

Research Article

Bayesian-Based Search Decision Framework and Search Strategy Analysis in Probabilistic Search

Liang Yu  and Da Lin 

School of Information Engineering (School of Big Data), Xuzhou University of Technology, Xuzhou 221000, Jiangsu, China

Correspondence should be addressed to Da Lin; 971244320@qq.com

Received 15 September 2020; Revised 20 October 2020; Accepted 2 November 2020; Published 18 November 2020

Academic Editor: Wenzheng Bao

Copyright © 2020 Liang Yu and Da Lin. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a sequence decision framework based on the Bayesian search is proposed to solve the problem of using an autonomous system to search for the missing target in an unknown environment. In the task, search cost and search efficiency are two competing requirements because they are closely related to the search task. Especially in the actual search task, the sensor assembled by the searcher is not perfect, so an effective search strategy is needed to guide the search agent to perform the task. Meanwhile, the decision-making method is crucial for the search agent. If the search agent fully trusts the feedback information of the sensor, the search task will end when the target is “detected” for the first time, which means it must take the risk of founding a wrong target. Conversely, if the search agent does not trust the feedback information of the sensor, it will most likely miss the real target, which will waste a lot of search resources and time. Based on the existing work, this paper proposes two search strategies and an improved algorithm. Compared with other search methods, the proposed strategies greatly improve the efficiency of unmanned search. Finally, the numerical simulations are provided to demonstrate the effectiveness of the search strategies.

1. Introduction

Unmanned search and rescue is a highly autonomous task and there are many cases of such spatial search problems [1–3], such as resource exploration, sea fishing, border patrols, search fugitive, and troubleshooting. Integrated high-efficiency mobile processor platform, effective sensor, and data fusion algorithms make possible the implementation of these highly autonomous tasks [4–6]. In the above tasks, probabilistic information is often used to describe the likelihood that the target is in a different location. However, due to the limitation of sensor accuracy and complex external interference, search agents cannot always obtain the correct information; although the search agent can update the status of a target location by collecting and processing incomplete observations, an appropriate search strategy is still needed to guide it when and where to detect [7, 8]. Besides, time is one of the key factors in search tasks, especially in rescue tasks or disaster management. As time goes by and the external uncertain interference, the position

of the target will become more and more uncertain, which greatly increases the difficulty of the task.

Therefore, there is an urgent need for a general framework which can integrate the probability characteristics of the search area and deal with erroneous observations. In order to solve the above problem, this paper proposes a Bayesian-based search decision framework and two adaptive strategies to guide the search agents to find a static target in an unknown place as soon as possible [9, 10]. A brief summary of prior literature: the classical search theory was introduced by Koopman during World War II [11], which focuses on using aircraft and warships to find enemy submarines in the shortest possible time. After that, the search theory has been extensively generalized by Stone [12].

In recent years, many researchers have treated search problems as a decision-making problem rather than an information collection task [13–18]. In decision theory, the search problem is considered as a decision between the current state of knowledge and the hypothesis of the

decision-maker. Focusing on the problem of how to manage mobile agent to search and track multiple static targets, a perception-based decision was developed for the static objects in [19]; although this method can guarantee tracking the state of the target in a short time, it still lacks the analysis of decision evolution. In order to compare the impact of different strategies on the search process, a Bayesian-based search framework is proposed in [20], which provides a platform for comparison of search methods. In addition, inspired by Thrun et al., the probabilistic approach [21] has arisen in the robotics community, the core of probabilistic robots is the idea of estimating states from sensor data, and the probability mass function (PMF) is used to represent the search agent's understanding of the environment in [22].

Many early studies are based on the assumption that the sensor has no false positives, including [23]. Although some scholars are devoted to solving the problems of false positives and false negatives in the search task [24, 25], they all assume that the cells in the search area are independent; this assumption leads to the inability to integrate the relevant information of the search into the search plan in time [26]. Aiming at the scenario of using a drone cluster to find targets in a hazardous environment, a collaborative search strategy for drones is proposed in [27], which instructs searchers to gradually move from one unit to the next to ensure that the search area is covered. The influence of the heuristic information on search agents was studied in [28]; Lanillos et al. compared the search strategy with heuristic information and the search strategy without heuristic information. The results show that the search strategy with the heuristic information can effectively avoid the search agent falling into the local optimal position. At the same time, some novel search strategies (i.e., random jump search, snapshot search, and drosophila-inspired search) were proposed and discussed in [22], but there is a lack of motion restrictions on the search agent when analyzing these strategies. Furthermore, relevant search strategies were divided into two categories in [29]. One type of search strategy is called the nonadaptive search strategy; it does not consider reoptimizing the search path but only consider collecting information. Another type of search strategy is called the adaptive search strategy, which updates the search path through the feedback of the current search information, which greatly improves the search performance. In order to prevent the collision between robots in the process of searching, a new distributed covering method of the mobile deformable convex region is proposed in [30]. The concept of the minimum expected time was proposed in [31], it is used to indicate the time required to complete the search task.

Although there have been many notable achievements in search theory [32, 33], there is still room for improvement; for example, the search plan was optimized using the cumulative detection probability in [23], but the sensor's false positive error has not been considered in the search process. In addition, when the search agent needs to check some places far away from itself, it needs some path planning algorithms to guide it; this process can consider using Dijkstra's algorithm [34], A* algorithm [35], or rapidly exploring random tree algorithm [36].

Contribution of this paper: based on the previous work, a sequential decision-making search framework and two adaptive search strategies are proposed in this paper. The main difficulty of this paper is that the search agent can only move a fixed distance at a time, and the sensor is not perfect. Compared with other works, the main contributions of this article are concentrated in the following four aspects:

- (i) In the search process, not only the movement ability of the motion agent but also various errors of the sensor are considered.
- (ii) In this paper, the evolution expression of sequence decision is derived, and a Bayesian-based search decision framework is proposed to deal with the incomplete information detected by the search agent.
- (iii) The evolution of the search decision is analyzed quantitatively from the mathematical expression, two key factors affecting the decision are obtained, and two effective adaptive search strategies are proposed according to the characteristics of these two factors.
- (iv) A repetitive detection mechanism is proposed to deal with imperfect observations of sensors, which saves search resources to a certain extent and prevents search agents from falling into a locally optimal position.

Organization: the remainder of this paper is organized as follows. In Section 2, the knowledge of the search problem is introduced and a Bayesian-based search decision framework is proposed. Through the analysis of decision evolution, two effective adaptive search strategies are proposed and analyzed in Section 3. The numerical simulation results are presented in Section 4. Section 5 concludes the paper with closing remarks and avenues of future research.

2. Problem Formulation

In this section, the preliminary knowledge of the search problem and search decision-making framework are presented. In the search decision-making framework, the uncertain state of the target is expressed as a PMF, the search agent combines the new information with the prior information in the form of probability and updates the knowledge state with Bayesian rules to form a new posterior PMF.

2.1. Search Area. Consider an immobile object x_T lost in region \mathcal{A} , the search area \mathcal{A} can be divided into $|\mathcal{A}|$ disjoint grid cells. Figure 1 shows the grid division of a square area. It is important to note that x_T is in the discrete grid and not on the grid boundary. $x_T \in \mathcal{A}$ means the target x_T exists in this area \mathcal{A} . Conversely, $x_T \notin \mathcal{A}$ indicates that the target is not in the area. Hence, we can use a Bernoulli random variable H to indicate whether the target x_T is really in the region \mathcal{A} :

$$H = \begin{cases} 0, & \text{if } x_T \notin \mathcal{A}, \\ 1, & \text{if } x_T \in \mathcal{A}. \end{cases} \quad (1)$$

Furthermore, use the variable $x_{a_{th}}$ to indicate whether the target is in the $x_{a_{th}}$ cell. If $x_{a_{th}} = 1$, it means that the target x_T

is in the $x_{a_{th}}$ cell; on the contrary, $x_{a_{th}} = 0$ means that the target is not here:

$$x_{a_{th}} = \begin{cases} 0, & \text{if } x_T \neq x_{a_{th}}, \\ 1, & \text{if } x_T = x_{a_{th}}. \end{cases} \quad (2)$$

2.2. Search Model. In the search process, the information detected by the sensor may be incorrect due to the false positive or false positive errors, and the $d_a(t) \in \{0, 1\}$ is used to represent the detection result of the search agent in the a_{th} grid at the time t ; for convenience, a_{th} is abbreviated as a . Therefore, imperfect sensors can be modeled as follows:

$$\Pr(d_a(t)|x_T) = \begin{cases} \Pr(d_a(t) = 1 | x_a(t) = 0) = \alpha, \\ \Pr(d_a(t) = 1 | x_a(t) = 1) = \beta, \\ \Pr(d_a(t) = 0 | x_a(t) = 0) = 1 - \alpha, \\ \Pr(d_a(t) = 0 | x_a(t) = 1) = 1 - \beta, \end{cases} \quad (3)$$

where the error probabilities α and β quantify the characteristics of the imperfect sensing capabilities, which can be determined by experiment or sensor specifications. Note that the condition $\alpha + \beta \leq 1$ must be set here; otherwise, the search agent will not be able to get valid information.

2.3. Search Agent Motion Model. Due to the limited speed of the search agent and the incomplete sensing function, each cell can only be detected one by one. Figure 2 shows a search graph where each cell is connected to all adjacent cells, and the search agent can move between two vertices that are connected. When the search agent determines that a certain cell is a possible target location, it will use the Dijkstra algorithm to build the shortest path.

2.4. Bayesian Update of the Belief Map. The Bayesian approach provides an effective way to maintain and update all the quantitative and qualitative information related to search [23]. In the search task, the search agent marks a corresponding probability value where the target may appear. Then, the search agent collects a set of observation sequences $D(t) = \{d_1, \dots, d_t\}$; through this imperfect detection information, it can have a deeper understanding of the real state of the target. At the same time, the $\mathbf{B}(t)$ is used to represent the aggregate belief, which is defined as

$$\mathbf{B}(t) = \sum_{a=1}^{|\mathcal{A}|} \Pr(x_T = a | D(t)). \quad (4)$$

When the search task is launched, the search agent will have an initial aggregate belief ($\mathbf{B}(0) = \delta$, $0 < \delta \leq 1$); the initial aggregation belief is usually given by experience. In the belief map, each cell contains a confidence value, which represents the probability of target x_T in it. The recursive Bayesian approach provides a simple but effective way for updating the belief map; after the search agent obtains the detection result, we use it to update the belief map. The first step involves a simple application of the Bayesian rule to the individual cell belief:

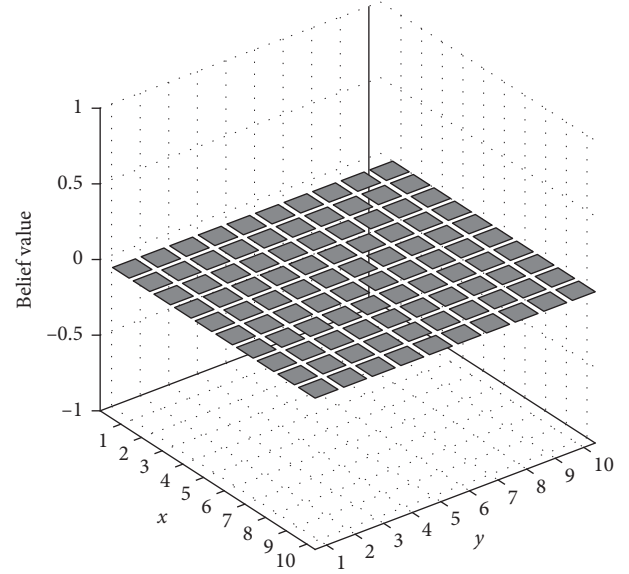


FIGURE 1: Search area discretization.

$$\begin{aligned} P_r(x_T = a | D(t)) \\ = \frac{[\Pr(d_a(t)|x_T = a, D(t-1)) \times P_r(x_T = a | D(t-1))]}{\Pr(d_a(t)|D(t-1))}, \end{aligned} \quad (5)$$

where the numerator term can be regarded as a detector model and the $\Pr(x_T = a | D(t-1))$ is the belief at the last moment of the a_{th} cell, which provides a recursive term for the recursive Bayesian method. By Markov's assumption of conditional independence $\Pr(d_a(t)|x_T = a, D(t-1)) = \Pr(d_a(t)|x_T = a)$, we can get

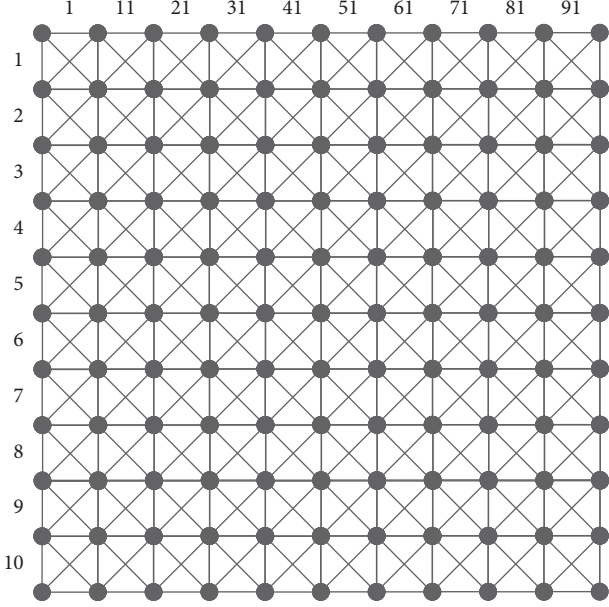
$$\begin{aligned} \Pr(x_T = a | D(t)) \\ = \frac{[\Pr(x_T = a | D(t-1)) \times \Pr(d_a(t)|x_T = a)]}{\Pr(d_a(t)|D(t-1))}, \end{aligned} \quad (6)$$

where $\Pr(d_a(t)|D(t-1))$ is the marginal distribution measured by the sensor; it can be computed by

$$\begin{aligned} P_r(d_a(t) | D(t-1)) = \sum_{H=\{0,1\}} [P_r(d_a(t)|H, D(t-1)) \\ \times P_r(H | D(t-1))]. \end{aligned} \quad (7)$$

After some algebraic manipulations, we can get the final recursive expression as follows:

$$\begin{aligned} \Pr(x_T = a | D(t)) = \frac{A}{B+C}, \\ \begin{cases} A = \Pr(d_k(t)|x_T = a, D(t-1)) \Pr(x_T = a | D(t-1)), \\ B = \Pr(d_k(t)|x_T = k, D(t-1)) \Pr(x_T = k | D(t-1)), \\ C = \Pr(d_a(t)|H = 0, D(t-1)) (1 - \Pr(x_T = k | D(t-1))). \end{cases} \end{aligned} \quad (8)$$

FIGURE 2: An example search graph G .

2.5. Decision-Making Condition. In the framework of search decision-making, \underline{B} and \overline{B} $\{0 \leq \underline{B} < \overline{B} \leq 1\}$ define the conditions for ending the search task. Once the aggregate belief $\mathbf{B}(t)$ is not within the threshold ranges, the search agent will make a decision and terminate the task. More specifically, if the condition $\mathbf{B}(t) > \overline{B}$ is satisfied, the search agent terminates the task, finds the cell with the highest confidence in the belief map, and determines that the target is in the cell. Conversely, if the condition $\mathbf{B}(t) < \underline{B}$ is satisfied, the search agent determines that the target is not in the area.

3. Search Strategy Analysis

In this section, we study two key factors that affect the success of the search task and propose two adaptive search strategies according to the characteristics of the two factors.

3.1. Decision Theory Analysis. First, the search agent collects a series of incompletely correct observations $D(t)$ after t -moment; therefore, we can quantify the process of decision evolution over time as follows:

$$\mathbf{B}(t) = 1 - \Pr(x_T = 1 | D(t)). \quad (9)$$

In order to simplify the expression, some intermediate functions are defined as follows:

$$\begin{aligned} f(d_a(t)) &= (1 - d_a(t))(1 - \alpha) + d_a(t)\alpha, \\ g(d_a(t)) &= (1 - d_a(t))\beta + d_a(t)(1 - \beta), \\ M(d_a(t)) &= g(d_a(t)) - f(d_a(t)), \\ N_a(d_a(t)) &= \begin{cases} f(d_a(t)), & \text{if } a(t) \neq a, \\ g(d_a(t)), & \text{if } a(t) = a. \end{cases} \end{aligned} \quad (10)$$

The probability that the search agent detects “1” in the current cell is represented by $f(d_a(t))$; it includes the

situation when the sensor has a false positive error. At the same time, $g(d_a(t))$ represents the searcher agent detects “0” in the current cell; it also includes the situation when the sensor has a false negative error.

Hence, equation (8) can be rewritten as

$$\Pr(x_T = a | D(t)) = \frac{A}{D \times E + F}, \quad (11)$$

where

$$\begin{cases} D = \Pr(d_a(t) | x_a(t) = 1) - \Pr(d_a(t) | x_a(t) = 0), \\ E = \Pr(x_a(t) = 1 | D(t-1)), \\ F = \Pr(d_a(t) | x_a(t) = 0), \end{cases} \quad (12)$$

$D(t) = \{d(1), \dots, d(t)\}$, so we can get a new closed expression to update the individual belief:

$$\Pr(x_T = a | D(t)) = \frac{G}{H + I \times J}, \quad (13)$$

where

$$\begin{cases} G = \prod_{k=1}^t f(d_a(k)) \Pr(x_T = 1 | D=0), \\ H = \prod_{k=1}^t f(d_a(k)), \\ I = \sum_{k=1}^t \left(\prod_{l=1}^{k-1} N_a(k)(d_a(l)) \right) M(d_a(k)) \left(\prod_{j=k+1}^t f(d_a(j)) \right), \\ J = \Pr(x_a(k) = 1 | D=0). \end{cases} \quad (14)$$

Finally, through equations (11) and (13) to calculate a one-step change in the belief map, we can find that $\mathbf{B}(t)$ is controlled by the following two factors:

- (i) $\Pr(x_a(t) = 1 | D(t-1))$ indicated the search agent should try to reach the higher belief grid at the next step
- (ii) $d_a(t)$ indicated the detected result at the next step should be positive, which also includes the false positive error

Due to the bounded speed of the robots in practical applications, the search agent cannot reach the cell with the highest belief immediately. Not only this, but also the above two conditions cannot always be compatible. Hence, according to the characteristics of these two factors, two different search strategies are proposed here. The first search strategy is called “myopic strategy” because the search agent always selects the cell with the highest confidence value in the single-step reachable cells as the location for the next detection. In the second strategy, the search agent always pays attention to the cell with the largest belief value in the map, so it needs to scan the belief distribution of the entire search region, which is called the “saccadic strategy.”

3.2. Myopic Search Strategy. Once the search agent adopts the myopic strategy, it checks the belief values of all the cells around it that can be reached in one step. In this process, the searcher tries to maximize $\mathbf{B}(t)$ in every step. Figure 3 intuitively reflects the nature of the strategy; for convenience, the unrelated cells in the belief map set as “0.” When the search agent is at (7, 4), the belief value around (8, 5) is the largest, so it will go there in the next step. Furthermore, the pseudocode when the search agent adopts the myopic strategy is given in Algorithm 1.

3.3. Saccadic Search Strategy. In the search area, the cell on the belief map with the largest belief value is critical to the searcher and should be checked as soon as possible, so the search agent needs to build the shortest path from the current location to there. In order to visualize the strategy, the belief value of the unrelated point on the belief map is set as “0” in Figure 4. Assuming the search agent at (7, 4) and the belief value at (9, 10) are the largest, so it will construct the shortest path to there by the Dijkstra algorithm. But once the peak has changed during the update process, the search agent will cancel the original plan and rebuild a new path. Similarly, the pseudocode when the search agent adopts the saccadic strategy is given in Algorithm 2. Among them, x_c represents the current location of the search agent, x_d represents the current destination that needs to go, and P is an array containing the cells that need to pass from x_c to x_d .

3.4. Repeat Detection Mechanism. In addition, in order to avoid the unreasonable behavior of the search strategy in the following, we also propose a repeated detection mechanism. The mechanism can be divided into the following three steps:

- (1) When the search agent initializes the belief map, an expected detection map is simultaneously initialized; as shown in Figure 5, the expected value is defined as “1” only in the position with the highest belief, while the others are defined as “0.”
- (2) Once the detected result detected in the cell is different from the corresponding cell in the expected detection map, the search agent will check the cell repeatedly until it is identical.
- (3) The expected detection map is the same as the belief map, which is always updated in each step. In addition, the update process for the detection map is similar to step “1.”

4. Results and Discussion

In this section, the performance parameters of the search strategy are obtained by Monte Carlo simulation. In addition, the minimum expected time to detection $\mathbb{E}[\text{TTD}]$ (includes the average simulation steps for a search task completion $\mathbb{E}[\text{TTD}]_s$, the average CPU time for a search task completion $\mathbb{E}[\text{TTD}]_c$) and the accuracy \mathbb{P} are used as the index for evaluating the performance of strategies. In the

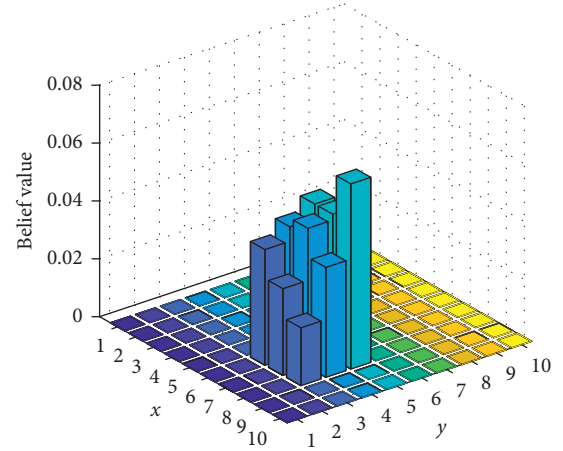


FIGURE 3: Myopic strategy.

- (1) Initialize the belief map M
- (2) Initialization decision threshold B , B and time $t=0$
- (3) Calculate the aggregate belief $B(0)$
- (4) while $B < B(t) < B$ do
- (5) Calculate the cell a_{th+1} that should be detected at the next moment according to the myopic strategy
- (6) Check the cell a_{th} and get detection result $D_a(t)$
- (7) Update M based on $D_a(t)$
- (8) Calculate $B(t)$
- (9) $t = t + 1$
- (10) end while
- (11) return Search result

ALGORITHM 1: The searching process of the myopic strategy.

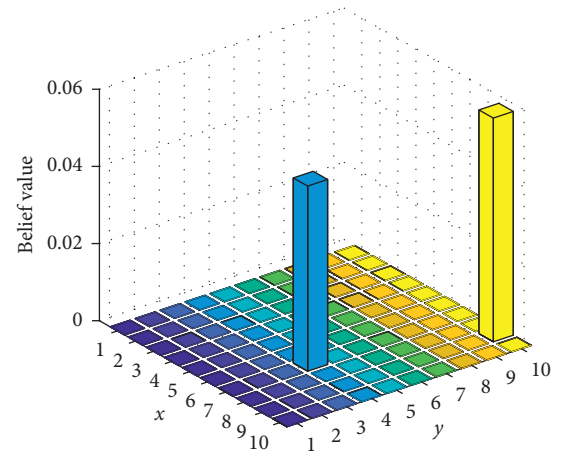


FIGURE 4: Saccadic strategy.

simulation of this section, the computer configuration and software are CPU i5 8250U, 1.6 GHz, 8 G Ram, MATLAB 2018b.

4.1. Search Environment. Consider that an object without moving ability is lost in a 10×10 block area \mathcal{A} . As shown in Figure 6, the initial probability distribution is modeled as a

```

(1) Initialize the belief map M
(2) Initialization decision threshold  $\bar{B}$ , B and time  $t=0$ 
(3) Calculate the aggregate belief  $B(0)$ 
(4) Initialization parameter  $i=1$ 
(5) while  $B < B(t) < \bar{B}$  do
(6)   Find the cell  $(x_d, y_d)$  with the largest belief on the belief map M
(7)   if  $t=0$  then
(8)     Construct a path  $P$  from  $(x_c, y_c)$  to  $(x_d, y_d)$  by the Dijkstra algorithm
(9)      $a_{th+1} = P(1)$ 
(10)  else
(11)   if The  $(x_d, y_d)$  did not change then
(12)      $a_{th+1} = P(i)$ 
(13)   else
(14)     Rebuild the path  $P$ 
(15)      $a_{th+1} = P(1)$ 
(16)     Reset  $i=1$ 
(17)   end if
(18) end if
(19)  $i=i+1$ 
(20) Check the cell  $a_{th+1}$  and get detection result  $D_a(t)$ 
(21) Update  $M$  based on  $D_a(t)$ 
(22) Calculate  $B(t)$ 
(23)  $t=t+1$ 
(24) end while
(25) return Search result

```

ALGORITHM 2: The searching process of the saccadic strategy.

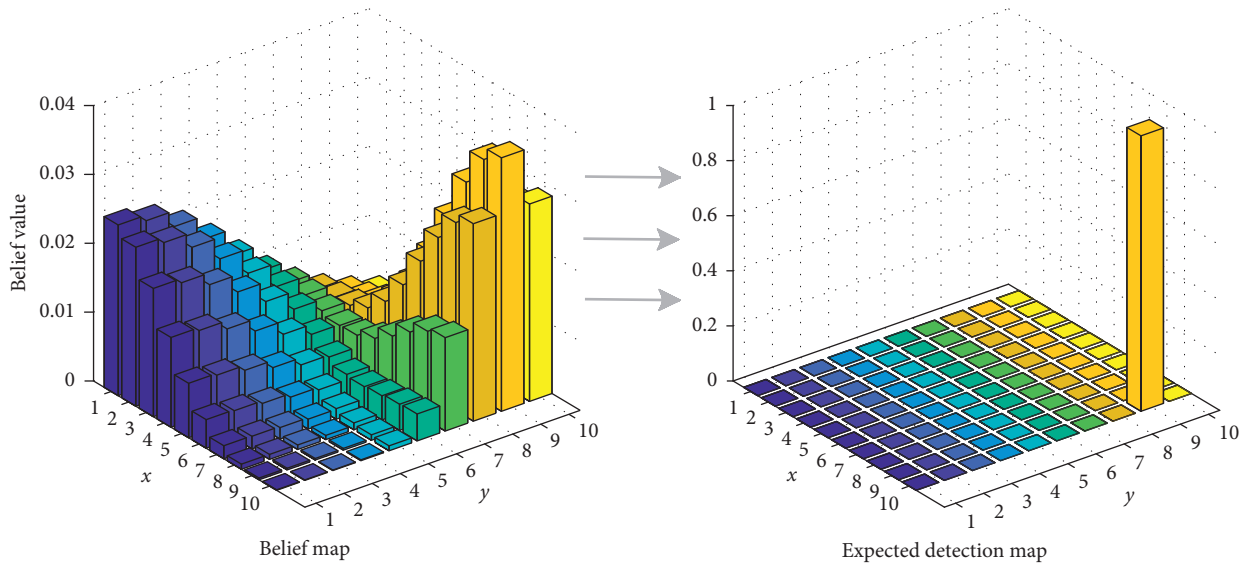


FIGURE 5: Mapping between belief map and expected detection map.

discrete approximation of Gaussian distribution, and the initial belief $B(0) = 0.5$; the search agent starts at $(1, 1)$ and, equipped with an imperfect sensor, the sensor parameters are set as false alarm probability $\alpha = 0.2$, false negative probability $\beta = 0.2$, and search decision threshold set as $\bar{B} = 0.9$.

4.2. Performance Comparison. To test the performance of the search strategies proposed in this paper, three different strategies (sweeping strategy [27], random jump strategy [20], and snapshot strategy [37]) are selected and compared.

All the strategies are tested 10,000 times; $\mathbb{E}[\text{TTD}]_s$, $\mathbb{E}[\text{TTD}]_c$, and \mathbb{P} are shown in Table 1. Not only is the myopic strategy simple to calculate, but also it has a relatively short CPU time; although the saccadic strategy has the highest accuracy, it needs strong computing power because it involves the shortest path planning in every step. In addition, Figures 7–11 show the path of the search agent under the five different search strategies, respectively. The trajectory of the random jumping strategy is too complicated, so only a part of the trajectory is shown in Figure 10.

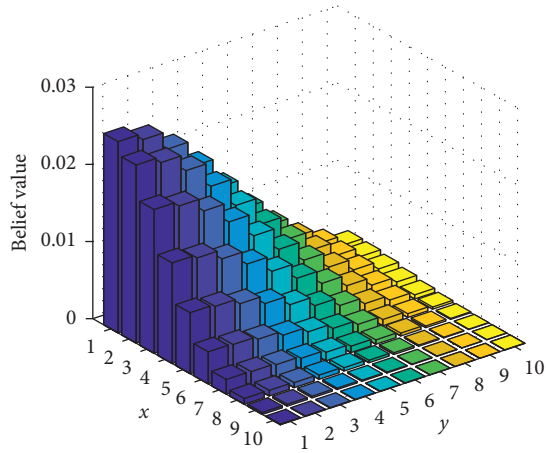


FIGURE 6: The initial belief map.

TABLE 1: Performance comparison of search strategies.

Strategy	Myopic	Saccadic	Sweeping	Random jump	Snapshot
$E[TTD_s]$	65.51	65.58	581.68	453.32	71.90
$E[TTD_c]$	54.03	527.20	23.88	167.98	469.20
P	98.01%	98.34%	87.75%	72.26%	98.21%

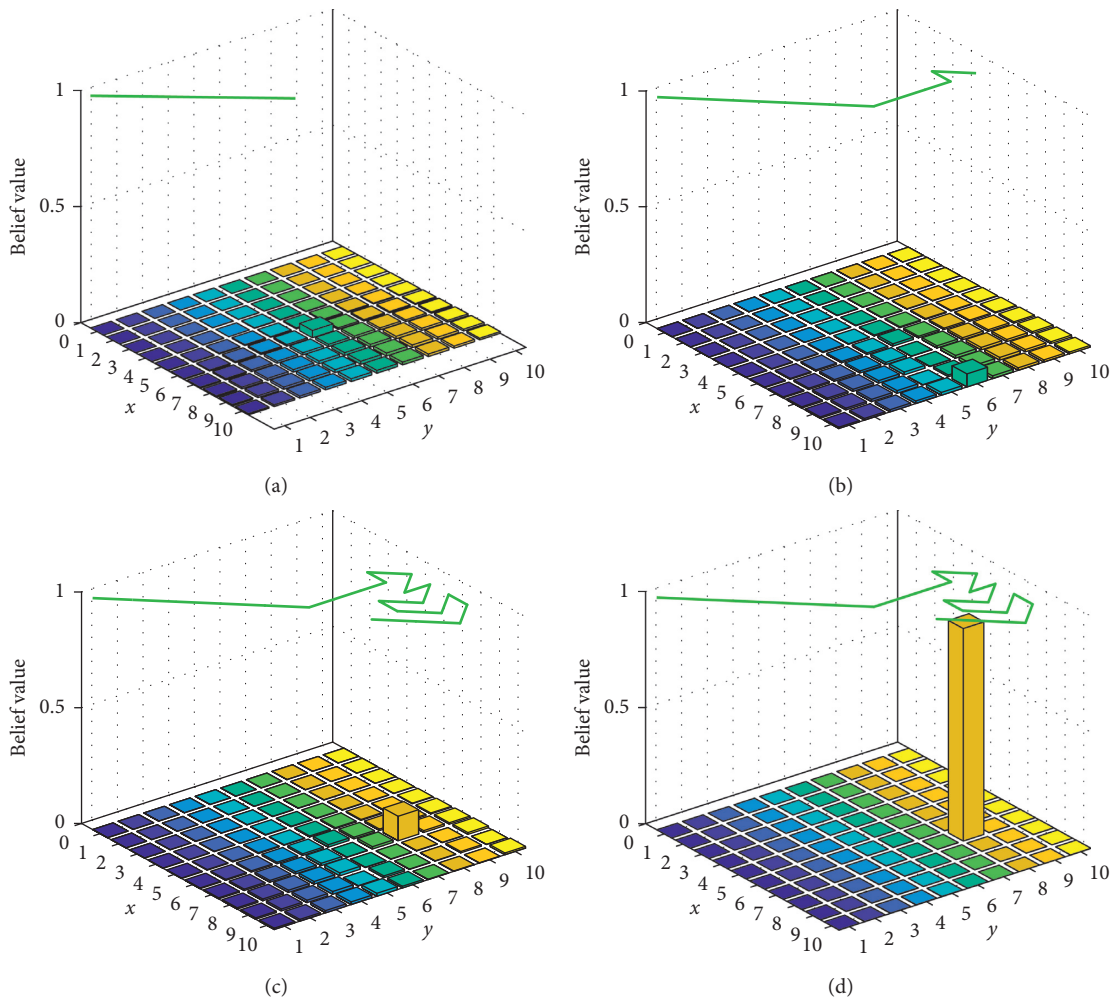


FIGURE 7: Myopic strategy: (a) step = 10, (b) step = 25. (c) step = 35, and (d) step = 66.

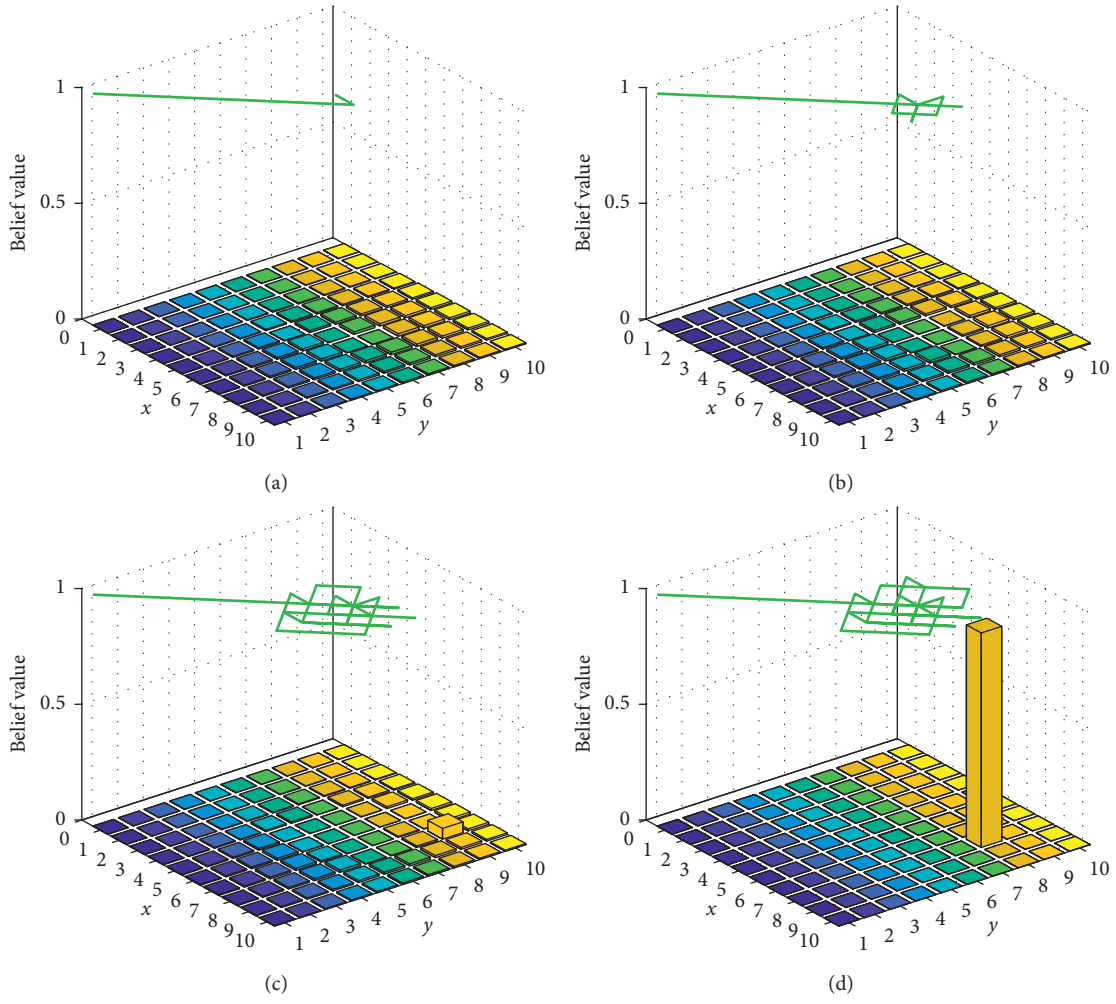


FIGURE 8: Saccadic strategy: (a) step = 15, (b) step = 25, (c) step = 45, and (d) step = 65.

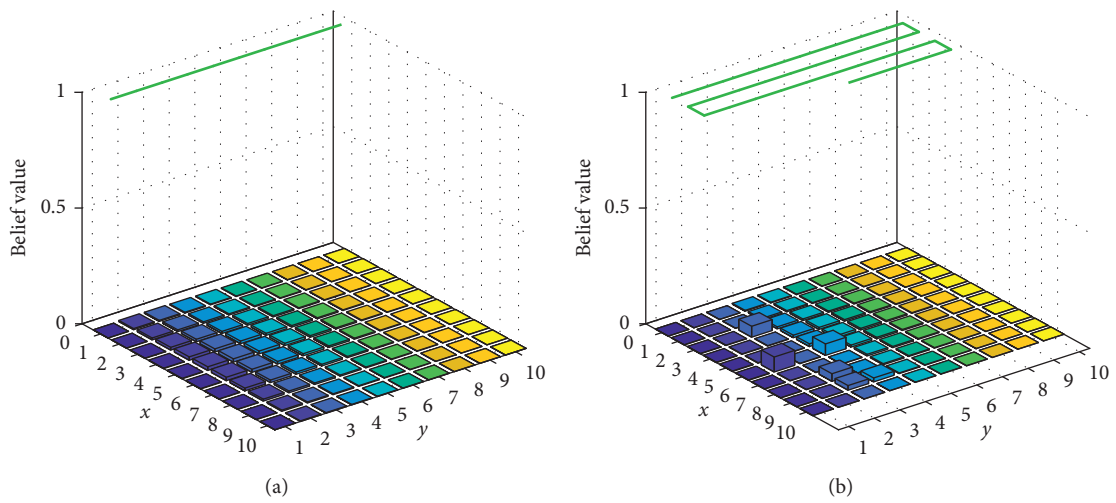


FIGURE 9: Continued.

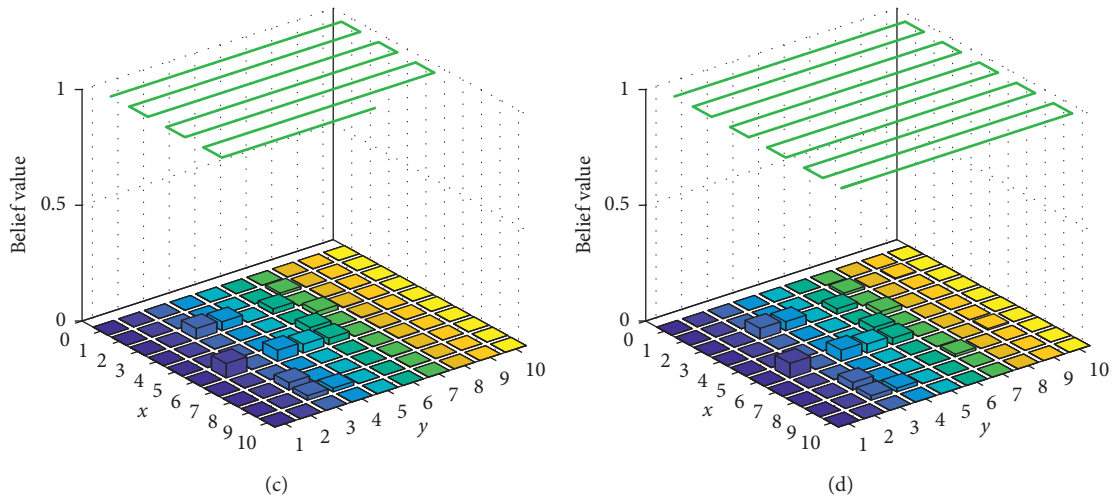


FIGURE 9: Sweeping strategy: (a) step = 10, (b) step = 36, (c) step = 68, and (d) step = 575.

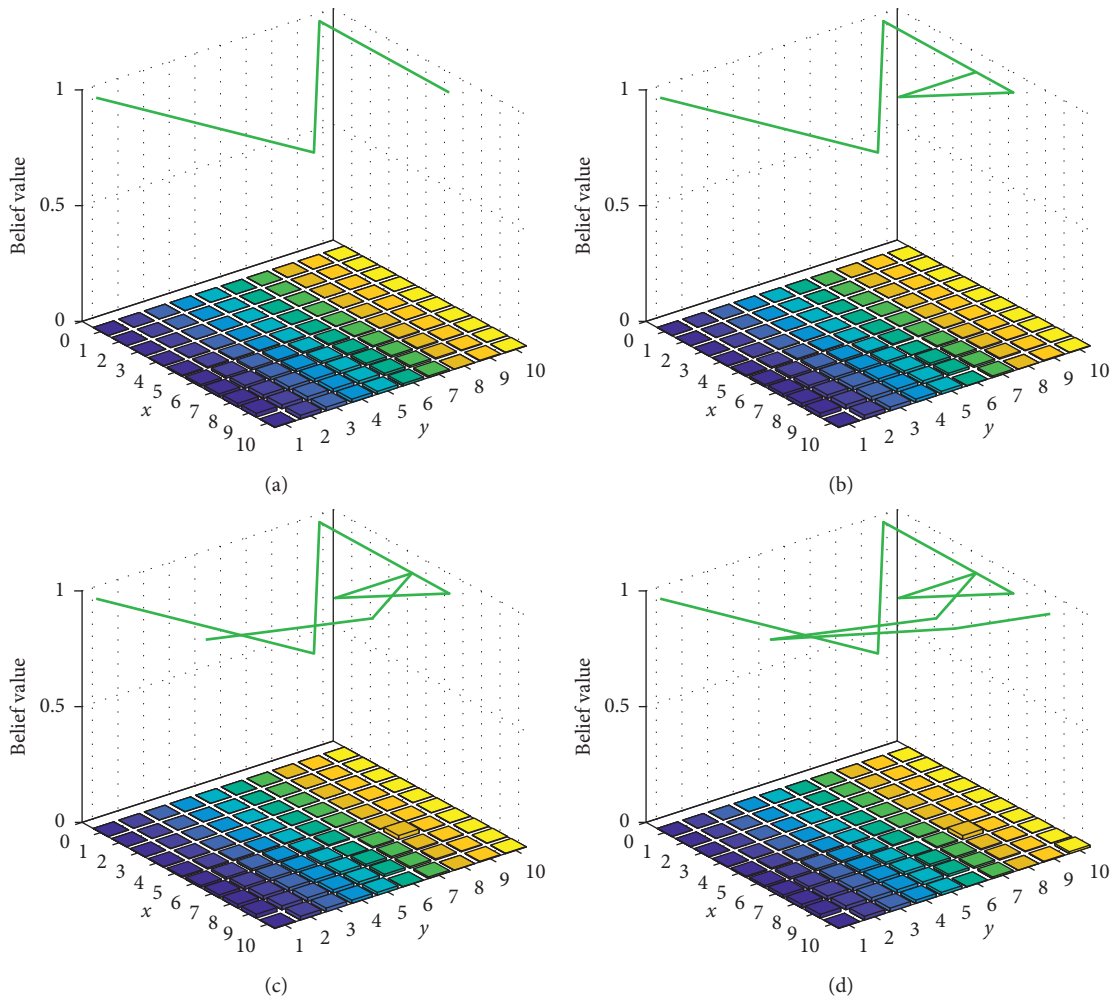


FIGURE 10: Random jump strategy: (a) step = 4, (b) step = 7, (c) step = 18, and (d) step = 32.

In addition, the aggregate belief $\mathbf{B}(t)$ evolution of the five strategies is shown in Figure 12 and the myopic strategy and the saccadic strategy can quickly reach the decision threshold (within 100 steps, which are similar to the

snapshot strategy) because they can use the Bayesian method to continuously collect new information to update the belief map, thus saving a lot of search resources. However, the sweeping strategy and the random jump strategy cannot use

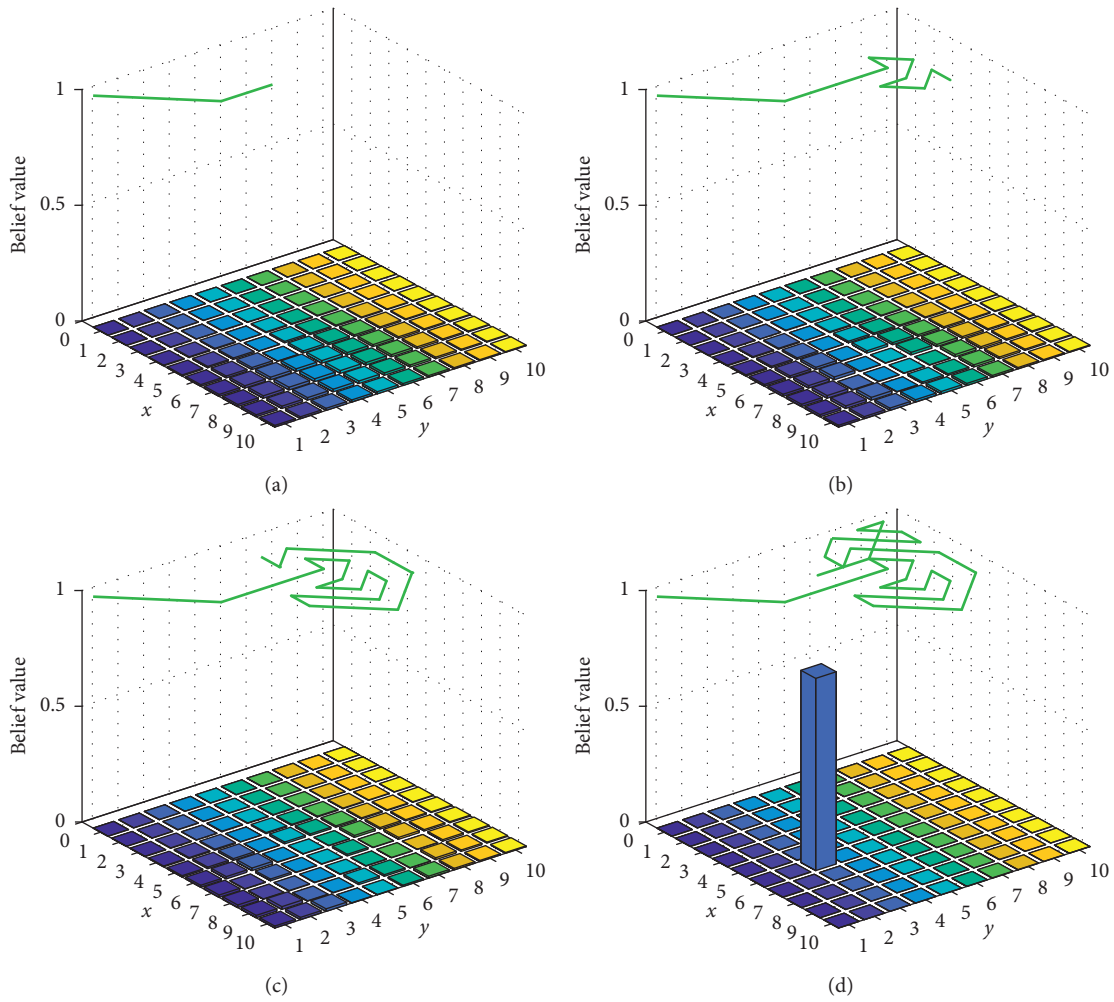


FIGURE 11: Snapshot strategy: (a) step = 7, (b) step = 18, (c) step = 32, and (d) step = 70.

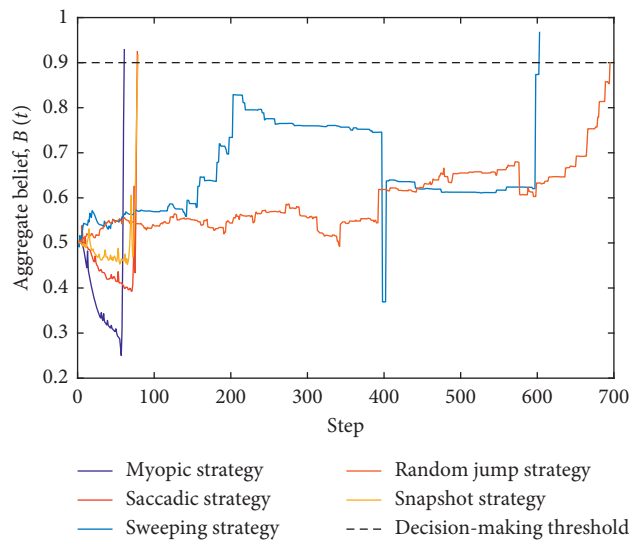


FIGURE 12: The evolution of aggregate belief.

the prior information to guide the searcher’s behavior, resulting in the fact that the search task needs more than 600 steps. Furthermore, the information entropy is used to

quantify the uncertainty of the search area. As can be seen from Figure 13, the myopic strategy and saccadic strategy can reduce the uncertainty of an unknown environment

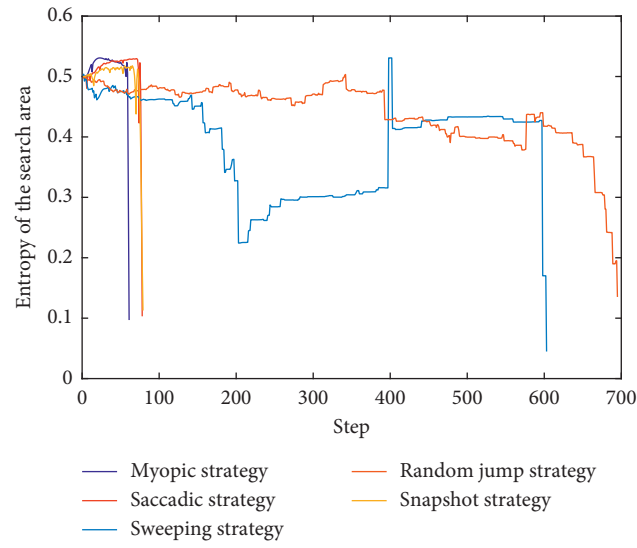


FIGURE 13: Entropy evolution of the search area.

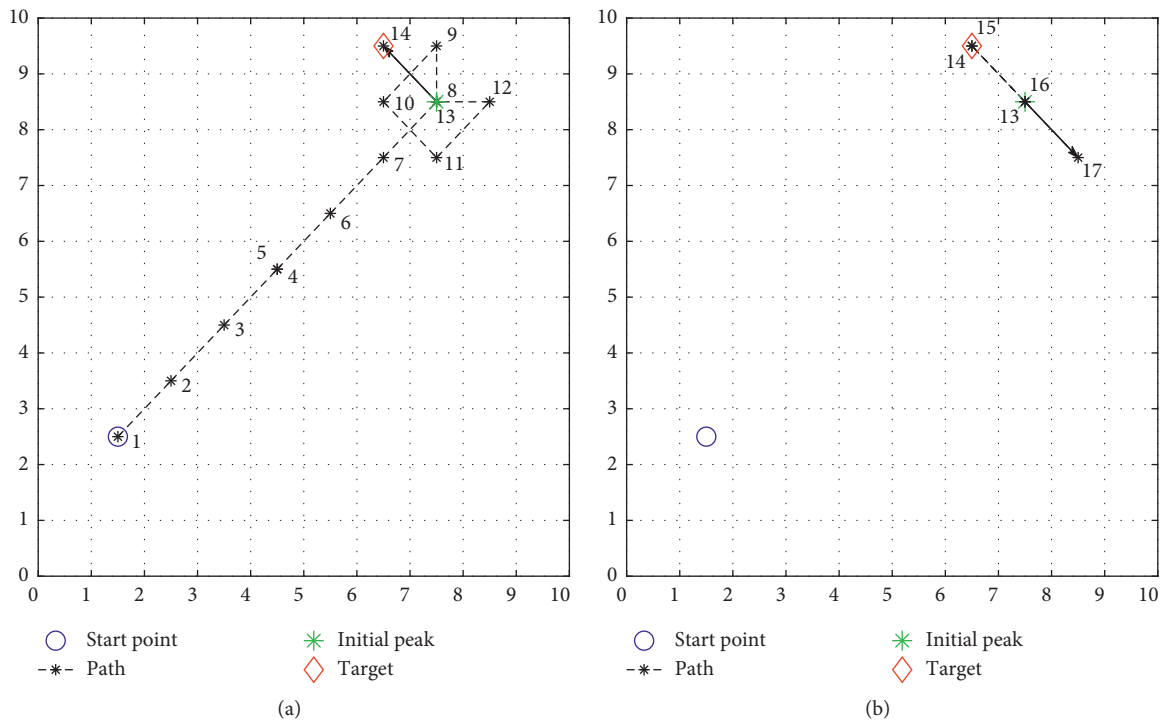


FIGURE 14: Continued.

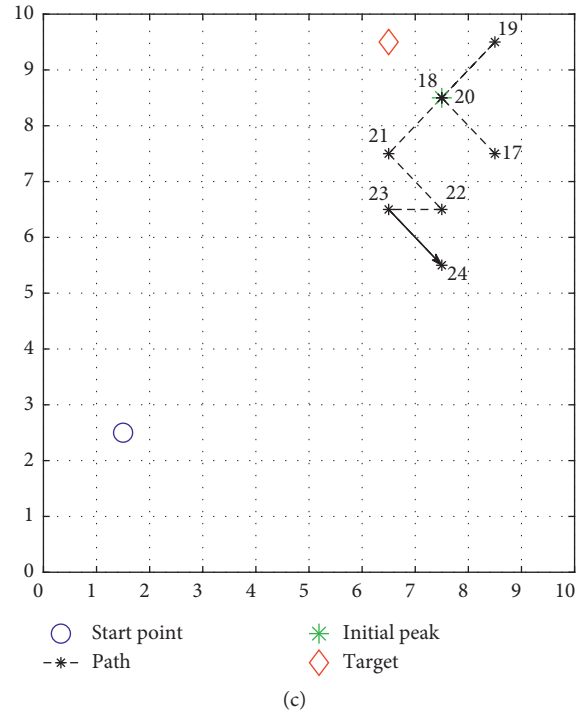


FIGURE 14: The irrational behavior in the saccadic strategy. (a) Step = 14: the search agent arrives at the target location. (b) Step = 17: at the target position, the sensor results in a false negative error. (c) Step = 24: the search agent updates the belief map with the wrong detection value and then heads to new global peak.

more quickly than the random jump strategy and sweeping strategy too. Due to the characteristics of the sweeping strategy, once the search agent misses the target, it will only reach the target position in the next traversal, which also results in the fact that its entropy is not reduced uniformly.

4.3. Search Strategy Analysis. According to the experimental data in Table 1, we find $\mathbb{E}[\text{TTD}_s]$ and \mathbb{P} of the myopic strategy and saccadic strategy are very close. Hence, we further studied how the two strategies influence the search agent.

4.3.1. Searcher with Imperfect Sensor. Through field tests, we find that the saccadic strategy has unreasonable behavior at some time, but the myopic strategy does not exist. Hence, a set of representative test data is selected to demonstrate the irrational behavior. The unreasonable behavior of the search agent is shown in Figure 14, where the number in the cell represents the order of the cells detected by the search agent.

Because of the characteristics of the myopic strategy itself, it is inherently equipped with the repeat detection mechanism. After deploying the mechanism on the saccadic strategy, the performances of the improved saccadic strategy and saccadic strategy are compared again through 10,000 experiments: $\mathbb{E}[\text{TTD}_s]_{\text{saccadic}} \approx 66$ steps, whereas the improved saccadic strategy yields $\mathbb{E}[\text{TTD}_s]_{\text{improvedsaccadic}} \approx 64$ steps. Compared with the saccadic strategy, the improved saccadic strategy has no significant performance improvement, but the phenomenon is slightly caused by α and β .

Then, we checked the effects of α and β for the decision, comparing these strategies for each set of parameters (10,000 tests per set of parameters). Relevant experimental statistics are shown in Tables 2 and 3, from which we can see that with the increase of α or β , the performance of the improved saccadic strategy has an obvious advantage.

4.3.2. Multiple Scenarios. The above experimental data shows that the myopic strategy performs so well that it can even compete with the improved saccadic strategy. In order to test the effect of the prior distribution on strategy, we have carried out a series of experiments on myopic strategy, saccadic strategy, and improved saccadic strategy under different prior belief maps.

First of all, different prior distributions are formed by varying degrees of disturbance. Figure 15 shows that heuristic information for the myopic strategy provides better robustness, as evidenced by the search agent which can easily correct the “bad initial belief map.” Furthermore, the improved saccadic strategy is also robust by introducing the repeat detection mechanism. For the myopic strategy, the search agent is not guided to a local peak when it is far from the global peak; although the saccadic strategy provides a better precision for the search agent, the downside is that the performance is heavily dependent on the initial belief distribution.

The performance of these three strategies in different situations is tested, and the relevant statistics are shown in Table 4. By comparing with other search methods and

TABLE 2: Comparison of search strategies under different false negative detection error.

(α, β)	(0.20, 0.10)	(0.20, 0.20)	(0.20, 0.30)	(0.20, 0.40)	(0.20, 0.50)
E[TTD _s]improved saccadic	42.60	60.52	94.20	140.02	200.42
E[TTD _c]improved saccadic	400.23	500.02	782.63	1000.05	1800.52
Pimproved saccadic	98.88%	98.82%	98.50%	97.88%	97.66%
E[TTD _s]myopic	43.80	65.51	98.89	146.39	238.01
E[TTD _c]myopic	48.28	54.03	65.88	86.89	138.48
Pmyopic	98.10%	98.02%	97.48%	97.10%	96.19%
E[TTD _s]saccadic	48.74	65.58	96.17	142.39	228.56
E[TTD _c]saccadic	413.33	527.20	808.84	1238.03	2019.07
Psaccadic	98.48%	98.34%	97.74%	97.66%	97.40%
E[TTD _s]sweeping	471.53	581.68	645.59	684.54	698.40
E[TTD _c]sweeping	21.48	23.88	25.65	26.19	26.60
Psweeping	94.50%	87.75%	75.34%	58.42%	40.53%
E[TTD _s]random jump	403.60	453.33	478.12	494.39	499.42
E[TTD _c]random jump	150.25	167.98	176.05	179.83	192.50
Prandom jump	84.47%	72.26%	59.69%	44.50%	31.02%
E[TTD _s]snapshot	53.03	71.90	105.50	157.42	263.17
E[TTD _c]snapshot	357.34	469.20	686.90	1048.95	1979.19
Psnapshot	98.09%	98.21%	98.13%	97.83%	97.59%

TABLE 3: Comparison of search strategies under different false positive detection error.

(α, β)	(0.10, 0.20)	(0.20, 0.20)	(0.30, 0.20)	(0.40, 0.20)	(0.50, 0.20)
E[TTD _s]improved saccadic	42.60	60.52	94.20	140.02	200.42
E[TTD _c]improved saccadic	400.23	500.02	782.63	1000.05	1800.52
Pimproved saccadic	98.88%	98.82%	97.88%	97.69%	97.66%
E[TTD _s]myopic	49.65	65.51	88.64	131.45	221.10
E[TTD _c]myopic	51.20	54.03	65.59	80.02	132.73
Pmyopic	98.05%	98.02%	97.43%	97.19%	96.65%
E[TTD _s]saccadic	50.13	65.58	92.11	130.49	217.11
E[TTD _c]saccadic	365.74	527.20	719.66	1085.39	1949.19
Psaccadic	98.52%	98.34%	97.69%	97.53%	97.15%
E[TTD _s]sweeping	401.08	581.68	670.16	698.94	699.00
E[TTD _c]sweeping	18.53	23.88	26.81	26.72	27.59
Psweeping	95.12%	87.75%	75.93%	55.77%	35.12%
E[TTD _s]random jump	362.41	453.33	478.20	498.72	499.99
E[TTD _c]random jump	142.69	167.98	185.31	196.06	200.48
Prandom jump	86.23%	72.26%	55.51%	40.09%	27.69%
E[TTD _s]snapshot	57.69	71.90	102.86	145.77	230.56
E[TTD _c]snapshot	363.62	469.20	709.80	1112.78	1941.73
Psnapshot	98.31%	98.21%	97.69%	97.67%	97.18%

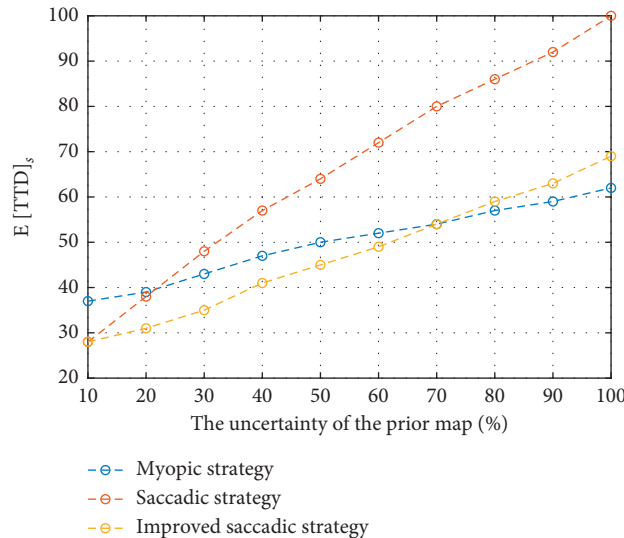


FIGURE 15: Robustness comparison.

TABLE 4: Performance comparisons in three different scenarios.

Evaluation parameters	Scenario a	Scenario b	Scenario c
E[TTD _s]myopic	60.03	61.62	61.86
E[TTD _c]myopic	68.32	69.33	69.67
Pmyopic	97.85%	96.80%	95.86%
E[TTD _s]saccadic	69.04	71.61	72.35
E[TTD _c]saccadic	413.36	502.21	603.33
Psaccadic	97.88%	96.89%	96.02%
E[TTD _s]improved saccadic	40.22	55.32	60.14
E[TTD _c]improved saccadic	330.65	405.44	500.21
Pimproved saccadic	98.10%	97.85%	97.35%

Scenario a: the search agent starts at the same location. Scenario b: the search agent starts from a local peak in a priori graph with multiple peaks. Scenario c: the search agent starts from the grid with the lowest confidence in the initial belief map.

testing in different scenarios, the statistical data shows that the proposed search decision framework and adaptive search strategies have better performance. It also can be seen from the data in Tables 2 and 3 that the detection mechanism proposed in this paper solves the unreasonable behavior caused by false alarms and false negatives of the sensor to a certain extent.

5. Conclusion and Future Work

This work studies the search problem when the sensor is incomplete and constrained by motion. At the same time, a Bayesian-based decision search framework, two adaptive search strategies, and a repeat detection mechanism are proposed. Compared with other works, the scheme proposed in this paper greatly improves the search time and the success rate of search tasks.

Future research will consider using distributed heterogeneous agents to search dynamic targets or targets with avoidance ability. In this case, information fusion is very important, such as how to fuse two or more different initial confidence graphs and how to update data between heterogeneous search agents. If we can coordinate the control of heterogeneous search agents and reasonably allocate search resources, team search will greatly improve the search efficiency.

Data Availability

All data used to support the study are included within the manuscript.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

All the authors have approved the final manuscript.

Acknowledgments

This work was supported in part by the Qinglan Project of Jiangsu Province under Grant 2018, in part by the

Cultivation Project of Xuzhou University of Technology under Grant XKY2018126, Natural Science Fund for Colleges and Universities in Jiangsu Province under no. 19KJB520016, and Jiangsu Provincial Natural Science Foundation under no. SBK2019040953.

References

- [1] J. N. McRae, C. J. Gay, B. M. Nielsen, and A. P. Hunt, "Using an unmanned aircraft system (drone) to conduct a complex high altitude search and rescue operation: a case study," *Wilderness & Environmental Medicine*, vol. 30, no. 3, pp. 287–290, 2019.
- [2] C. Xiong, Q. Li, and X. Lu, "Automated regional seismic damage assessment of buildings using an unmanned aerial vehicle and a convolutional neural network," *Automation in Construction*, vol. 109, Article ID 102994, 2020.
- [3] M. Pan, T. Linner, W. Pan, H. Cheng, and T. Bock, "Structuring the context for construction robot development through integrated scenario approach," *Automation in Construction*, vol. 114, Article ID 103174, 2020.
- [4] A. Khan, E. Yanmaz, and B. Rinner, "Information merging in multi-UAV cooperative search," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3122–3129, IEEE, Hong Kong, China, May 2014.
- [5] T. Takeda, K. Ito, and F. Matsuno, "Path generation algorithm for search and rescue robots based on insect behavior—parameter optimization for a real robot," in *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics*, pp. 270–271, Lausanne, Switzerland, October 2016.
- [6] A. Krishna Lakshmanan, R. Elara Mohan, B. Ramalingam et al., "Complete coverage path planning using reinforcement learning for tetromino based cleaning and maintenance robot," *Automation in Construction*, vol. 112, Article ID 103078, 2020.
- [7] E. C. Garrido-Merchán and D. Hernández-Lobato, "Predictive entropy search for multi-objective bayesian optimization with constraints," *Neurocomputing*, vol. 361, no. 7, pp. 50–68, 2019.
- [8] K. Asadi, A. Kalkunte Suresh, A. Ender et al., "An integrated UGV-UAV system for construction site data collection," *Automation in Construction*, vol. 112, Article ID 103068, 2020.
- [9] W. Bao, C.-A. Yuan, Y. Zhang et al., "Mutli-features prediction of protein translational modification sites," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 5, pp. 1453–1460, 2017.
- [10] W. Bao, D. Wang, and Y. Chen, "Classification of protein structure classes on flexible neutral tree," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 5, pp. 1122–1133, 2016.
- [11] B. O. Koopman, "The optimum distribution of effort," *Journal of the Operations Research Society of America*, vol. 1, no. 2, pp. 52–63, 1953.
- [12] L. D. Stone, *Theory of Optimal Search*, Academic Press, New York, NY, USA, 1976.
- [13] Y. Wang and I. I. Hussein, "Bayesian-based decision-making for object search and classification," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 6, pp. 1639–1647, 2010.
- [14] P. B. Sujit and D. Ghose, "Self assessment-based decision making for multiagent cooperative search," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 4, pp. 705–719, 2011.

- [15] A. Khan, E. Yanmaz, and B. Rinner, "Information exchange and decision making in micro aerial vehicle networks for cooperative search," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 4, pp. 335–347, 2015.
- [16] M. Shaghghi, R. S. Adve, and Z. Ding, "Multifunction cognitive radar task scheduling using Monte Carlo tree search and policy networks," *IET Radar, Sonar & Navigation*, vol. 12, no. 12, pp. 1437–1447, 2018.
- [17] W. Arthur, Y. Oh, M. Fishman, N. Kumar, and S. Tellex, "Multi-object search using object-oriented pomdps," in *Proceedings of the International Conference on Robotics and Automation*, pp. 7194–7200, IEEE, Montreal, Canada, May 2019.
- [18] Z. Zhang, J. Zhang, Z. Wei et al., "Application of tabu search-based bayesian networks in exploring related factors of liver cirrhosis complicated with hepatic encephalopathy and disease identification," *Scientific Reports*, vol. 9, no. 1, pp. 62–51, 2019.
- [19] Y. Wang, I. I. Hussein, and R. Scott Erwin, "Awareness-based decision making for search and tracking," in *Proceedings of the American Control Conference*, pp. 3169–3175, IEEE, Seattle, WA, USA, June 2008.
- [20] T. H. Chung and J. W. Burdick, "A decision-making framework for control strategies in probabilistic search," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4386–4393, IEEE, Roma, Italy, April 2007.
- [21] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, Cambridge, MA, USA, 2005.
- [22] T. H. Chung and J. W. Burdick, "Multi-agent probabilistic search in a sequential decision-theoretic framework," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 146–151, IEEE, Pasadena, CA, USA, May 2008.
- [23] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, "Optimal search for a lost target in a bayesian world," in *Field and Service Robotics*, pp. 209–222, Springer, Berlin, Germany, 2003.
- [24] M. Kress, R. Szechtman, and J. S. Jones, "Efficient employment of non-reactive sensors," *Military Operations Research*, vol. 13, no. 4, pp. 45–57, 2008.
- [25] B. Kriheli and E. Levner, "Search and detection of failed components in repairable complex systems under imperfect inspections," in *Proceedings of the Mexican International Conference on Artificial Intelligence*, pp. 399–410, Springer, San Luis Potosi, Mexico, November 2012.
- [26] T. H. Chung, "On probabilistic search decisions under searcher motion constraints," in *Algorithmic Foundation of Robotics*, pp. 501–516, Springer, Berlin, Germany, 2009.
- [27] P. Vincent and I. Rubin, "A framework and analysis for cooperative search using UAV swarms," in *Proceedings of the ACM Symposium on Applied Computing*, pp. 79–86, Association for Computing Machinery, Nicosia, Cyprus, 2004.
- [28] P. Lanillos, S. K. Gan, E. Besada-Portas, G. Pajares, and S. Sukkarieh, "Multi-UAV target search using decentralized gradient-based negotiation with expected observation," *Information Sciences*, vol. 282, pp. 92–110, 2014.
- [29] T. H. Chung and J. W. Burdick, "Analysis of search decision making using probabilistic search strategies," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 132–144, 2011.
- [30] E. Teruel, R. Aragues, and G. López-Nicolás, "A distributed robot swarm control for dynamic region coverage," *Robotics and Autonomous Systems*, vol. 119, pp. 51–63, 2019.
- [31] I. Wegener, "Optimal search with positive switch cost is np-hard," *Information Processing Letters*, vol. 21, no. 1, pp. 49–52, 1985.
- [32] B. Mohamed and M. Abd Allah El-Hadidy, "Parabolic spiral search plan for a randomly located target in the plane," *ISRN Mathematical Analysis*, vol. 2013, Article ID 151598, 8 pages, 2013.
- [33] H. M. Abou Gabal and M. A. Allah El Hadidy, "Optimal searching for a randomly located target in a bounded known region," *International Journal of Computing Science and Mathematics*, vol. 6, no. 4, pp. 392–403, 2015.
- [34] H. Wang, W. Mao, and L. Eriksson, "A three-dimensional dijkstra's algorithm for multi-objective ship voyage optimization," *Ocean Engineering*, vol. 186, pp. 106–131, 2019.
- [35] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [36] Z. Tahir, A. H. Qureshi, Y. Ayaz, and R. Nawaz, "Potentially guided bidirectionalized RRT* for fast optimal path planning in cluttered environments," *Robotics and Autonomous Systems*, vol. 108, pp. 13–27, 2018.
- [37] A. A. Robie, *Multimodal sensory control of exploration by walking drosophila melanogaster*, Ph.D. thesis, California Institute of Technology, Pasadena, CA, USA, 2010.