81	0.85	0.29	0.15	0.51	0.53	0.83	0.59	0.72	0.03	0.48	0.70
s_3	0.09	0.21	0.66	0.68	0.58	0.84	0.20	0.13	0.17	0.81	0.74	0.06	0.12
s_4	0.97	0.70	0.27	0.42	0.95	0.28	0.15	0.68	0.08	0.74	0.58	0.51	0.64
s_5	0.14	0.88	0.21	0.73	0.84	0.50	0.75	0.94	0.82	0.64	0.95	0.40	0.14
s_6	0.35	0.26	0.53	0.63	0.08	0.10	0.91	0.45	0.15	0.09	0.39	0.58	0.08
s_7	0.17	0.84	0.63	0.91	0.85	0.32	0.21	0.11	0.06	0.99	0.51	0.96	0.74
s_8	0.77	0.80	0.04	0.26	0.94	0.65	0.16	0.06	0.58	0.41	0.79	0.96	0.19
s_9	0.75	0.36	0.93	0.01	0.99	0.87	0.79	0.77	0.46	0.07	0.39	0.99	0.01
s_{10}	0.20	0.73	0.76	0.72	0.81	0.76	0.61	0.01	0.46	0.74	0.16	0.51	0.02
s_{11}	0.88	0.56	0.39	0.80	0.24	0.81	0.77	0.60	0.90	0.99	0.56	0.72	0.40
s_{12}	0.44	0.31	0.63	0.37	0.26	0.47	0.55	0.35	0.15	0.54	0.59	0.14	0.59
s_{13}	0.43	0.32	0.56	0.31	0.10	0.28	0.29	0.39	0.79	0.59	0.82	0.18	0.04
s_{14}	0.68	0.39	0.35	0.89	0.28	0.45	0.59	0.39	0.67	0.15	0.99	0.23	0.01
s_{15}	0.11	0.10	0.21	0.46	0.16	0.28	0.16	0.41	0.91	0.63	0.34	0.91	0.57
s_{16}	0.55	0.27	0.99	0.92	0.73	0.35	0.37	0.82	0.56	0.54	0.01	0.08	0.35

Table 3 (continued) Sensor detection efficiency on target

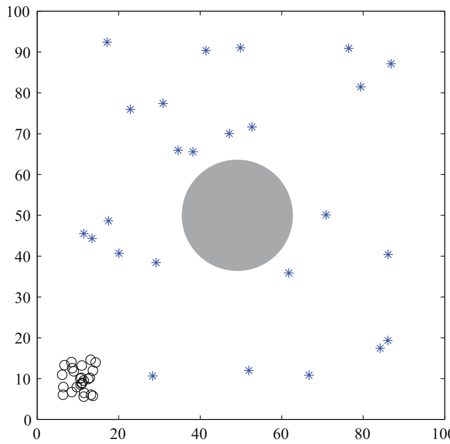
ID	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9	k_{10}	k_{11}	k_{12}	k_{13}
s_{17}	0.97	0.03	0.70	0.67	0.44	0.61	0.90	0.72	0.20	0.28	0.06	0.69	0.16
s_{18}	0.60	0.66	0.52	0.12	0.74	0.41	0.41	0.90	0.25	0.65	0.46	0.18	0.62
s_{19}	0.65	0.90	0.41	0.45	0.99	0.92	0.83	0.78	0.62	0.99	0.98	0.81	0.55
s_{20}	0.66	0.37	0.20	0.33	0.82	0.32	0.97	0.32	0.37	0.50	0.03	0.32	0.84
s_{21}	0.43	0.25	0.80	0.80	0.57	0.75	0.56	0.20	0.35	0.20	0.36	0.80	0.40
s_{22}	0.51	0.87	0.40	0.52	0.49	0.20	0.68	0.65	0.02	0.63	0.17	0.81	0.05
s_{23}	0.23	0.74	0.10	0.09	0.23	0.30	0.54	0.10	0.76	0.68	0.86	0.31	0.99
s_{24}	0.55	0.26	0.87	0.45	0.15	0.26	0.74	0.88	0.23	0.45	0.63	0.20	0.04
s_{25}	0.59	0.36	0.61	0.96	0.71	0.69	0.64	0.58	0.66	0.08	0.88	0.16	0.12

ID	k_{14}	k_{15}	k_{16}	k_{17}	k_{18}	k_{19}	k_{20}	k_{21}	k_{22}	k_{23}	k_{24}	k_{25}
s_1	0.45	0.68	0.01	0.73	0.90	0.99	0.25	0.62	0.98	0.46	0.41	0.20
s_2	0.80	0.20	0.36	0.84	0.41	0.72	0.76	0.51	0.43	0.94	0.54	0.25
s_3	0.40	0.88	0.68	0.51	0.78	0.28	0.52	0.84	0.17	0.33	0.96	0.01
s_4	0.75	0.61	0.40	0.34	0.50	0.59	0.10	0.80	0.01	0.88	0.07	0.33
s_5	0.33	0.03	0.23	0.18	0.26	0.74	0.68	0.49	0.27	0.03	0.04	0.74
s_6	0.34	0.84	0.95	0.08	0.37	0.45	0.66	0.86	0.44	0.71	0.27	0.28
s_7	0.61	0.13	0.24	0.46	0.47	0.04	0.90	0.49	0.04	0.85	0.61	0.01
s_8	0.94	0.23	0.03	0.09	0.71	0.95	0.48	0.96	0.05	0.28	0.39	0.03
s_9	0.17	0.37	0.26	0.58	0.57	0.48	0.46	0.07	0.04	0.62	0.70	0.69
s_{10}	0.11	0.20	0.80	0.84	0.46	0.08	0.66	0.37	0.65	0.98	0.38	0.27
s_{11}	0.45	0.66	0.79	0.04	0.25	0.05	0.93	0.59	0.03	0.50	0.12	0.09
s_{12}	0.97	0.12	0.12	0.38	0.64	0.63	0.32	0.21	0.86	0.81	0.35	0.38
s_{13}	0.79	0.83	0.12	0.52	0.03	0.30	0.02	0.67	0.19	0.68	0.83	0.63
s_{14}	0.15	0.70	0.06	0.64	0.06	0.74	0.77	0.66	0.23	0.03	0.71	0.34
s_{15}	0.79	0.29	0.83	0.10	0.73	0.56	0.08	0.75	0.09	0.11	0.56	0.63
s_{16}	0.71	0.75	0.58	0.75	0.02	0.71	0.53	0.78	0.91	0.98	0.27	0.65
s_{17}	0.30	0.83	0.33	0.39	0.64	0.64	0.79	0.61	0.16	0.25	0.58	0.91
s_{18}	0.14	0.27	0.75	0.03	0.88	0.30	0.54	0.80	0.35	0.07	0.38	0.71
s_{19}	0.99	0.52	0.67	0.06	0.89	0.54	0.80	0.29	0.08	0.01	0.17	0.01
s_{20}	0.77	0.01	0.27	0.01	0.29	0.79	0.31	0.26	0.59	0.49	0.02	0.68
s_{21}	0.51	0.74	0.61	0.57	0.81	0.67	0.39	0.06	0.38	0.21	0.91	0.23
s_{22}	0.37	0.96	0.47	0.76	0.56	0.97	0.47	0.72	0.47	0.80	0.84	0.87
s_{23}	0.26	0.97	0.20	0.08	0.93	0.66	0.26	0.44	0.98	0.56	0.87	0.62
s_{24}	0.59	0.66	0.58	0.06	0.79	0.24	0.77	0.12	0.16	0.32	0.75	0.05
s_{25}	0.10	0.41	0.85	0.64	0.27	0.45	0.49	0.70	0.94	0.51	0.61	0.69

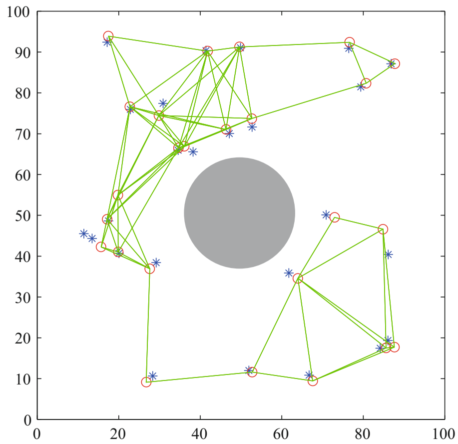
Table 4 Target assignment result

s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}
k_{12}	k_{13}	k_{21}	k_2	k_8	k_{16}	k_{25}	k_{19}	k_3	k_{10}	k_1	k_{22}	k_{14}
s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}	s_{20}	s_{21}	s_{22}	s_{23}	s_{24}	s_{25}	
k_4	k_5	k_6	k_7	k_{20}	k_{15}	k_{24}	k_{17}	k_9	k_{23}	k_{18}	k_{11}	

Set a circular obstacle area located at point (50, 50), shown as a gray circle, and 25 targets in star mode. Let $\omega_1 = 1, \omega_2 = 1, \omega_3 = 1, \mu_1 = 100, \mu_2 = 1, \mu_3 = 1, \mu_4 = 2000$, and $r_c = 30$. And let the number of key neighbors $L = 2$. In Figure 3, the black circles represent the initial position of the sensors, the red circles represent the final position of the sensors, the blue asterisks represent the location of the targets, the shadows represents the undeployable areas and the green lines represent the connectivity between sensors. The initial position of the sensors, the location of the targets and the undeployable area are shown in Figure 3(a). First, through target assignment, each target is assigned to the appropriate sensor, as shown in Table 4. It can be seen from the experimental results in Figure 3(b) that after 70 iterations, the sensors in the network can effectively avoid the obstacle areas and keep the network connected. At the same time, sensors can be close to the target for effective deployment.



(a) The initial position of the sensors, the location of the targets and the undeployable area



(b) The final location of the sensor and the network topology

Figure 3 The initial and final deployment of the sensors

7 Conclusion

According to the actual engineering requirements, we design a dynamic deployment strategy to allocate the detection target in sensor network. The objective function includes the detection value of the target, detection cost, detection revenue and other indicators. Considering the threatened areas and un-deployed areas set by enemy as obstacles, we propose a method of maintaining connectivity so that sensors can avoid those obstacles while detecting the assigned targets. In this paper, game theory is used to obtain the optimal decision-making. It is verified that the deployment strategy proposed above is effective, and it is also suitable for dynamic optimization deployment in complex environment.

References

[1] Ozturk C, Karaboga D, and Gorkemli B, Artificial bee colony algorithm for dynamic deployment of wireless sensor networks, *Turkish Journal of Electrical Engineering* & *Computer Sciencesv*, 2012, **20**(2): 255–262.

[2] Ozturk C, Karaboga D, and Gorkemli B, Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm, *Sensors*, 2011, **11**(6): 6056–6065.

[3] Liao W H, Kao Y, and Li Y S, A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks, *Expert Systems with Applications*, 2011, **38**(10): 12180–12188.

[4] Liu Z Z, Wang F B, and Zhang K W, WSN coverage optimization analysis based on improved firefly optimization algorithm, *Journal of Sensing Technology*, 2013, **26**(5): 675–682.

[5] Jin L Z, Chang G R, and Jia J, Distribution optimization of mobile sensor network nodes based on differential evolution algorithm, *Control and Decision*, 2010, **25**(12): 1857–1860.

[6] Kuang L and Cai Z, Immune system based redeployment scheme for wireless sensor networks, *Proceedings of the IET International Conference on Wireless Sensor Network*, Beijing, 2010, 69–72.

[7] Bi X J, Dong C, and Wang P Y, The research on target coverage of sensor network based on immune algorithm, *Computer Engineering*, 2011, **37**(24): 83–85.

[8] Aziz N, Mohemmed A W, and Sagar B S D, Particle swarm optimization and Voronoi diagram for wireless sensor networks coverage optimization, *Proceedings of Intelligent and Advanced Systems*, Kuala Lumpur, 2007, 961–965.

[9] Howard A, Matarić M J, and Sukhatme G S, Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem, *Distributed Autonomous Robotic Systems*, Springer, Japan, 2002, 299–308.

[10] Poduri S and Sukhatme G, Constrained coverage for mobile sensor networks, *IEEE International Conference on Robotics and Automation*, 2004, **1**(1): 165–171.

[11] Kim Y and Mesbahi M, On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian, *IEEE Transactions on Automatic Control*, 2006, **51**(1): 116–120.