*electronics*

Article

# A Multi-Mode Sensor Management Approach in the Missions of Target Detecting and Tracking

**Ce Pang [1], Ganlin Shan [1], Xiusheng Duan [2,*] and Gongguo Xu [1]**

[1] Department of Electronic and Optical Engineering, Shijiazhuang Campus, Army Engineering University, Shijiazhuang 050000, China; guanglongzhu@163.com (C.P.); 18392447996@163.com (G.S.); xgg2019@163.com (G.X.)
[2] School of Mechanical Engineering, Shijiazhuang Tiedao University, Shijiazhuang 05000, China
* Correspondence: sjzdxsh@163.com

Received: 11 December 2018; Accepted: 4 January 2019; Published: 8 January 2019

**Abstract:** In this paper, sensor management is divided into two processes: sensor deployment and sensor scheduling, after which a multi-mode sensor management approach based on risk theory is proposed. Firstly, the definition of risk is provided, on the basis of which the target detecting risk and the target tracking risk are separately presented, along with their computing methods. Secondly, when deploying sensors, the objective is to obtain the minimum target detecting risk. Similarly, when scheduling sensors, the objective is to obtain the minimal sum of target detecting risk and target tracking risk. Furthermore, to obtain sensor management schemes according to the objective functions, the improved bee colony algorithm based on double-probability and in combination with the particle swarm optimization algorithm is proposed. Finally, simulations are conducted, which indicate that the models and the algorithm in the paper possess some advantages over existing ones.

**Keywords:** sensor deployment; sensor scheduling; target detecting; target tracking; risk theory; artificial bee colony algorithm

## 1. Introduction

Sensor networks are applied in both military and civilian domains to acquire information. Especially in combat, they play an important role in target detecting and tracking. For many years, attempts have been made by researchers worldwide to make the best of sensor resources, called sensor management [1–6].

In sensor management, two steps should be taken. The first is to establish a sensor management model—in other words, an objective function obeying some rules should be introduced. The second is to get the optimal scheme from the objective function by traversing through all potential schemes or using optimization algorithms.

When modeling the objective functions, existing approaches could be roughly classified into three types, namely, the task-based approach [7–13], the information-based approach [14–20], and the risk-based approach [21–25]. Utilizing the first two approaches, some important technical indicators can be obtained, for instance, the target detecting probability and the target tracking precision, but actual operational requirements are not taken into consideration. For example, when tracking targets, if all domains in which a target might appear are covered by a radar beam, the target will not be missed, even though the target tracking precision is not so accurate. Compared with the former approaches, the third comes close to meeting the actual operational needs. With the operation risk considered, this approach can better utilize the sensor resource to get required technical indicators, which has become a research focus in recent years [26,27].

As for the category of optimization algorithms, there are two main kinds, namely the centralized algorithm [28–34] and the distributed algorithm [35–41]. Compared with distributed algorithms, schemes obtained by centralized algorithms are better, but have slower computing speeds. Furthermore, they can give more communication pressures on sensor networks.

After studying the literature above, the following problems still need to be addressed:

(1) Continuity and relevance are lacking in sensor management (including sensor deployment and sensor scheduling). For example, when conducting sensor scheduling, authors usually did not consider sensor deployment, and there are not suitable sensor networks deployed by other sensor deployment approaches to apply their sensor scheduling approaches. Obviously, sensor scheduling is based on sensor deployment and, accordingly, operates after it. In addition, when deploying sensor networks, the sensing radius of sensors must be taken into consideration. However, when scheduling sensors in reality, it is ignored.

(2) Different kinds of combat missions (including target detecting and target tracking) are analyzed independently. In a combat, with new targets appearing and acquired targets disappearing, different kinds of missions emerge at the same time, which means that a sensor network must detect targets and track targets simultaneously. Obviously, at this point, the sensor management model is totally different from those only considering one type of combat mission. Unfortunately, multi-mission sensor management approaches are seldom reported.

To solve the problems mentioned above, the sensor management approach (supposing that the sensor is a radar and the target is an aircraft) combining target detecting and target tracking is studied based on risk theory. Risk is defined as the product of the probability of a harmful event and the loss that the event brings [21]. The rest of this paper is organized as follows. In Section 2, the problem analysis and some assumptions are provided. In Section 3, two self-defined concepts, namely the target detecting risk and the target scheduling risk, are put forward, and their computing methods are provided. Furthermore, the sensor deployment model is established based on the target detecting risk and the sensor scheduling model is established based on the target tracking risk. In Section 4, to get schemes from the models, an improved artificial bee colony algorithm is introduced. Simulations and experiments are conducted to verify the proposed model and algorithm in Section 5, and the paper is concluded in Section 6.

## 2. Problem Analysis and Model Hypothesis

### 2.1. Combat Situation

Our monitoring domain is a 200 km $\times$ 200 km square $\Omega_0$, and the monitoring center is the center of the square in Figure 1.

As Figure 1 shows, for the process of target detecting and warning, a sensor network should first be deployed, and then, when a target appears, sensor scheduling can operate based on the deployed sensor network, realizing target detecting and tracking. As mentioned above, sensor deployment and sensor scheduling are closely related and inseparable. When tracking a target, the sensor network must keep detecting new targets simultaneously; in other words, target detecting and target tracking should be running at the same time. Of particular note is that the sensing radius can influence the tracking performance of the sensor, indicating that there are different variances of measurement noise due to different distances from targets to the sensor.

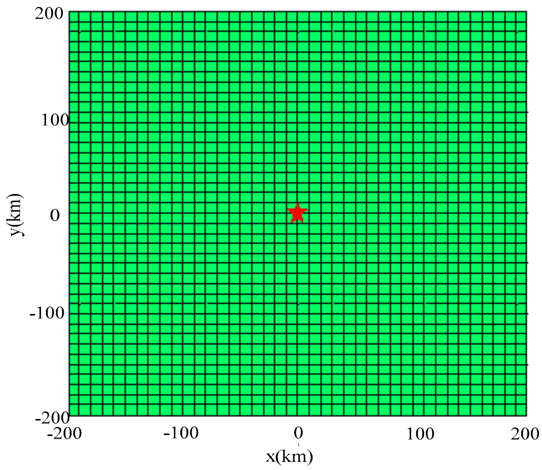



**Figure 1.** The monitoring domain.

### 2.2. Model Hypothesis

There are some assumptions as follows.

**Assumption 1.** *A sensor can work in two modes, namely 'target detecting' and 'target tracking', but the sensor can operate in only one working mode at a certain moment.*

**Assumption 2.** *In different measuring periods, the radiating power and radiating time of a sensor remain unchanged.*

**Assumption 3.** *During the target detecting period, the time interval is ignored, meaning that the detected domain is observed by the sensor continuously.*

**Assumption 4.** *Divide the monitoring domain $\Omega_0$ into small unit squares, and note that there is only one target at most in a unit square at a certain moment.*

## 3. Sensor Management Models

### 3.1. Target Detecting Model and Sensor Deployment Model

#### 3.1.1. Calculation of Target Detecting Probability

To calculate the detecting probability $p_d$, North [42] gave a fully precise solution by the following:

$$p_d = 0.5\text{erfc}(\sqrt{-\ln p_f} - \sqrt{SNR + 0.5}) \tag{1}$$

$$\text{erfc}(z) = 1 - \frac{2}{\sqrt{\pi}} \int_0^z e^{-v^2} dv \tag{2}$$

where $p_f$ is the given false alarm probability, $p_f = 0.01$ in this paper, and $SNR$ is the signal to noise ratio (SNR).

According to the radar equation [43], $SNR$ can be calculated as follows:

$$SNR = \frac{P\tau\sigma\lambda^2 G_r G_t}{(4\pi)^3 k T_0 F_n C_B r^4} \tag{3}$$

where $P$ is the radiation power of radars, $\tau$ is the radiation time, $\sigma$ is the area of target reflection, $\lambda$ is the radar wavelength, $G_r$ is the transmitting antenna gain, $G_t$ is the receiving antenna gain, $k$ is the

Boltzmann constant, $T_0 = 273K$ is the normal room temperature, $F_n$ is the noise coefficient, $C_B$ is the bandwidth correction factor, and $r$ is the distance between the target and the sensor.

In this paper, parameters are constants, except for radar radiation time, radiation power, and the distance from sensors to targets; then, the computing method of $SNR$ can be simplified as follows [44]:

$$SNR = SNR_{cal}(\frac{P\tau}{P_{cal}\tau_{cal}})(\frac{r}{r_{cal}})^{-4} \tag{4}$$

where $\tau$ is the radiation time and $P$ is the radiation power. When a sensor is working in the target detecting mode, the variable $\tau$ is approximately equal to the beam dwell time at the scanned domain in a measuring period. When a sensor is in the target tracking mode, the variable $\tau$ is approximately equal to the beam dwell time at the tracked target in a measuring period. The variable $r$ is the distance from a radar and a target. The variable $SNR_{cal}$ is a calibrated value—the $SNR$ of the received signal on condition that a sensor radiates for $\tau_{cal}$ time with the radiation power $P_{cal}$ and the distance $r_{cal}$. In this paper, $SNR_{cal} = 50$ dB, $P_{cal} = 1$ kW, $\tau_{cal} = 10$ ms, and $r_{cal} = 20$ km.

### 3.1.2. Calculation of Monitoring Priority

In this paper, the monitoring priority of a certain point (a unit square) in $\Omega_0$ is defined as the monitoring need of this point, and is affected by two factors, namely the importance priority, marked as $\theta$, and the initial target emergence probability, marked as $p$. A larger importance priority and emergence probability will result in higher monitoring priority.

Note that the coordinate of a point $c$ in $\Omega_0$ is $(x, y)$, the coordinate of the center $o$ in $\Omega_0$ is $(x_0, y_0)$, and the distance from $c$ to $o$ is calculated as follows:

$$d = \sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{5}$$

Suppose that the importance priority of $o$ is 1, and the minimum safe distance from a target to $o$ is $\alpha$. The importance priority of $c$ yields to the following:

$$\theta = \begin{cases} 1 - \frac{d}{\alpha},\ 0 \le d \le \alpha \\ 0,\ d > \alpha \end{cases} \tag{6}$$

Provided that the probability of a target with its initial emergence at $c$ is $p$, targets move towards $o$ from far to near, and the appearing domain is the marginal domain of $\Omega_0$ at the beginning of a combat. Note that the margin length in $\Omega_0$ is $a_0$, and $p$ can be calculated as follows:

$$p = \begin{cases} \frac{1}{4a_0}, c \text{ is in the marginal domains} \\ 0, c \text{ is in other domains} \end{cases} \tag{7}$$

Taking both $\theta$ and $p$ into consideration, the monitoring priority of $c$ can be calculated as follows:

$$f^c = \omega_1\theta + \omega_2 p \tag{8}$$

where $\omega_1$ and $\omega_2$ are the weights of $\theta$ and $p$, respectively.

In this paper, $\omega_1 = 1$, $\omega_2 = 1000$, $a_0 = 400$, $\alpha = 300$, $o = (0, 0)$, and the width of the marginal region of $\Omega_0$ is 20 km. The distribution of monitoring priority in domain $\Omega_0$ is shown in Figure 2.

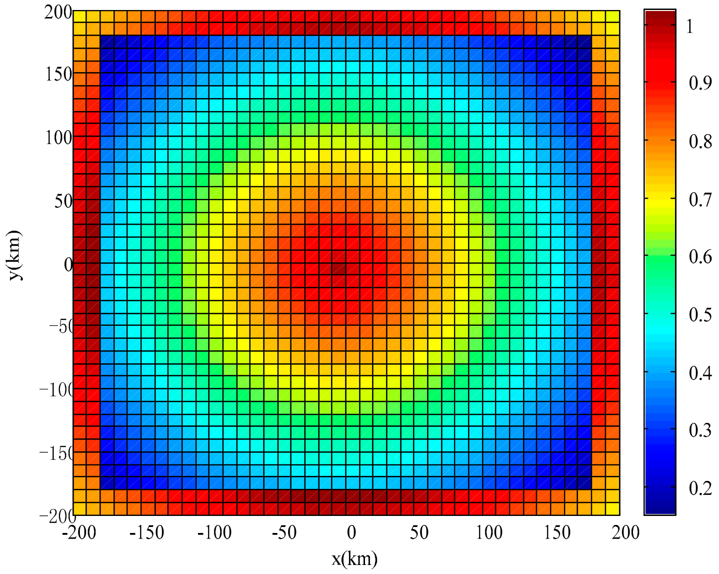



**Figure 2.** The distribution of the monitoring priority in domain $\Omega_0$.

#### 3.1.3. The Objective Function in Sensor Deployment

Note that the number of sensors applied to monitor the domain $\Omega_0$ is $m$. Assume that the sensing domain of each sensor is regarded as a circle with the sensing radius $R$, and its center is exactly the sensor. If the point $c$ is detected by $m_c$ sensors at the same time, note that the target detecting probabilities at $c$ by the sensors are $\left\{p_d^{(1)}, p_d^{(2)}, \ldots, p_d^{(m_c)}\right\}$; then the joint target detecting probability at $c$ is $p_d^{(*m_c)} = 1 - \prod_{i=1}^{m_c}(1 - p_d^{(i)})$, and the false dismissal probability at $c$ is $p_g^{(*c)} = \prod_{i=1}^{m_c}(1 - p_d^{(i)})$.

The target detecting risk at $c$ can be calculated according to the definition of risk as follows:

$$risk_1^c = p_g^{*c} \times f^c \tag{9}$$

When a sensor network is deployed, the average target detecting risk of $\Omega_0$ should be decreased to the minimum, then the optimal sensor deployment scheme can be calculated by the following:

$$\pi = \mathrm{argmin} E(risk_1) = \mathrm{argmin} \iint_{\Omega_0} risk_1^c / S^{\Omega_0} \tag{10}$$

where $E(\cdot)$ is the mean calculation, $E(risk_1)$ is the average target detecting risk, and $S^{\Omega_0}$ is the area of $\Omega_0$.

### *3.2. Target Tracking Model and Sensor Scheduling Model*

#### 3.2.1. Calculation of Target Missing Probability

Note that the motion state of target $t$ at the time instant $k$ is $\boldsymbol{X}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T$.

The state transition matrix is $\boldsymbol{F} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$, where $T$ is the sampling period and $T = 1s$.

At the time instant $k+1$, the motion state of target $t$ is $\boldsymbol{X}_{k+1} = \boldsymbol{F}\boldsymbol{X}_k + \boldsymbol{W}$, where $\boldsymbol{W}$ is the process evolution noise, following the Gaussian distribution; the mean value is 0; and the covariance matrix is

$\boldsymbol{Q} = \begin{bmatrix} T^4/4\sigma_x^2 & T^3/2 & 0 & 0 \\ T^3/2\sigma_x^2 & T^2\sigma_x^2 & 0 & 0 \\ 0 & 0 & T^4/4\sigma_y^2 & T^3/2\sigma_y^2 \\ 0 & 0 & T^3/2\sigma_y^2 & T^2\sigma_y^2 \end{bmatrix}$, where $\sigma_x$ and $\sigma_y$ are noise power spectral densities and $\sigma_x = \sigma_y = 1$.

At the time instant $k+1$, the measurement of sensor $s$ to target $t$ is $\boldsymbol{Y}_{k+1} = \boldsymbol{h}(\boldsymbol{X}_{k+1}) + \boldsymbol{V}_k$, where $\boldsymbol{V}_k$ is the measurement noise, following the Gaussian distribution; the mean value is 0; and the covariance matrix is $\boldsymbol{R}_k = diag((\sigma_k^r)^2, (\sigma_k^\alpha)^2)$.

In this paper, $\boldsymbol{h}(\boldsymbol{X}_{k+1}) = [d_{k+1}, \alpha_{k+1}]^T$, where $d_{k+1} = \sqrt{(x_{k+1} - x_0)^2 + (y_{k+1} - y_0)^2}$ is the distance from sensor $s$ to target $t$, $\alpha_{k+1} = \arctan\frac{x_{k+1} - x_0}{y_{k+1} - y_0}$ is the azimuth of target $t$, and $(x_0, y_0)$ is the coordinate of sensor $s$.

The variables $\sigma_k^r$ and $\sigma_k^\alpha$ are the function values of SNR, and they are given by the following [44]:

$$\sigma_k^r = \sigma_{cal}^r \sqrt{SNR_{cal}/SNR} \tag{11}$$

$$\sigma_k^\alpha = \sigma_{cal}^\alpha \sqrt{SNR_{cal}/SNR} \tag{12}$$

In this paper, when $SNR_{cal} = 50$ dB, $P_{cal} = 1$ kW, $\tau_{cal} = 10$ ms, $r_{cal} = 20$ km, and $SNR_{cal} = 50$ dB, $\sigma_{cal}^r = 200$ m and $\sigma_{cal}^\alpha = 0.1$ rad.

The extended Kalman filtering is used to estimate the motion state of targets as follows:

$$\begin{cases} \overline{\boldsymbol{X}}_{k+1|k} = \boldsymbol{F}\hat{\boldsymbol{X}}_{k|k} \\ \overline{\boldsymbol{P}}_{k+1|k} = \boldsymbol{F}\hat{\boldsymbol{P}}_{k|k}\boldsymbol{F}^T + \boldsymbol{Q}_k \\ \boldsymbol{S}_k = \boldsymbol{H}_k\overline{\boldsymbol{P}}_{k+1|k}\boldsymbol{H}_k^T + \boldsymbol{R}_k \\ \boldsymbol{K}_k = \overline{\boldsymbol{P}}_{k+1|k}\boldsymbol{H}_k^T\boldsymbol{S}_k^{-1} \\ \hat{\boldsymbol{X}}_{k+1|k+1} = \overline{\boldsymbol{X}}_{k+1|k} + \boldsymbol{K}_k(\boldsymbol{Y}_{k+1} - \boldsymbol{h}(\overline{\boldsymbol{X}}_{k+1|k})) \\ \hat{\boldsymbol{P}}_{k+1|k+1} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\overline{\boldsymbol{P}}_{k+1|k} \end{cases} \tag{13}$$

After linearization, it can be obtained that $\boldsymbol{H}_k = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{X}_k}$, which can be calculated as follows:

$$\boldsymbol{H}_k = \begin{bmatrix} \frac{\Delta x}{l} & \frac{\Delta y}{l} & 0 & 0 \\ -\frac{\Delta y}{l^2} & \frac{\Delta x}{l^2} & 0 & 0 \end{bmatrix} \tag{14}$$

where $\Delta x = x_{k+1|k} - x_0$, $\Delta y = y_{k+1|k} - y_0$, and $l = \sqrt{(\Delta x)^2 + (\Delta y)^2}$.

During target tracking, the target missing probability is related to the target detecting probability and the precision of the motion state estimation. In other words, when the target is not missed, there are two requirements that need to be satisfied. One is that the target should be in the beam of the sensor, and the other one is that the target can be detected by the sensor.

As Figure 3 shows, the motion state estimation of target $t$ follows the Gaussian distribution $N(\hat{\boldsymbol{X}}_{k|k}, \hat{\boldsymbol{P}}_{k|k})$ [45]. Suppose that the beam $a$ is focused on $(\hat{x}_{k|k}, \hat{y}_{k|k})$, the probability that the target is in the beam $a$ is $p_g$, and the sensing radius of a sensor is $R$. $p_g$ can be calculated as follows:

$$p_g = \iint\limits_{\Omega_1} f(x, y)dxdy \tag{15}$$

$$f(x, y) = \frac{1}{\sqrt{2\pi}\hat{\sigma}_{k|k}^x} e^{-\frac{(x - \hat{x}_{k|k})^2}{(\hat{\sigma}_{k|k}^x)^2}} \cdot \frac{1}{\sqrt{2\pi}\hat{\sigma}_{k|k}^y} e^{-\frac{(y - \hat{y}_{k|k})^2}{(\hat{\sigma}_{k|k}^y)^2}} \tag{16}$$

where $\Omega_1$ is the covered domain by beam $a$.

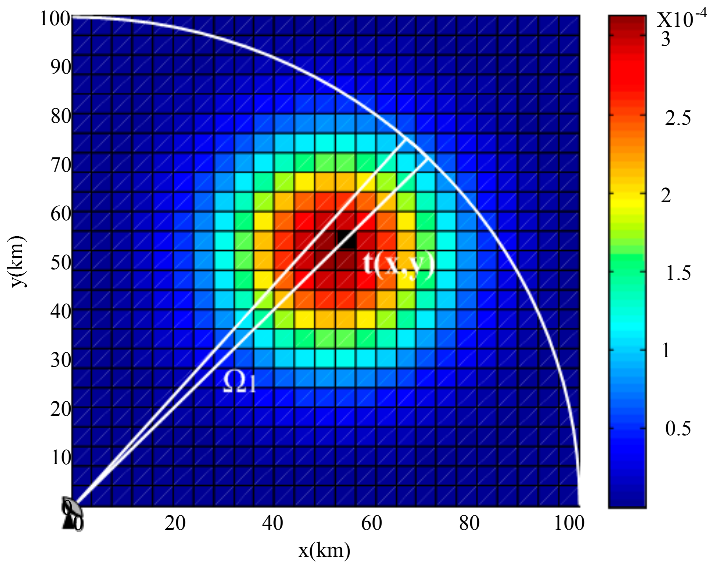



**Figure 3.** The domain covered by beam $a$ and the position distribution of target $t$.

In this paper, the width of beam $a$ is $\theta = 0.1$, and $R = 100$ km.

The target missing probability can be calculated by the following:

$$p_m = 1 - p_d p_g = 1 - 0.5\text{erfc}(\sqrt{-\ln p_f} - \sqrt{SNR + 0.5}) \iint_{\Omega} \frac{1}{\sqrt{2\pi}\hat{\sigma}^x_{k|k}} e^{-\frac{(x-\hat{x}_{k|k})^2}{(\hat{\sigma}^x_{k|k})^2}} \frac{1}{\sqrt{2\pi}\hat{\sigma}^y_{k|k}} e^{-\frac{(y-\hat{y}_{k|k})^2}{(\hat{\sigma}^y_{k|k})^2}} dxdy \tag{17}$$

### 3.2.2. Calculation of Target Threat Priority

In this paper, the target threat priority is related to the distance from the target to the monitoring center and the velocity of the target. If the target $t$ is in the monitoring domain $\Omega_0$, a shorter distance $d$ will result in higher target threat priority. Similarly, a larger velocity $||v||$ will lead to higher target threat priority. The target threat priority $\gamma$ of target $t$ can be calculated as follows:

$$\gamma = \begin{cases} \exp\left(\frac{-d^2}{(k_0||v||+m_0)^2}\right) \in (0,1), 0 \leq d \leq \alpha \\ 0, d > \alpha \end{cases} \tag{18}$$

where $k_0$ and $m_0$ are coefficients, and $k_0 = 1$ and $m_0 = 100$ in this paper.

The target threat priorities with different distances and velocities are displayed in Figure 4.

### 3.2.3. The Objective Function in Sensor Scheduling

In the sensor network, there are $m$ sensors applied to defend the domain $\Omega_0$, and at the time instant $k$, $n_k(m > n_k)$ targets are detected and tracked. Note that the sensor scheduling scheme is a $m \times n_k$ matrix $U_k$, where $u_k^{i,j}$ is the element in the $i$th row and $j$th column. When $u_k^{i,j} = 1$, sensor $s^i$ is used to track target $t^j$ at time $k$, and when $u_k^{i,j} = 0$, this means that sensor $s^i$ is not used to track target $t^j$ at the time instant $k$. When $\sum_{j=1}^{n} u_k^{i,j} = 0$, sensor $s^i$ is not used to track any target, but rather to detect targets.

Once a sensor is used to track a target, the target tracking precision can be improved with the target tracking risk reducing, but the sensor cannot monitor the domain and detect other targets any longer, so the latter emerging targets may be not detected by this sensor with the target detecting risk increasing. The two kinds of risk are not independent. Therefore, when scheduling sensors, both the target detecting risk and the target tracking risk should be taken into consideration.

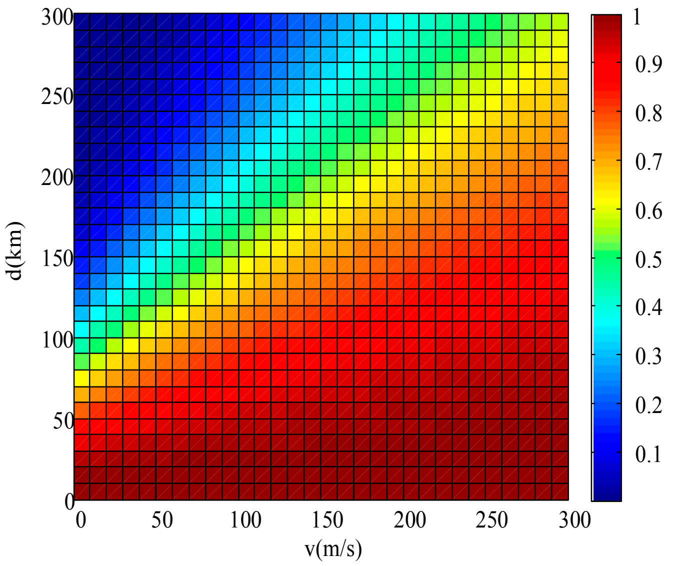



**Figure 4.** The target threat priorities with different distances and velocities.

As the target detecting risk and the target tracking risk are both added to the objective function when conducting sensor scheduling, the optimal sensor scheduling scheme can be calculated by the following:

$$\pi = \text{argmin}\{\beta_1 E(risk_1) + \beta_2 E(risk_2)\} = \text{argmin}\left\{\beta_1(\iint\limits_{\Omega_0} risk_1^c / S^{\Omega_0}) + \beta_2(\sum_{i=1}^{m}\sum_{j=1}^{n_k} u_k^{i,j} risk_2^{i,j}/n_k)\right\} \quad (19)$$

where $risk_1^c$ is the target detecting risk and can be calculated by Equation (9); $risk_2^{i,j} = p_m^{i,j}\gamma^j$ is the target tracking risk; $p_m^{i,j}$ is the target missing probability when target $t^j$ is being tracked by sensor $s^i$; and $\beta_1$ and $\beta_2$ are the weights, which represent the importance of the target detecting mission and the target tracking mission, respectively. In this paper, $\beta_1 = \beta_2 = 1$.

Subject to the following:

(1) Each target is tracked by only one sensor, therefore $\sum_{i=1}^{m} u_k^{i,j} = 1$.

(2) Each sensor processes two kinds of working mode, target detecting and target tracking, but it can select only one working mode at any moment. Furthermore, when a sensor selects the target tracking mode, it can track only one target, namely $\sum_{j=1}^{n_k} u_k^{i,j} \le 1$.

(3) Sensor $s^i$ can detect target $t^j$ at the time instant $k$ only if the target emerges in the detecting area of sensor $s^i$, namely $d_k^{i,j} \le R$.

## 4. Algorithm Design

The artificial bee colony (ABC) algorithm proposed by Karaboga [46] has been studied and applied widely, owning to its unique advantages like a simple form and easy implementation. However, it still has some disadvantages, including being prone to sticking into a local optimum and a slow convergence rate in late computation. By contrast, the particle swarm optimization (PSO) algorithm can effectively avoid these defects by applying the optimal value to update positions of particles. In this paper, ABC and POS are combined. To further improve the global searching capability of the algorithm, double-probability is brought in, thus the improved double-probability particle bee colony algorithm is proposed. The improved algorithm is exhibited as follows:

(1) $n_1$ nectar sources (feasible schemes) are first provided, and the $i$th nectar source can be given utilizing the following equation:

$$x_{ij} = x_{j\min} + rand(0,1)(x_{j\max} - x_{j\min}), i \in (1,2,\ldots,n_1), j \in (1,2,\ldots,n_2) \tag{20}$$

where $x_{ij}$ is the $j$th dimension value of $i$th nectar source; $x_{j\max}$ and $x_{j\min}$ are the maximum and minimum values of the $j$th dimension in the $i$th nectar source, respectively; $rand(0,1)$ is a random number satisfying uniform distribution between 0 to 1; and $n_2$ is the total dimensions of a feasible scheme.

(2) Each nectar source is assigned a searching bee and searched around $n_3$ times. Once the new source is better than the old one, the old source is replaced by the new one. The nectar source searching equation is shown as follows:

$$y_{ij} = x_{ij} + randn(-1,1)(x_{ij} - x_{iq}) \tag{21}$$

where $y_{ij}$ is the $j$th dimension value of the $i$th new nectar source, $x_{iq}$ is a random neighborhood of the current nectar source, and $randn(-1,1)$ is a random number satisfying the uniform distribution between $-1$ and 1.

(3) Calculate the fitness value of updated $n_1$ nectar sources according to Equation (10) or Equation (19) separately.

(4) Searching bees return to the honeycomb and change into leading bees. $n_4$ and $n_5$ following bees choose different leading bees according to the forward probability and reverse probability separately. The following bees go after their chosen leading bees to related nectar sources and search new sources according to Equation (21). The forward probability and reverse probability are shown by Equation (22) [47] and Equation (23).

$$p_i = \frac{f(X_i)}{\sum\limits_{i=1}^{n_1} f(X_i)} \tag{22}$$

$$q_i = \frac{1/f(X_i)}{\sum\limits_{i=1}^{n_1} 1/f(X_i)} \tag{23}$$

where $p_i$ and $q_i$ are the chosen probabilities of $i$th nectar source according to the forward probability and the reverse probability, respectively.

(5) If a nectar source has never been updated continuously up to $n_6$ times, this nectar source can be discarded, and a new nectar source can be obtained by Equation (20) or Equation (24) randomly.

$$\begin{cases} x_{ij} = x_{ij} + v_{ij} \\ v_{ij} = \sigma_1 v_{ij} + c_1 rand(0,1)(x_{pj} - x_{ij}) + c_2 rand(0,1)(x_{gj} - x_{ij}) \end{cases} \tag{24}$$

where $v_{ij}$ is the speed of $j$th dimension, $\sigma_1$ is the inertia weight, $c_1$ and $c_2$ are positive constants, $x_{pj}$ is the local optimum of $j$th dimension, and $x_{gj}$ is the global optimum of the $j$th dimension.

(6) Make a judgment of whether the maximum number of iterations is reached. If reached, end the algorithm, otherwise, go back to (3).

## 5. Simulations

### *5.1. Simulations of Sensor Deployment*

Before the sensor network is optimized, sensors in the networks are ordered as shown in Figure 5a. It can be found that in Figure 5a, 12 sensors are uniformly distributed in a 100 km radius circle. During each scanning period, the radiation time and radiation power are $\tau_{cal} = 10$ ms and $P_{cal} = 1$ kW, respectively. The target detecting probability in $\Omega_0$ is distributed as shown in Figure 5a, and the target detecting risk in $\Omega_0$ is distributed as shown in Figure 5b. Before optimization, the average target detecting risk of $\Omega_0$ is 0.2922.

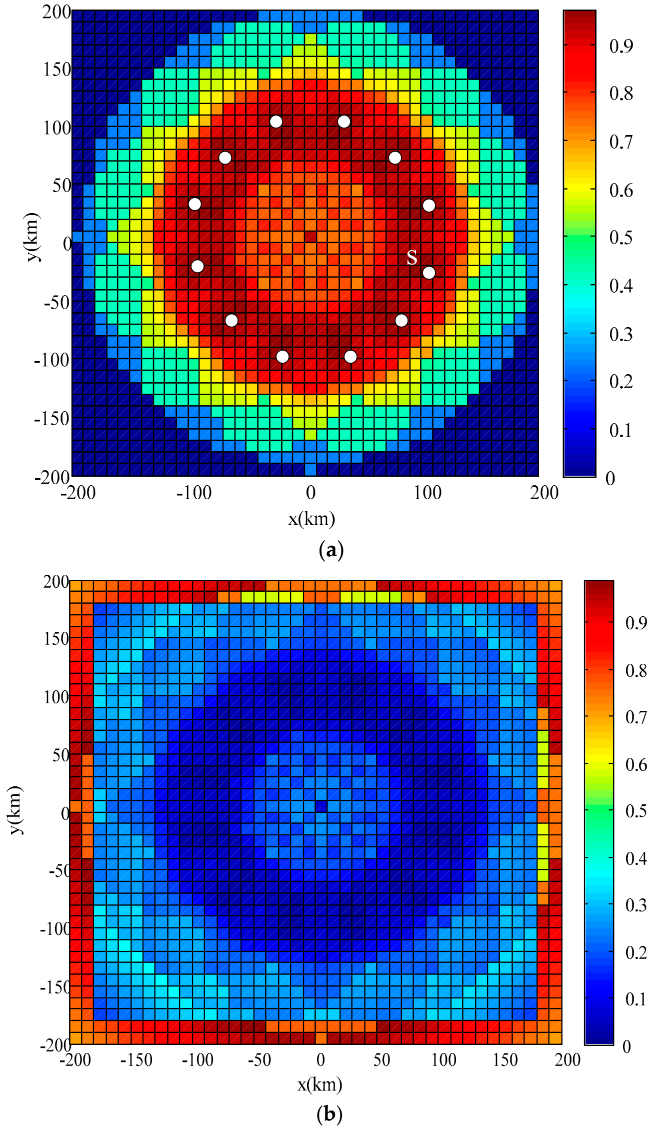



**Figure 5.** Sensor deployment before optimization. (**a**) Target detecting probability distribution; (**b**) target detecting risk distribution.

Three different algorithms are applied to optimize the sensor distribution, and the three algorithm iteration curves are shown in Figure 6. It should be noted that algorithm 1 denotes the double-probability particle bee colony algorithm, algorithm 2 denotes the basic artificial bee colony algorithm, and algorithm 3 denotes the improved bee colony algorithm in another paper [47]. In the calculations, the number of leading bees is 10, the number of following bees is 50, and the number of calculation iterations is 100.

It can be seen from Figure 6 that algorithm 1 outperforms algorithm 2 and algorithm 3 in convergence speed and solution quality. After optimization, the average target detecting risk of $\Omega_0$ is 0.2658, and the optimized sensor deployment (target detecting probability distribution) and detecting risk distribution are shown in Figure 7.

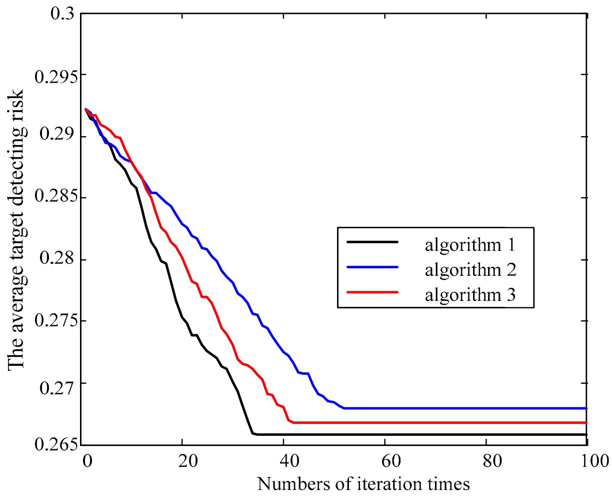
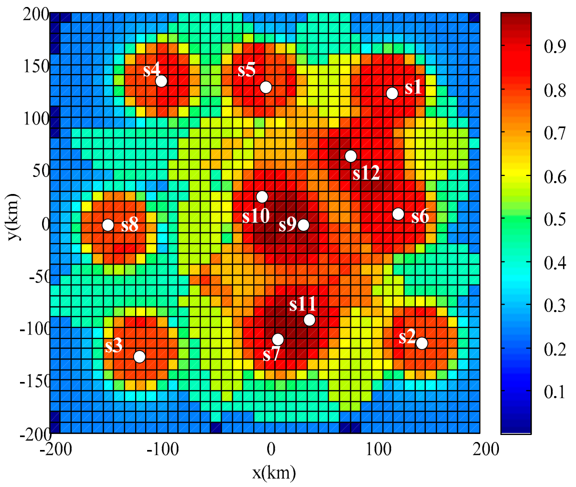
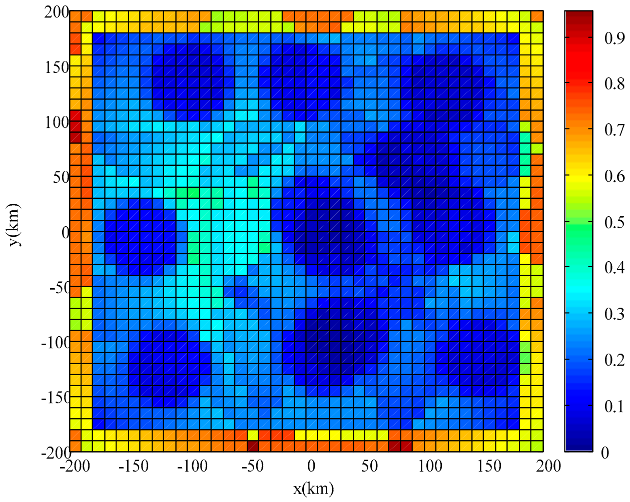



**Figure 6.** Algorithm iteration curves.




(**a**)




(**b**)

**Figure 7.** Sensor deployment after optimization. (**a**) Target detecting probability distribution; (**b**) target detecting risk distribution.

The sensor deployment scheme (coordinates of 12 sensors) calculated by algorithm 1 is shown as follows:

$s_1$ (115, 112), $s_2$ (138, −124), $s_3$ (−128, −121), $s_4$ (−118, 131), $s_5$ (3, 118), $s_6$ (132, −8), $s_7$ (10, −113), $s_8$ (−126, 5), $s_9$ (10, −2), $s_{10}$ (−1, 14), $s_{11}$ (12, −96), $s_{12}$ (85, 78).

From the comparison of Figures 5 and 7, it can be concluded that after optimization, except for gathering at the center of the domain, sensors also tend to move to margins where monitoring priorities are also pretty high.

### *5.2. Simulations of Sensor Scheduling*

At the time instant $k = 0$, four targets were detected in the domain $\Omega_0$, and their initial motion states are as follows:

$\hat{x}^1_{0|0} = (-168, 0.3, 48, 0.4)^T$; $\hat{x}^2_{0|0} = (-81, 0.5, -154, 0.5)^T$; $\hat{x}^3_{0|0} = (176, -0.7, -38, 0.1)^T$; $\hat{x}^4_{0|0} = (32, 0, 165, -0.5)^T$.

Applying the three algorithms to calculate sensor scheduling scheme, the calculation results are shown in Figure 8. In the calculations, the number of leading bees is 10, the number of following bees is 50, and the number of calculation iterations is 15.

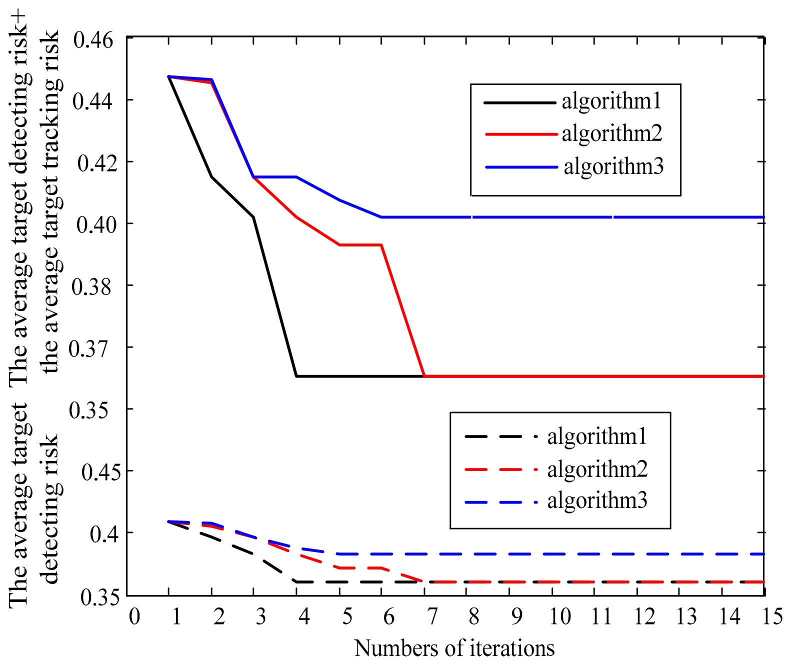
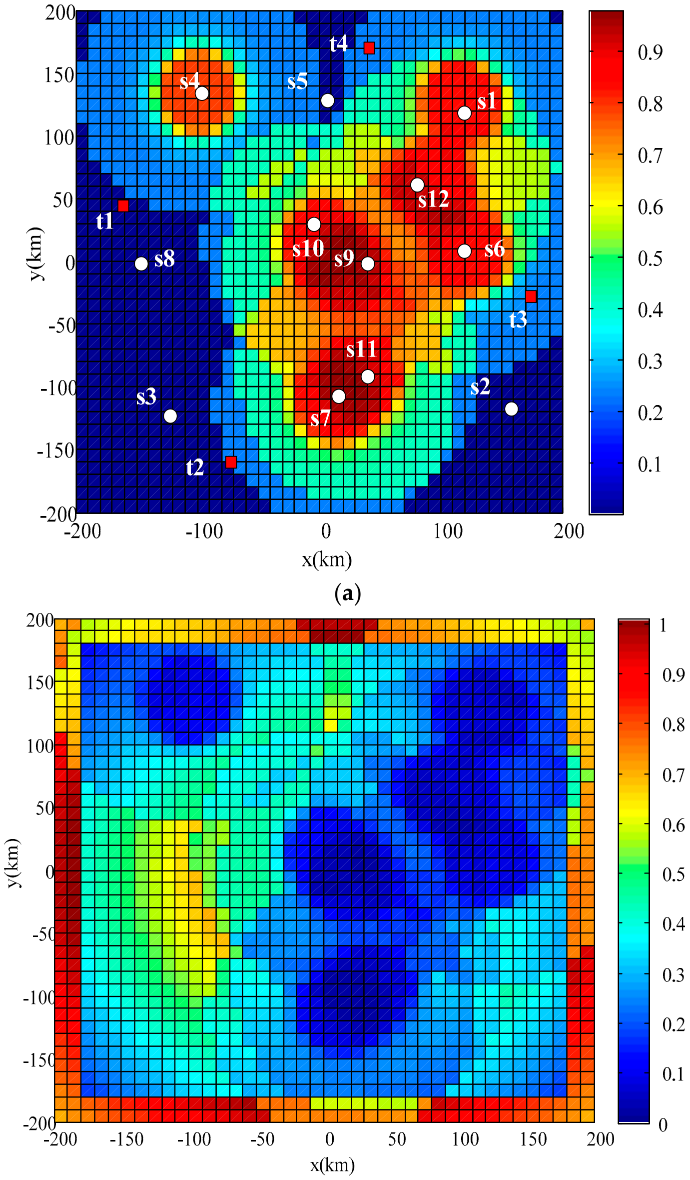



**Figure 8.** Algorithm iteration curves.

It can be seen from Figure 8 that algorithm 1 outperforms algorithm 2 and algorithm 3 in convergence speed and solution quality. The numbers of iterations in solving the problem of sensor scheduling is greatly reduced in comparison with that in solving the problem of sensor deployment. The reason for this is that while conducting sensor scheduling, sensors can only detect targets that are in their sensing domains. Therefore, although the total number of different kinds of sensor scheduling schemes is 12!/8! using 12 sensors to detect 4 targets, most of the schemes can be kicked off. At the time instant $k = 0$, target $t^1$ can be detected by sensors $\{s^4, s^8\}$, target $t^2$ can be detected by sensors $\{s^3, s^7\}$, target $t^3$ can be detected by sensors $\{s^2, s^6\}$, and target $t^4$ can be detected by sensors $\{s^1, s^5, s^{12}\}$. The number of choices in sensor scheduling is actually only 24.

While sensors are scheduled to track targets, the average target detecting risk of $\Omega_0$ is 0.3605, the average target tracking risk is 0.2201, and the total risk is 0.5806. The sensor–target allocation scheme is as follows: $t^1 - s^8$, $t^2 - s^3$, $t^3 - s^2$, $t^4 - s^5$.

The target detecting probability in $\Omega_0$ is shown in Figure 9a, and the target detecting risk in $\Omega_0$ is distributed as shown in Figure 9b.




(**a**)

(**b**)

**Figure 9.** Sensor deployment after scheduling. (**a**) Target detecting probability distribution; (**b**) target detecting risk distribution.

Comparing Figures 7 and 9, it can be discovered that with the appearance of the target, the detecting capability of the sensor network decreases and the target detecting risk increases due to undertaking the mission of target tracking.

## 5.3. Simulations of Sensor Scheduling during a Time Period

In a time period $k \sim [0, 50T](T = 1s)$, the sensor scheduling schemes vary as shown in Figure 10. Figure 10a–d show the target trajectories, sensor–target allocations, target threat priorities, and risks in the mentioned time period of $k \sim [0, 50T]$, respectively.

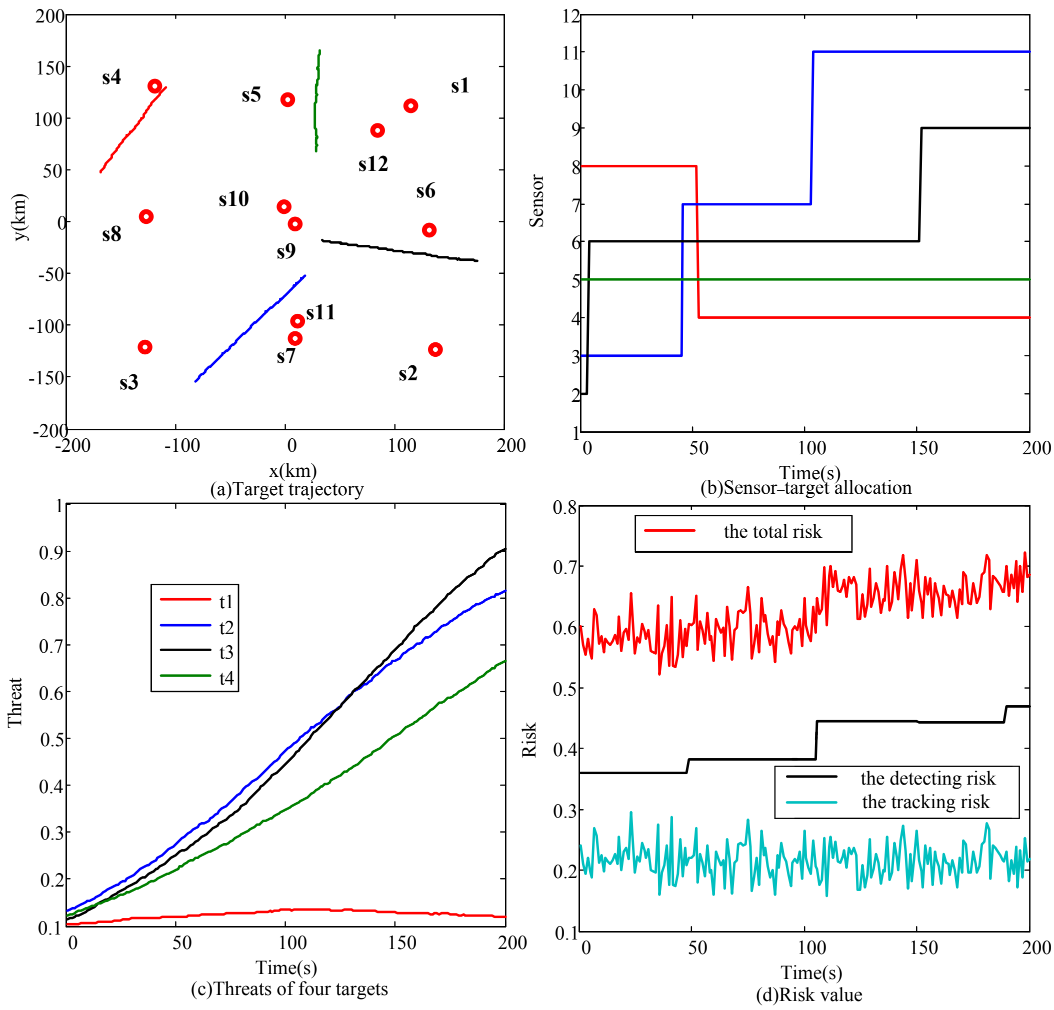



**Figure 10.** Sensor scheduling process.

Comparing Figures 10a and 10c, as the targets move close to the monitoring center, the target threat priorities increase. In the comparison of Figures 10b and 10d, once the sensor–target allocation scheme changes, the sensor detecting risk changes as well, and when targets are close to the monitoring center, sensors near the monitoring center are used to track targets, thus the target detecting risk increases prominently. From Figure 10d, compared with the target detecting risk, the target tracking risk performs more randomly. The reason for this is that the tracking risk is not only related to the target detecting probability, but also to the precision of target motion state estimation, which performs more randomly. It will also be transmitted to the total risk (the sum of the target detecting risk and the target tracking risk).

### *5.4. Influences of the Weights to Sensor Scheduling Schemes*

In the above simulations, the weights are set as $\beta_1 = \beta_2 = 1$, that is to say, the target detecting risk and the target tracking risk are of equal importance. In this section, the influences of weights on sensor scheduling schemes are studied.

The initial motion state of target $t^5$ is $\hat{x}^5_{0|0} = (40, 0, 200, -1)^T$ at the time instant $k = 0$. In the simulation, three different cases in which $\beta_1 = 1\beta_2 = 0$, $\beta_1 = 0\beta_2 = 1$, and $\beta_1 = \beta_2 = 1$ are considered. In these cases, sensor scheduling processes during the time period of $k \sim [0, 400]$ are shown in Figure 11.

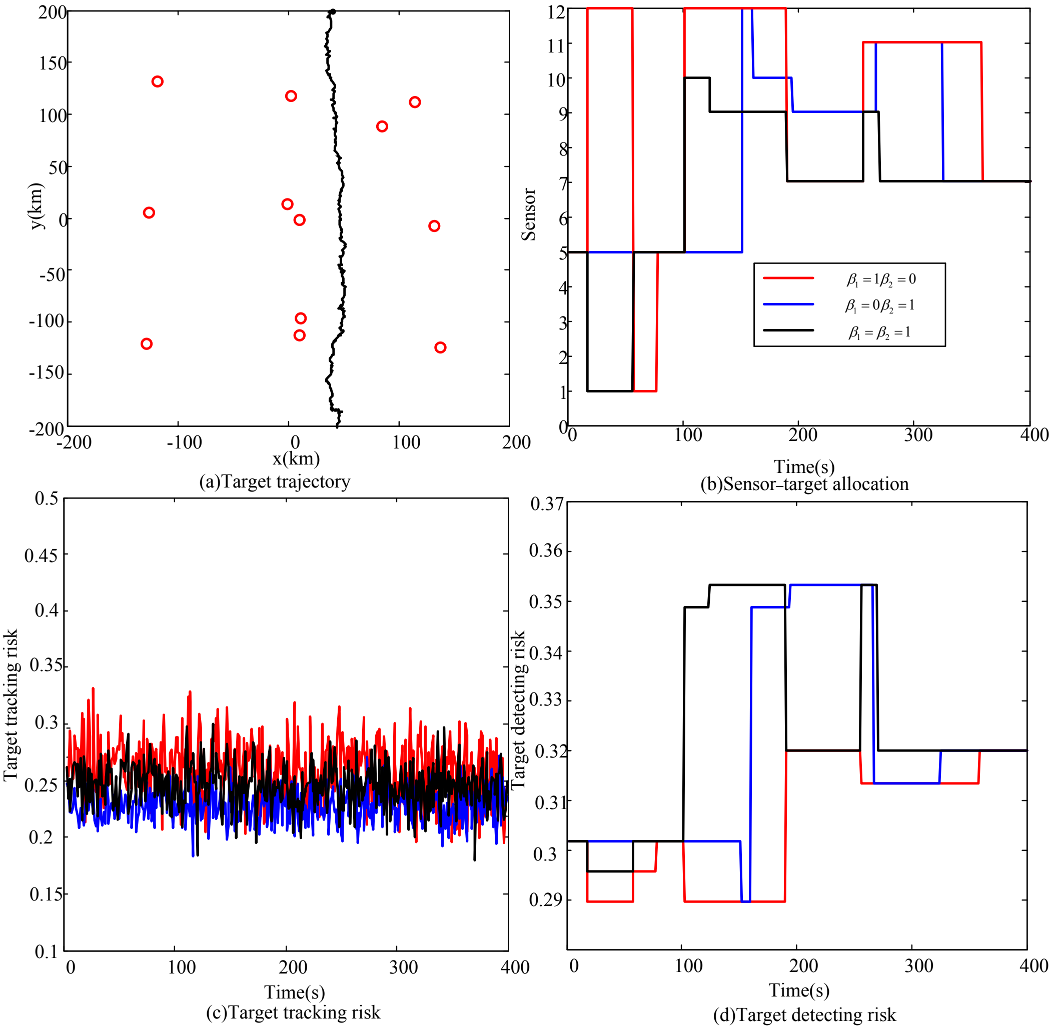



**Figure 11.** Comparisons of sensor scheduling in different sensor weights.

Figure 11a–d show the target trajectory, sensor–target allocations, target tracking risks, and target detecting risks in the time period of $k \sim [0, 400]$, respectively.

From Figure 11, when $\beta_1 = 1\beta_2 = 0$, sensors are scheduled in obedience with the minimum target detecting risk rule, and in this case, the target detecting risk is the minimum, but the target tracking risk is not the most optimal. When $\beta_1 = 0\beta_2 = 1$, sensors are scheduled in obedience with the rule of the minimum target tracking risk, and in this case, the target tracking risk is minimal, but the target detecting risk is not the most optimal. When $\beta_1 = \beta_2 = 1$, sensors are scheduled in obedience with the minimum total risk rule (the sum of the target detecting risk and the target tracking risk), and the target tracking risk and the target detecting risk are taken into account at the same time.

Furthermore, in contrast, sensor scheduling models are established for individual combat missions in the literature [1–25], belonging to the first two sensor scheduling approaches mentioned in the introduction. However, obviously, because of the various situations in the battle field, targets do not reach the monitoring domain simultaneously. All targets must be detected and tracked upon arriving. Therefore, during target-tracking missions, the sensor network should also keep watching to capture unknown targets simultaneously—only by doing so, targets can be detected and tracked in real time. As for target detecting and tracking discussed in this paper, taking the first two sensor scheduling approaches, utilizing the technical indicators—which are target detecting probability and target tracking precision in these cases—to model the objective function following a weighted-sum approach, the combination cannot make physical or operational sense [26]. Only the risk-based sensor approach can be better applied to sensor scheduling in the case of multi-missions, with their physical sense being risk.

## 6. Conclusions

In this paper, a sensor management (including sensor deployment and sensor scheduling) approach based on risk theory is proposed, considering both the target detecting and tracking. Firstly, the sensor deployment approach is studied. The target detecting risk is defined and its computing method is provided. The minimum target detecting risk rule is employed to establish the sensor deployment model. Secondly, the sensor scheduling approach is studied, with the target tracking risk defined and its computing method proposed. The sensor scheduling model is established, combining both the target detecting risk and the target tracking risk. Furthermore, in order to solve the sensor deployment and scheduling objective functions, an improved artificial bee colony algorithm, namely the double-probability particle bee colony algorithm, is designed. Finally, simulations are conducted and the experiment results indicate that the models and the algorithm in this paper have some advantages over the existing ones. With combat situations becoming more complex and more diverse, sensor scheduling models aiming at an individual combat missions seem to struggle to meet current requirements. Therefore, increasing attention is being paid by researchers across the world to solve the fairly difficult problem. In the future, further study will be conducted on combinations of target tracking, target recognition, and target threat priority assessment.

**Author Contributions:** All authors contributed significantly to the writing and/or final editing of the manuscript. C.P. conceived the ideas of theory and wrote the paper. G.S. final edited the paper. X.D. and G.X. Checked and corrected the paper.

**Funding:** This research was funded by DPRFC (Defense Pre-Research Fund Project of China), grant number 012015012600A2203, and NSFC (Natural Science Foundation of China), grant number 61573374.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Nash, J.M. Optimal allocation of tracking resource. In Proceedings of the IEEE Conference on Decision and Control, New Orleans, LA, USA, 7–9 December 1977; pp. 1177–1180.
2. Evans, J.; Krishnamurthy, V. Optimal sensor scheduling for Hidden Markov Model state estimation. *Int. J. Control* **2001**, *74*, 1737–1742. [CrossRef]
3. Chhetri, A.S.; Orrell, D.; Papandreou, S.A. On the use of binary programming for sensor scheduling. *IEEE Trans. Signal Process.* **2007**, *55*, 2826–2839. [CrossRef]
4. Ahmad, B.A.; Syed, T.J.; Stephen, L.S. A complete greedy algorithm for infinite-horizon sensor scheduling. *Automatica* **2017**, *281*, 335–341.
5. Castanon, D.A. Approximate dynamic programming for sensor management. In Proceedings of the 36th IEEE Conference on Decision and Control, New Orleans, LA, USA, 7–9 December 1977; Volume 2, pp. 1202–1207.
6. Hoang, H.G.; Vo, B.T. Sensor management for multi-target tracking via multi-Bernoulli filtering. *Automatica* **2014**, *50*, 1135–1142. [CrossRef]
7. Kevin, M.N.; Jonathan, T.B. An autonomous sensor management strategy for monitoring a dynamic space domain with diverse sensors. In Proceedings of the 2018 AIAA Information Systems, Kissimmee, FL, USA, 8–12 January 2018; pp. 1–29.
8. Fang, D.L.; Ren, X.M.; Li, O. Energy efficient distributed sensor management algorithm. *J. Xidian Univ.* **2017**, *44*, 171–177.
9. Saurav, S.; Yimin, D.Z.; Moeness, G.A. Cramer–Rao type bounds for sparsity-aware multi-sensor multi-target tracking. *Signal Process.* **2018**, *145*, 68–77.
10. Yang, C.; Kaplan, L.; Blasch, E. Performance measures of covariance and information matrices in resource management for target state estimation. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 2594–2613. [CrossRef]
11. Siddharth, J.; Stephen, B. Sensor selection via convex optimization. *IEEE Trans. Signal Process.* **2009**, *57*, 451–462.

12. Wayes, T.; Nipun, W.; Wen, T.L.; Chau, Y.H.; Vincent, P.; Tapan, K.S.; Kristin, L.W. Internet of things for green building management: Disruptive innovations through low-cost sensor technology and artificial intelligence. *IEEE Signal Process. Mag.* **2018**, *3*, 100–110.
13. Yan, T.; Han, C.Z. Sensor management for multi-target detection and tracking based on PCRLB. In Proceedings of the 2017 20th International Conference on Information Fusion, Xi'an, China, 10–13 July 2017.
14. Hong, H.G.; Maule, R. The Cauchy-Schwarz divergence for Poisson point process. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *61*, 4475–4485.
15. Tharmarasa, R.; Kirubajan, T. Decentralized sensor selection for large-scale multisensor-multitarget tracking. *IEEE Trans. Aerosp. Electrion. Syst.* **2011**, *47*, 1307–1324. [CrossRef]
16. Mario, B.I.M.; Ivan, P. Score matching based assumed fensity filtering with the von mises-Fisher distributed. In Proceedings of the 2017 20th International Conference on Information Fusion, Xi'an, China, 10–13 July 2017; pp. 433–438.
17. Sayin, M.O.; Lin, C.W.; Shiraishi, S.; Shen, J.; Başar, T. Information-driven autonomous intersection control via incentive compatible mechanisms. *IEEE Trans. Intell. Transp. Syst.* **2018**, *99*, 1–13. [CrossRef]
18. Ferri, G.; Andera, M.; Kevin, D.L. An autonomous underwater vehicle data-driven control strategy for target tracking. *IEEE J. Ocean. Eng.* **2018**, *43*, 323–343. [CrossRef]
19. Li, Y.; Devesh, K.; Jha, A.R.; Thomas, A.W. Information-theoretic performance analysis of sensor networks via Markov modeling of time series data. *IEEE Trans. Cybern.* **2018**, *48*, 1898–1909. [CrossRef] [PubMed]
20. Li, W.; Han, C.Z. Dual sensor control scheme for multi-target tracking. *Sensors* **2018**, *18*, 1653. [CrossRef] [PubMed]
21. Bolderheij, F.; Absil, F.; Van, G.P. A risk-based object-oriented approach to sensor management. In Proceedings of the 8th International Conference on Information Fusion, Philadelphia, PA, USA, 25–29 July 2005; Volume 1, p. 8.
22. Wang, Y.; Hussein, I.I.; Erwin, R.S. Risk-based sensor management for integrated detection and eistimation. In Proceedings of the 2011 American Control Conference, San Francisco, CA, USA, 29 June 29–1 July 2011; pp. 168–189.
23. Sean, M. Risk-based sensor resource management for field of view constraint sensor. In Proceedings of the 18th International Conference on Information Fusion, Washington, DC, USA, 6–9 July 2015; pp. 2041–2048.
24. Marcos, E.G.B.; Dominique, M.; Philippe, V. Sensor management using expected risk reduction approach. In Proceedings of the 19th International Conference on Information Fusion, Heidelberg, Germany, 5–8 July 2016; pp. 2050–2058.
25. Marcos, E.G.B.; Dominique, M.; Philippe, V. A risk-based sensor management using random finite sets and POMDP. In Proceedings of the 20th International Conference on Information Fusion, Xi'an, China, 10–13 July 2017; pp. 1588–1596.
26. Fptios, K.; Hans, D.; Alexander, Y. Threat-based sensor management for target tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 2772–2784.
27. Gostar, A.; Hoseinnezhad, R.; Weifebg, L. Sensor-management for multi-target filters via minimization of posterior dispersion. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 2877–2884. [CrossRef]
28. Ozturk, O.; Begen, M.A.; Zaric, G.S. A branch and bound algorithm for scheduling unit dize jobs on parallel batching machines to minimize makespan. *Int. J. Prod. Res.* **2017**, *55*, 1815–1831. [CrossRef]
29. Stiven, S.D.; Marcelo, G.S.B. Cooperative target tracking using decentralized particle filtering and RSS sensors. *IEEE Trans. Signal Process.* **2013**, *61*, 3632–3646.
30. Mohammed, A.Z.; Nabil, Z.S.; Shigenobu, S.; Sabah, M.A. A centralized immune-voronoi deployment algorithm for coverage maximization and energy conservation in mobile wireless sensor networks. *Information Fusion.* **2016**, *30*, 36–51.
31. Tkach, I.; Jevtić, A.; Nof, S.Y.; Edan, Y. A modified distributed bees algorithm for multi-sensor task allocation. *Sensors* **2018**, *18*, 759. [CrossRef] [PubMed]
32. Sang, J.Y.; Anish, P.S.; Seo, M.; Han, C.H.; Minho, P.; Lee, K.E. Joint spectrum sensing and resource allocation optimization using genetic algorithm for frequency hopping–based cognitive radio networks. *Int. J. Commun. Syst.* **2018**, *31*, e3733.
33. Vaishali, R.; Kulkarni, V.D. Computational intelligence for localization of mobile wireless sensor networks. *Adv. Intell. Syst. Comput.* **2018**, *2*, 449–463.

34. Liu, X.M.; Guo, Y.; Li, W.; Hua, M.; Ding, E.J. A complete feasible and nodes-grouped scheduling algorithm for wireless rechargeable sensor networks in tunnels. *Sensors* **2018**, *18*, 3410. [CrossRef] [PubMed]
35. Enrique, C.N.; Alfredo, G.; Lin, C.Y. A game-theoretic approach to efficient power management in densor networks. *Oper. Res.* **2008**, *56*, 552–556.
36. Fu, Y.F.; Ling, Q.; Tian, Z. Distributed sensor allocation for multi-target tracking in wireless sensor networks. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 3538–3553. [CrossRef]
37. Severson, T.A.; Derek, A.P. Distributed multitarget search and track assignment with consensus-based coordination. *IEEE Sens. J.* **2015**, *15*, 864–875. [CrossRef]
38. Peyman, N.; Mahmoud, N.Z.; Saeid, A.; Yaghmaee, M.H. Distributed clustering-task scheduling for wireless sensor networks using dynamic hyper round policy. *IEEE Trans. Mob. Comput.* **2018**, *17*, 334–347.
39. Jia, B.; Pham, K.D.; Blasch, E.; Shen, D. Consensus-based auction algorithm for distributed sensor management in space object tracking. In Proceedings of the 2017 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2017; pp. 1–8.
40. Hemalatha, S.V.; Malathi, S.; Scholar, P.G. Auction based patient health detection scheme using group management techniques in wireless sensor networks. *Int. J. Eng. Sci. Comput.* **2018**, *8*, 18342–18345.
41. Rahmani, H.A.A.; Babaee, T.E. A gravitational emulation local search algorithm for task scheduling in multi-agent system. *Int. J. Appl. Optim. Stud.* **2018**, *1*, 11–24.
42. Mahafza, B.R. *Radar Systems Analysis and Design Using MATLAB*, 2nd ed.; Publishing House of Electronics Industry: Beijing, China, 2008; pp. 201–231.
43. Alexander, G.; Voronovich; Valery, U.Z. Bistatic radar equation for signals of opportunity revisited. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 1959–1968.
44. Alexey, S.; Narykov; Alexander, Y. Sensor selection algorithm for optimal management of the tracking capability in multisensor radar system. In Proceedings of the 2013 European Microwave Conference, Nuremberg, Germany, 6–10 October 2013; pp. 499–502.
45. James, Z.; Hare, S.G.; Thomas, A. POSE: Prediction-based opportunistic sensing for energy efficiency in sensor networks using distributed supervisors. *IEEE Trans. Cybern.* **2017**, *48*, 2114–2127.
46. Kanaboga, D.; Basturk, B. A comparative study of artificial bee colony algorithm. *Appl. Soft Comput.* **2008**, *8*, 687–696. [CrossRef]
47. Ning, J.; Liu, T.T.; Zhang, C.; Zhang, B. Source-updating information-guided artificial bee colony algorithm. *Neural Comput. Appl.* **2018**, *30*, 775–787. [CrossRef]