



## Research Article

# A Physarum-Inspired Decision-Making Strategy for Multisource Task Searching of Mobile Robots

Laihao Jiang,<sup>1</sup> Hongwei Mo<sup>1</sup> , and Lifang Xu<sup>1</sup> <sup>2</sup>

<sup>1</sup>College of Automation, Harbin Engineering University, Harbin 150001, China

<sup>2</sup>Engineering Training Centre, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Hongwei Mo; honwei2004@126.com and Lifang Xu; mxlfang@163.com

Received 24 June 2020; Revised 17 August 2020; Accepted 27 August 2020; Published 21 September 2020

Academic Editor: Zhile Yang

Copyright © 2020 Laihao Jiang et al. 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 the real world, there are many different kinds of sources, such as light, sound, and gas, distributed randomly over an area. Source search can be carried out by robotic system in applications. However, for a single robot, the multisource search has been receiving relatively little attention compared to single-source search. For multisource task searching, a single robot has a high travel cost and is easy to trap a source which has been located before. In order to overcome these shortages, two multisource search algorithms inspired by the foraging behavior of *Physarum polycephalum* are proposed in this paper. First, a Physarum-inspired Strategy (PS) is designed based on the gradient climbing characteristic of *Physarum polycephalum* during foraging. The PS is simple and effective to let a mobile robot traverse all sources. Then, an extension algorithm named Physarum-inspired Decision-making Strategy (PDS) is proposed based on PS. Therein the synthetical field gradient model is established by introducing decision-making factor to obtain more accurate gradient information estimation. The PDS also introduces an obstacle avoidance model. Various simulation results obtained in the multisource environments show that the performance of PDS is better than other algorithms.

## 1. Introduction

Source search is one of the most popular behaviors in the biological world, such as foraging, mate-finding, and cell migration. Strategies behind these behaviors have encouraged the development of mobile robots that can search various sources. Many applications can be found in locating chemical sources leakage [1–3], searching and rescuing targets [4, 5], and searching for a gas/odor source [6–8].

Using robotic system to search for a single source has been broadly researched. However, in the real world, multiple sources are often distributed over an area and need to be discovered [9, 10]. We may consider a realistic scenario, such as survivor rescue after an earthquake. The people waiting to be rescued are scattered in different areas, and we can use many autonomous mobile robots equipped with life signal detectors to find all people as soon as possible. In general, multisource search task has distinct challenges for robots [9]: (1) how to continue after each source is found;

(2) how to avoid obstacles; and (3) how to find all sources in minimal time.

The methods for source search task can be divided into two categories: single-robot based methods and multirobot based methods [11]. Generally, most of the source search problems have a common characteristic that no matter what kind of source is, the closer the source is, the stronger source signal is. Thus, the robot can perceive the gradient information of the surroundings to search for the potential source, and a number of gradient-based source search methods for a single robot have been published. The gradient-based source search strategies can be classified into two categories: chemotaxis and biased random walk. The simplest chemotaxis-based algorithm was proposed for a search task [1]. By comparing concentration difference between two chemical sensors or different positions, the robot moves to the direction of higher intensity value at a constant speed. The limitation of this model is that fewer sensors will decrease the accuracy of the chosen direction. A

chemotaxis-based source search algorithm takes the wind direction as an important factor [2]. The motion vector of robot is a linear combination of the concentration gradient direction and wind direction. Similar strategies were also presented in [6, 12]. Although these methods have achieved in single-source search, they have not been extended to solve multi-source search. Biased random walk is also called bacterial chemotaxis, which is inspired by the foraging behavior of bacteria on the liquid surface. There is evidence that the gradient-induced search strategy inspired by bacterial chemotaxis can guide a robot to locate the sources [13]. In this approach, the robot makes a decision on either moving straight or turning a random angle by comparing current measured concentrations with previous ones. For the environment that is not fully covered by the source signals, a Yuragi-Based adaptive biased random walk strategy has been proposed to guide a mobile robot to search for a sound source [14]. This method improves success rate and efficiency of source search by combining Bacterial Chemotaxis and Levy walk. Another related study has investigated a bacteria-inspired mobile robot with I-DOF and a single sensor, revealing the relation between the navigation behavior and environmental noise [15]. However, the related algorithms based on biased random walk cannot guarantee the repeatability of path.

Nowadays, multi-robot based algorithms have been used in the multi-source search tasks. An agent-based attraction-rejection model has been proposed for searching for multiple sources [16]. However, the model is centralized, which means that the robots know relative positions to each other in real time. A novel 3D pseudolinear Kalman filter (PLKF) and gradient-descent path optimization algorithm are proposed for angle-of-arrival (AOA) target tracking using multiple own-ships [17]. As an extension study, a distributed 3D angle-of-arrival (AOA) target searching and tracking method in three-dimensional (3D) space has been proposed for multiple UAVs [18]. This method consists of a distributed estimator and path optimization algorithm. Particle Swarm Optimization (PSO) as a distributed algorithm has been used in robotic multi-source search task. The related algorithms inspired by PSO consider each robot as a particle, and the communication and cooperation between robots are similar to that of particles in PSO [19–23]. The MPSO is developed that chemotaxis enables the robots to follow a local gradient of the chemical concentration, while anemotaxis-driven PSO measures the direction of wind and guild robot to locate the odor source [19]. The subgroup-based PSO algorithms were proposed to divide swarm robots into subgroups, in which there is a main robot [20, 21]. The main robot will attract the members within its team and reject other main robots to ensure different subgroups search for different sources. The Adaptive Robotic PSO for target search has been introduced, considering the obstacle avoidance and local optimal escape [22]. The inertia weight of each robot is adaptively adjusted by evolutionary speed factor and aggregation degree at each iteration. A modified guaranteed convergence particle swarm optimizer (MGCPSO) was presented [23], in which the low fitness robot will give up the role in the current group and to assist the globally best robot. This algorithm is

superior to other algorithms in terms of success rate and iteration time. As the number of robots increases, the multi-robot algorithms have the following problems, such as collision, high load of the communication network, and increased computational complexity. Researchers should make effort to find a better way of completing the multi-source search task by fewer robots. In addition, there are some cases where the number of sources is much larger than that of robots. In these cases, each robot in swarm robotic system gets desire to maximize the chance of finding at least one source, the chance of finding as many sources as possible within a limited time. Therefore, it is essential to have an effective multi-source search strategy for a single robot, in which the robot plans its movement whilst considering environment status and obstacle avoidance.

This paper deals with the multi-source search using a mobile robot, considering two aspects: First, robot avoids getting trapped around a source and traverses all the sources. Second, the robot balances the gradient information of each source and makes an autonomous decision action during the search process, which results in reducing the probability of generating the redundant path. The proposed algorithms are inspired by the foraging behavior of *Physarum polycephalum* in this paper. *Physarum polycephalus* has been shown to rely on reactive navigation to explore the environment and search for multiple food. A number of foraging models have been proposed [24–29]. Related researches illustrate that *Physarum* has an ability to solve maze navigation [30] and graph theory problems [31–34].

The contributions of this work are as follows: (1) we propose a novel multi-source search strategy named PS by modifying the growth rule of the *Physarum* foraging model in [28]; and (2) an extension algorithm named PDS is developed, considering the decision-making of each source and obstacle avoidance. In simulations, the PDS has the best performance compared to other three algorithms: Sweep, Yuragi-based strategy, and PS.

The rest of this paper is organized as follows. Section 2 describes problem description. Section 3 presents the PS and PDS, respectively. The simulation results and discussions are presented in Section 4. Finally, Section 5 concludes this paper and discusses future work.

## 2. Problem Description

In this paper, the shape of each source is represented as an ellipse. The center of ellipse has the maximum field intensity. Each source diffuses particles from the center to the surroundings layer by layer. The synthetical field intensity at one point is inversely proportional to the distance from the sources. This paper establishes a Four-tuple  $\langle R, E, S, L \rangle$  to describe the multi-source search task for a robot:

- (1) where  $R = \langle f, \delta, r \rangle$  represent the properties of robot: where  $f$  is the field intensity perceived by robot at the current position.  $\delta$  is the speed of robot, and  $r(x, y)$  is the current coordinate of robot. All the parameters in  $R$  are known to robot.
- (2) where  $E \subset R^2$  denote a closed environment.

- (3) where  $S = \{s_1, s_2, \dots, s_m\}$  denote the set of sources. All the elements in  $S$  are unknown parameters to robot.  $m$  is the number of sources.
- (4) where  $L = \langle C, O, \lambda \rangle$  represent the relevant source properties: where  $C = \{c_1, c_2, \dots, c_m\}$  is the size information set of sources. For  $c_i \in C$ ,  $c_i = (a_i, b_i)$ .  $a_i$  and  $b_i$  are the semimajor axis and semiminor axis, respectively. The size of source represents the quantity of substances in the source.  $O = \{o_1, o_2, \dots, o_m\}$  denote the center position set of sources. Let  $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$ ,  $\forall \lambda_i \in \lambda$ ,  $\lambda_i > 0$ , denote the central field intensity set of sources. The central field intensity represents the quality of source and determines the degree of attraction to robot. All the parameters in  $L$  are unknown to robot.

The synthetical field intensity is obtained by equations (1)–(3) [28]. If there is only one source in the environment  $E$ , the field intensity at the point  $n (x, y)$  is defined as follows:

$$f(n) = \lambda_i e^{-d_{no}}, \quad (1)$$

where  $\lambda_i$  is the central field intensity of  $s_i$ , and  $e$  is the natural logarithm.  $d_{no}$  can be calculated based on equation (2) to denote distance information between the center of  $s_i$  and the point  $n (x, y)$ , where  $a$  and  $b$  are the semimajor axis and semiminor axis of  $s_i$ , respectively.  $o_i (x_i, y_i)$  represents the center coordinate of  $s_i$ .

$$d_{no} = \sqrt{\left(\frac{x - x_i}{a}\right)^2 + \left(\frac{y - y_i}{b}\right)^2}. \quad (2)$$

When multiple sources are distributed in the environment, concentration field is the superposition of them. The synthetical field intensity at the  $n (x, y)$  is calculated by equation (3), where  $m$  is the number of sources:

$$f_s(n) = \sum_{i=1}^m \lambda_i e^{-d_{no}}. \quad (3)$$

### 3. Methods

In this section, two Physarum-inspired search strategies, PS and PDS, are proposed. The former modifies the growth rules of Physarum foraging model in [28], mainly serving as a benchmark. The robot is equipped with corresponding sensors according to the source types. During the process of searching, the robot perceives the concentration information of the current position in real time. For the latter, we introduce the decision-making factor of each source and obstacle avoidance model.

**3.1. The Steps of the PS.** This part describes the steps of the PS. The dynamic process of source search based on PS is shown in Algorithm 1, where the initial coordinate of robot  $r_{ini} (x_{ini}, y_{ini})$  is necessary to activate the model. The robot only determines the next direction of movement according to the field gradient of the current position. On initialization, we set maximum iterations  $I_{max}$  and  $o_i (x_i, y_i)$ . The synthetical

field gradient  $G_{fs}(n)$  and the normalized vector  $N_{fs}(n)$  at the point  $n (x, y)$  can be given by equations (4) and (5), respectively (line 4–5):

$$G_s(n) = \sum_{i=1}^m \left( \frac{\lambda_i e^{-d_{no}} (x_i - x)}{d_{no} a_i^2}, \frac{\lambda_i e^{-d_{no}} (y_i - y)}{d_{no} b_i^2} \right), \quad (4)$$

$$N_{fs}(n) = \frac{G_{fs}(n)}{\|G_{fs}(n)\|}. \quad (5)$$

The robot moves along the synthetical field gradient. The position of robot at the next iteration is updated by equation (6) (line 6):

$$r(t+1) = r(t) + \delta N_{fs}(n), \quad (6)$$

where  $\delta$  denotes the speed of robot and is dimensionless.

If the condition meets the  $\|r - o_i\| \leq \min\{a_i, b_i\}$ , it means that  $s_i$  is accessed (line 8). The robot will mark the current position as a new start point for searching for the next source. Ignoring the exploitation time at  $s_i$ , the robot immediately enters the exploration stage to search for other sources. At the same time, we set the central field intensity value of  $s_i$  to 0 and the  $S$  eliminate  $s_i$  (line 9). It results in updating the synthetical field intensity distribution. The robot relies on the updated synthetical field intensity distribution to search for the next source autonomously, which prevents the repeated search for the same source. If the robot cannot perceive any field intensity ( $S = \emptyset$ ) or  $t > I_{max}$ , the algorithm will terminate (line 11). Finally, a trajectory is obtained while the robot traverses all sources within a limited time or the task fails. The trajectory generated by the robot is a curve containing a start point and an end point.

**3.2. The Steps of the PDS.** In PS, this reactive search strategy lacks decision-making for each source and obstacle avoidance in the search process. During the food search process, Physarum exhibits two distinct capabilities: decision-making on food and repellent stimuli avoidance. Inspired by these foraging behavior characters of Physarum, the PDS based on PS is proposed. In the PDS, we remodel the synthetical field gradient by introducing decision-making factor for each source to improve the accuracy of gradient estimation and establish obstacle field gradient model to integrate into the synthetical field gradient.

At first, the original gradient component for each source at point  $n (x, y)$  is calculated by

$$g_{f_i}(n) = \left( \frac{\lambda_i e^{-d_{no}} (x_i - x)}{d_{no} a_i^2}, \frac{\lambda_i e^{-d_{no}} (y_i - y)}{d_{no} b_i^2} \right). \quad (7)$$

We defined that the field gradient for each source of the robot perception is the original gradient multiplied by its own decision-making factor. Therefore, the field gradient for all sources is calculated by

$$G_{fs}^p(n) = \sum_{i=1}^m p_i g_{f_i}(n), \quad (8)$$

```

(1) Initialize input a Four-tuple  $\langle R, E, S, L \rangle$ ,  $r_{\text{ini}}$  ( $x_{\text{ini}}, y_{\text{ini}}$ ),  $I_{\max}$  and  $t=0$ 
(2) For  $t = 1: I_{\max}$ 
(3) Calculate  $f_s(n)$  based on equation (3)
(4) Calculate  $G_s(n)$  based on equation (4)
(5) Normalized calculation  $N_s(n)$  based on equation (5)
(6) Choose the new position of the mobile robot  $r(t+1)$  based on equation (6)
(7) And  $t = t + 1$ 
(8) If  $\|r - o_i\| \leq \min\{a_i, b_i\}$  then
    S eliminate the  $s_i$ 
(9) End if
(10) If  $S = \emptyset \text{ and } I \geq I_{\max}$ 
    Break
(11) End if
(12) End for

```

ALGORITHM 1: The PS-based source search.

where function  $m$  is the number of sources.  $p_i$  is the decision-making factor for each source and endows the robot with the ability to balance the quality information of each source. In other words, during the robot movements without sensing obstacles, the robot will consider the proportion of field gradient of each source in synthetical field gradient and decide to choose the most appropriate direction of next move.  $p_i$  is defined as follows:

$$p_i = \frac{e^{\|g_{f_i}(n)\|}}{\sum_{i=1}^m e^{\|g_{f_i}(n)\|}}. \quad (9)$$

To avoid obstacles, an obstacle field gradient model is given by

$$g_{f_j}^p(n) = \begin{cases} p_j g_{f_j}(n), & \rho \leq \gamma, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

$o_j(x_j, y_j)$  is the center coordinate of obstacle  $j$ . As shown in Figure 1, in order to not change the movement trend of the robot, the influence range of obstacle is local.  $\rho$  is the shortest distance from robot to the obstacle outline.  $\gamma$  is the redundancy parameter, which denotes the influence range of obstacle.  $\lambda_j$  is the central field intensity of obstacle and is a negative value.  $p_j$  is decision-making factor for the obstacle  $j$ , which is calculated by

$$p_j = \begin{cases} \frac{e^{\|g_{f_j}(n)\|}}{\sum_{i=1}^m e^{\|g_{f_i}(n)\|} + e^{\|g_{f_j}(n)\|}}, & \rho \leq \gamma, \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

where  $\|g_{f_i}(n)\|$  and  $\|g_{f_j}(n)\|$  is modulus of original field gradient component of the source  $i$  and obstacle  $j$ , respectively. Finally, we remodel the synthetical field gradient as follows:

$$G_{f_{sj}}^p(n) = G_{f_s}^p(n) + g_{f_j}^p(n). \quad (12)$$

In Figure 2, we show the flowchart of source search based on PDS, and the details are described in Algorithm 2. In each

iteration, the robot performs source search and obstacle avoidance based only on the gradient of the current position (line 4-line 10). In the applications, once the robot enters the influence range of obstacle, the robot with distance measuring sensors (laser-ranging-sensor or camera, etc.) converts the distance information into the gradient information and integrate it into the synthetical field gradient. The robot can rely on the synthetical field gradient to move until the termination condition is triggered.

**3.3. Local Optima Escape.** When there are multiple sources in the environment, local optima points may be distributed in the map. Local optima points can be caused by a combination of sources. The original synthetical field gradient  $G_{fs}(n)$  obtained by the robot in the local optima point is  $(0, 0)$ . Therefore, to obtain a valid successful path and ensure that all sources are accessed, it is necessary to provide an escape strategy. In this situation with local optima points, we introduce a fluctuating noise  $\varepsilon$  to the robot which is similar to the biological fluctuation that exists in organisms. When the robot falls in the local optima point,  $\varepsilon$  becomes dominant and the robot transforms into a “stochastic state.” In stochastic state, the robot changes the direction randomly. We can construct kinematic equation of robot to describe the escape strategy which is as follows:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\alpha}(t) \end{bmatrix} = \begin{bmatrix} \delta \cos(\theta + \dot{\alpha}(t)) \\ \delta \sin(\theta + \dot{\alpha}(t)) \\ \omega(t) \end{bmatrix}, \quad (13)$$

$$\omega(t) = 2\pi * \varepsilon,$$

where  $\dot{\alpha}(t)$  represents the amplitude of change in the direction of movement.  $\theta$  is the movement direction of a robot when it is trapped in a local optimal position.  $\varepsilon$  affects the rotation of robot angular velocity and is set as zero-mean Gaussian noise. The triggering condition of escape strategy is that the robot can perceive the synthetical field intensity but the original synthetical field gradient is  $(0, 0)$ .

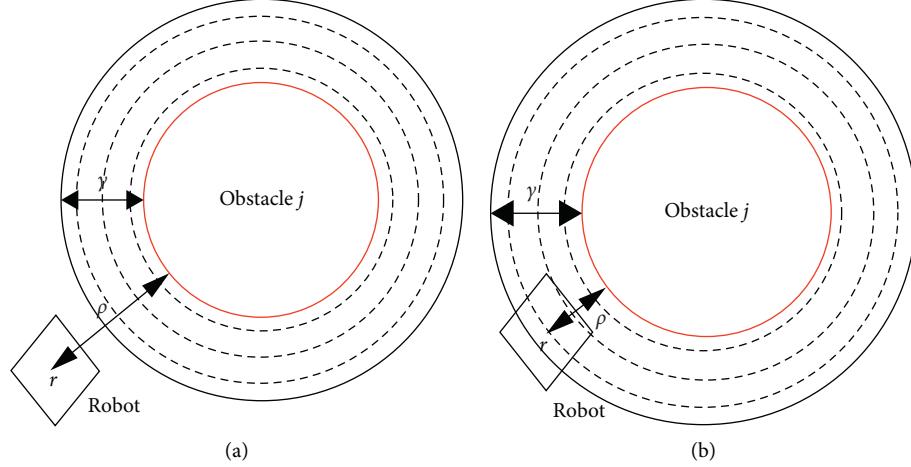


FIGURE 1: The autonomous movement of the robot in environment with obstacles. Red, black, and dotted circles represent obstacles, the boundary of the influence range of obstacle, and obstacle field distribution. (a) The robot is outside influence range of obstacle. (b) The robot is inside influence range of obstacle.

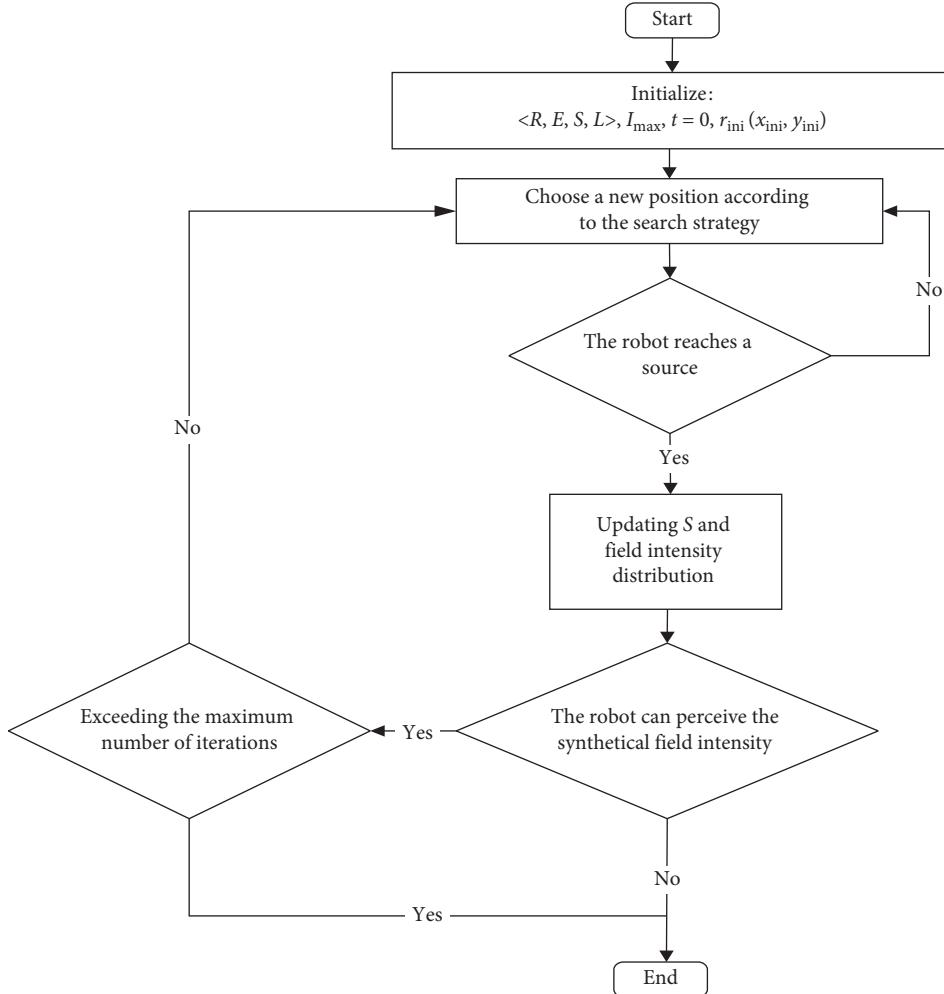


FIGURE 2: The flowchart of source search based on PDS.

```

(1) Initialize input a Four-tuple  $\langle R, E, S, L \rangle$ ,  $r_{\text{ini}}$  ( $x_{\text{ini}}, y_{\text{ini}}$ ),  $I_{\max}$  and  $t = 0$ 
(2) For  $t = 1: I_{\max}$ 
(3) Calculate  $f_s(n)$  based on equation (3)
(4) Calculate  $g_{f_i}(n)$  based on equation (7)
(5) Calculate  $p_i$  based on equation (9)
(6) Calculate  $p_j$  based on equation (11)
(7) Calculate  $G_{f_s}^P(n)$  based on equation (8)
(8) Calculate  $g_{f_j}^P(n)$  based on equation (10)
(9) Calculate  $G_{f_{sj}}^P(n)$  based on equation (12)
(10) Choose the new position of the mobile robot for next move  $r(t+1) = r(t) + \delta((G_{f_{sj}}^P(n)) / (\|G_{f_{sj}}^P(n)\|))$ 
(11) And  $t = t + 1$ 
(12) If  $\|r - o_i\| \leq \min\{a_i, b_i\}$  then
(13) S eliminate the  $s_i$ 
(14) End if
(15) If  $S = \varnothing \& I \geq I_{\max}$ 
(16) Break
(17) End if
(18) End for

```

ALGORITHM 2: The PDS-based source search.

#### 4. Simulation Results and Analysis

To verify effectiveness of the proposed algorithms, we present several simulations with respect to source types, different center field intensities of source, and environments with obstacles. These simulations are presented in four parts. Part 1 studies the impact of the different source models by comparing the trajectory of PDS with PS. The source properties are shown in Table 1. Part 2 studies the performance comparison of PS and PDS under different central field intensities. Part 3 studies several simulations in environments with obstacles. Part 4 studies comparison of statistical efficiency value for PDS and other algorithms. Related parameters, we set  $I_{\max}$  to 2000 and  $\delta$  to 0.1. Each source is programmed as an ellipse. This paper does not consider the real physical shape of the robot. The robot is represented by a diamond just to make the simulation effect more intuitive. The environments are unknown and have a size of  $20 * 20$ . We consider the terrain flat, without considering slopes, holes, or any other external factors.

**4.1. The Comparisons under Different Source Models.** In this part, it will be shown that how a robot accomplishes source search task based on PS and PDS under different source models. In Figure 3(a), the field intensity distribution of source is Gaussian-like shape. The field intensity at the center of the source is the maximum and decreases non-linearly outward from the source center. In Figure 3(b), there are six sources in the environment, the synthetical field intensity model based on equation (3), wherein we set all sources to have the same central field intensity  $\lambda_i = 1, \forall i \in \{1, 2, \dots, 6\}$ . The field intensity value is  $1.5 > 1$  at some points, because the synthetical field intensity distribution is the superposition of the field intensity distribution of each source. In Figures 3(a) and 3(b), the number and locations of sources can be directly observed.

TABLE 1: The source properties.

Source	$x$	$y$	$a$	$b$	$\lambda$
$s_1$	1	11	1	1	1
$s_2$	6	2	1	1	1
$s_3$	8	5	1	1	1
$s_4$	12	12	1	1	1
$s_5$	4	18	1	1	1
$s_6$	6	4	1	1	1

For the multi-source with Gaussian-like shape field intensity distribution, a comparison is made between PS and PDS. The process of searching for sources based on PDS is shown in Figures 4(a)–4(g). In Figure 4(a),  $r_{\text{ini}}(1, 1)$  is the initial position of the robot. The robot accesses the  $s_2$  at  $t = 43$ . Ignoring the exploitation time, the robot starts to search for the next source as shown in Figure 4(b). During the search process of  $s_2 \rightarrow s_6$ ,  $s_2$  will not contribute field intensity to any point in the environment. This strategy will increase the probability that the robot accesses the remaining sources and prevent the robot from trapping a source. Figures 4(c)–4(g) are the segments of robot movement process from  $t = 52$  to  $t = 333$ . Finally, the robot generates a trajectory that traverses all sources. The access sequence is  $s_2 \rightarrow s_6 \rightarrow s_3 \rightarrow s_1 \rightarrow s_4 \rightarrow s_5$ . Figure 4(h) represents the trajectory comparison between PS and PDS. The trajectories of  $r_{\text{ini}} \rightarrow s_2 \rightarrow s_6 \rightarrow s_3$  (from  $t = 0$  to  $t = 72$ ) have little difference, because  $\{s_2, s_6, s_3\}$  are aggregated, and the joint gradient contributed by them accounted for the largest proportion in the synthetical field gradient. During searching the first three sources, the robot based on the two algorithms makes not much difference in direction selection in each iteration. After accessing  $\{s_2, s_6, s_3\}$ , the robot starts to search for the remaining sources which are scattered. The most obvious difference based on the two algorithms is the trajectories of  $s_3 \rightarrow s_1$ , and the trajectory generated by robot based on

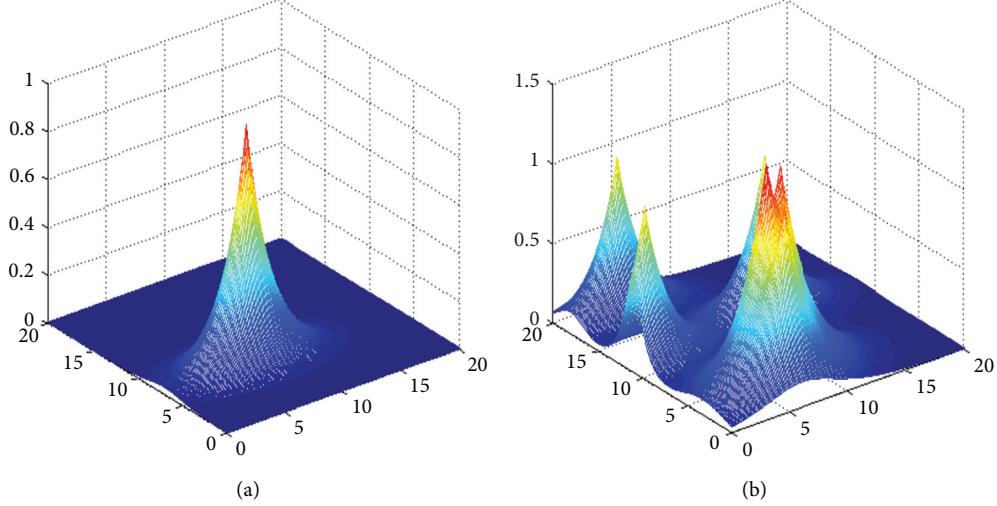


FIGURE 3: Field intensity distribution of sources. (a) Field intensity distribution of single source. (b) Synthetical field intensity distribution of six sources.

PDS is straighter. We can conclude that when the robot traverses the same sequence, the performance of PDS is better than PS.

The complexity of multi-source search mainly depends on the robot platform and source types. In the absence of background flow, the field intensity distribution of the source with chemical characteristics is a Gaussian-like shape distribution, while the light, RF, sound, etc. are modeled on an inverse square law distribution. The general search strategy is generally applicable to multi-source search tasks under different source types. If the source is light source, the field intensity model follows:

$$f_s(n) = \sum_{i=1}^m \frac{\lambda_i}{4\pi d_{no_i}^2}. \quad (14)$$

Equation (7) is replaced by equation (15) which denotes the original gradient component:

$$g_{f_i}(n) = \left( \frac{\lambda_i(x_i - x)}{2\pi d_{no_i}^3 a_i^2}, \frac{\lambda_i(y_i - y)}{2\pi d_{no_i}^3 b_i^2} \right). \quad (15)$$

The synthetical field intensity distribution is formed by six sources, as shown in Figure 5(a). However, it is difficult to directly observe the number and locations of the sources from a surface diagram. In Figure 5(b), the trajectories for two algorithms are similar to the Figure 4(h). In other words, both algorithms are versatile for searching for the light sources.

**4.2. The Comparisons under Different Central Field Intensities.** In this part, we study the performance of PS and PDS under different central field intensities on robotic source search task. All simulations were carried out using 20 sources. The central field intensity values of 60%, 20%, and 20% of the total number of sources are set to 1, 1.2, and 1.5, respectively. The locations of the sources are randomly generated, and there is no intersection between sources. In Figure 6, the

results show that both algorithms are effective for different source types with different central field intensities. In Figures 6(a)–6(d), it can be concluded that the trajectories for PS have more intersections than the corresponding ones for PDS. In other words, the trajectories for PDS are more reasonable. The relevant statistical efficiency analysis will be discussed in Section 4.5.

**4.3. Obstacle Avoidance Analysis.** This part presents several simulations in the environments with obstacles. The obstacle is represented by a circle ( $a = b$ ), and the central field intensity value  $\lambda_j$  is set to be opposite of that of the source.  $\gamma$  is set to 1/4 length of the sensing radius of robot, where  $\gamma = 0.25$ . Two set of comparison simulations were designed for different obstacle radii  $r_j$ . Although the obstacle provides repulsive gradient, the robot based on PS still cannot plan a reasonable direction of movement when encountering obstacles. In Figures 7(a) and 7(c), the robot hovers back and forth between the previous location and the obstacle center, and cannot avoid obstacles. In Figures 7(b) and 7(d), the robot based on PDS with decision-making factor has the ability to avoid obstacles, and the trajectories are smooth. On the other hand, it demonstrates that the PDS can also cope with complicated environment with obstacles.

**4.4. Local Optima Problem.** We construct a special case in which the robot is trapped in local optima point. For this simulation, the relevant parameters are shown in Table 2. In Figure 8, there is a point where the original synthetical field gradient is  $(0, 0)$  between two sources. In both PS and PDS, the position of robot is  $(9, 9)$ . However, the robot falls in the local optima trap as shown in Figure 9. The trajectories for PS and PDS with escape strategy (with  $\varepsilon$ ) are shown by Figures 10(a) and 10(b), respectively. The robot successfully escapes the local optima point. The robot can retrieve a non-zero field gradient information and continue its search task. It is important to note that the robot judges whether all the

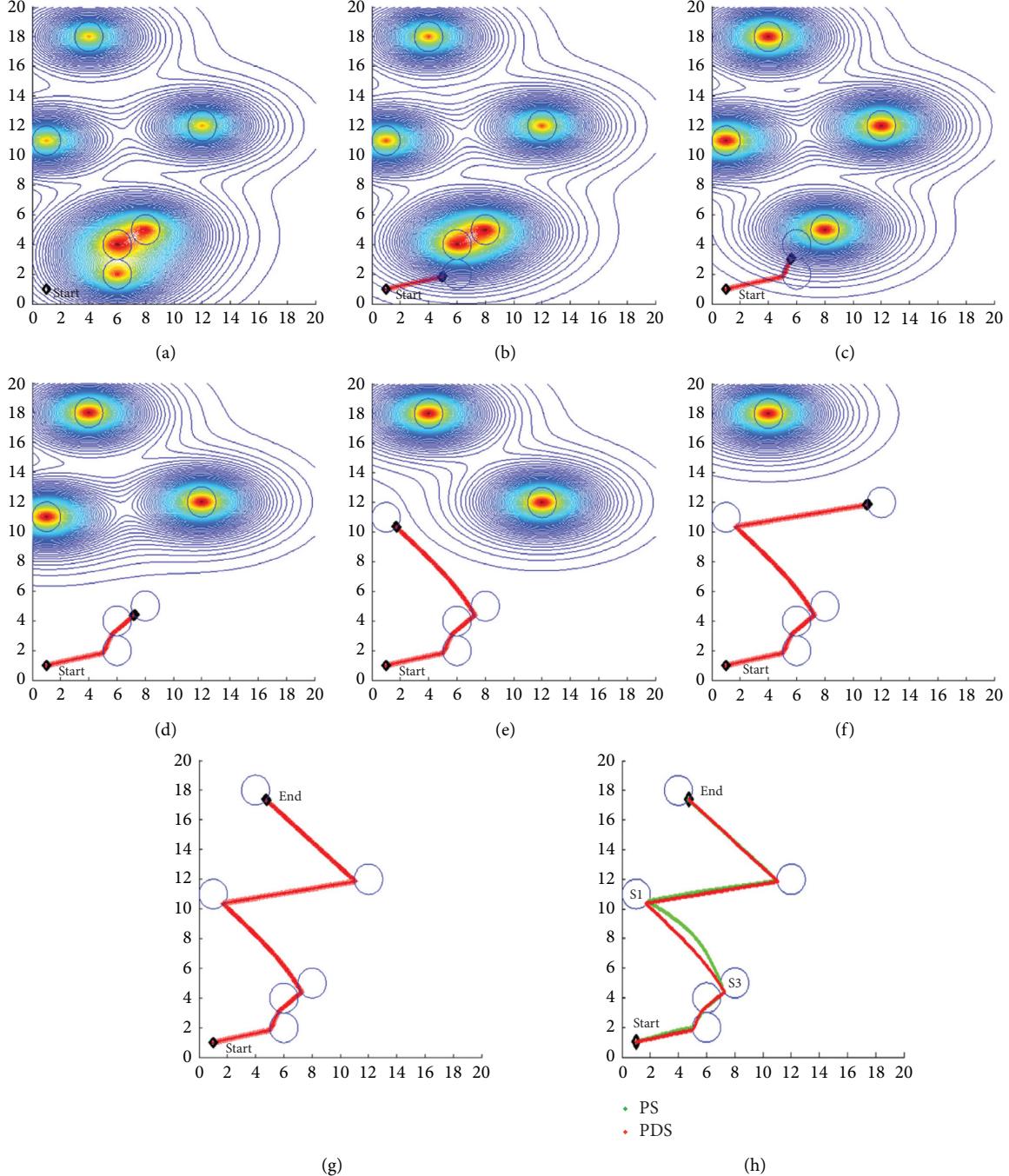


FIGURE 4: Process of generating the trajectory for PDS under sources with Gaussian-like shape field intensity distribution. (a–g) Access sequence and how a trajectory is built. (a)  $t=0$ , (b)  $t=43$ , (c)  $t=52$  (d)  $t=72$ , (e)  $t=156$ , (f)  $t=249$ , (g)  $t=333$ , (h) Trajectory comparison between PS and PDS.

sources have been accessed by perceiving the field intensity information rather than the field intensity gradient information.

**4.5. Algorithm Evaluation.** In this section, we present the statistical efficiency values for PDS compared with that for other algorithms in a closed environment without obstacles. First, a simple evaluation function is introduced. Then, the

comparison results of statistical efficiency value for various search strategies are given.

For the same source type, even the sources are distributed at different locations, the nature of all sources is similar. A simple and general statistical efficiency evaluation function is given in [10, 35] as follows:

$$\eta = \frac{N}{L}, \quad (16)$$

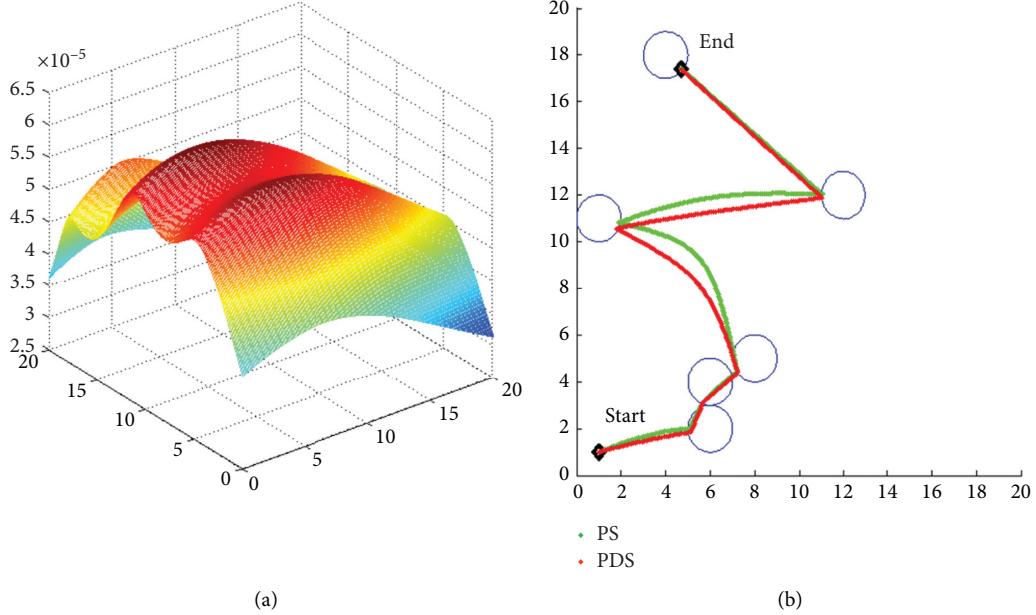


FIGURE 5: The simulations of robot search for multiple light sources. (a) The synthetical field intensity distribution for six light sources. (b) Trajectory comparison between PS and PDS.

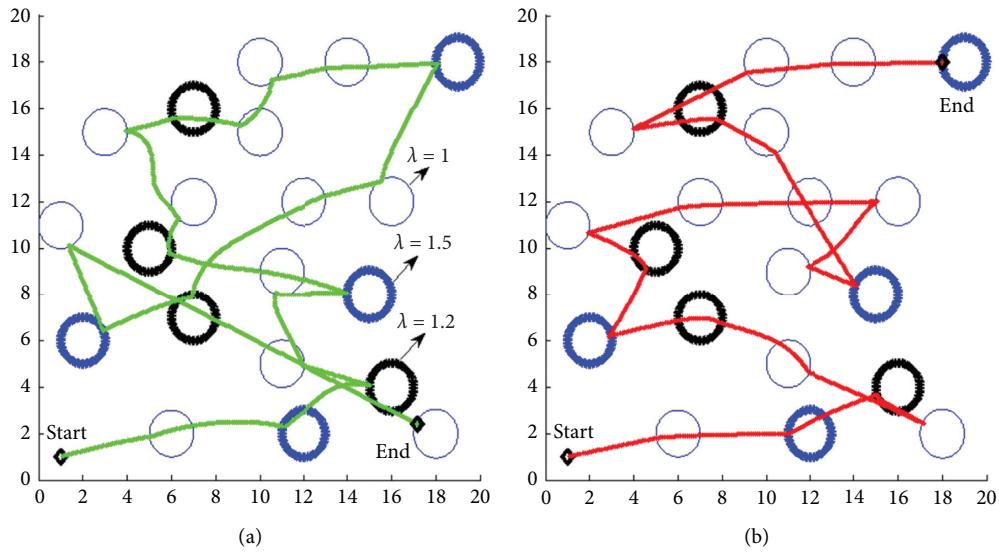


FIGURE 6: Continued.

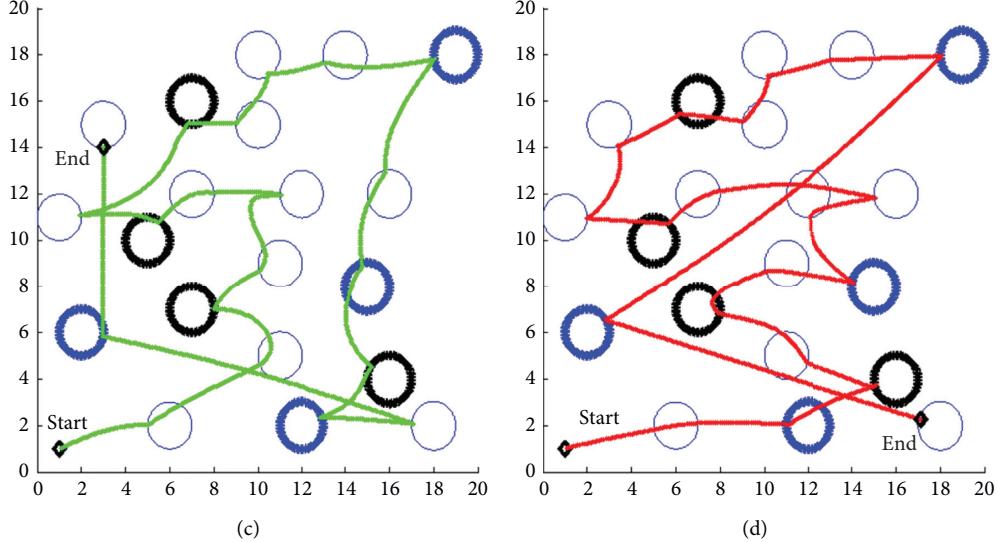


FIGURE 6: For different central field intensities, the trajectories for PS and PDS under the sources with different field intensity distributions. The correspondences are the blue circle- $\lambda = 1$ , color-emphasized blue circle- $\lambda = 1.5$ , and color-emphasized black circle- $\lambda = 1.2$ . (a) The trajectory for PS under the sources with Gaussian-like shape field intensity distribution. (b) The trajectory for PDS under the sources with Gaussian-like shape field intensity distribution. (c) The trajectory for PS under the light sources. (d) The trajectory for PDS under the light sources.

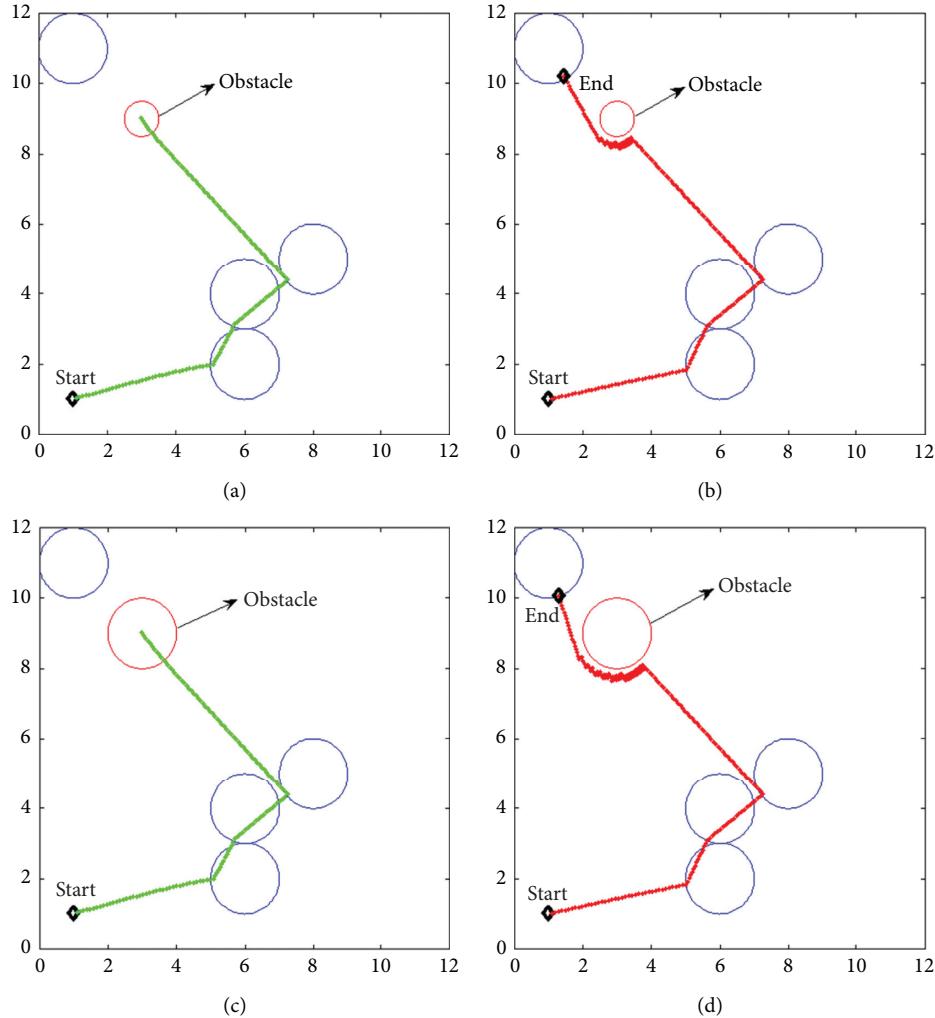


FIGURE 7: Comparison results for obstacle avoidance performance of robot based on two algorithms. The correspondences are green line-PS and red line-PDS. (a, b)  $r_j = 0.5$ . (c, d)  $r_j = 1$ .

TABLE 2: The source properties.

Source	$x$	$y$	$a$	$b$	$\lambda$
$s_1$	3	3	1	1	1
$s_2$	15	15	1	1	1

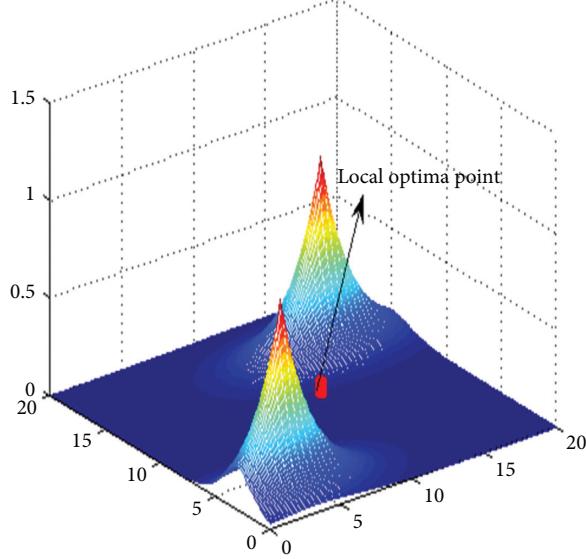


FIGURE 8: Local optima point problem.

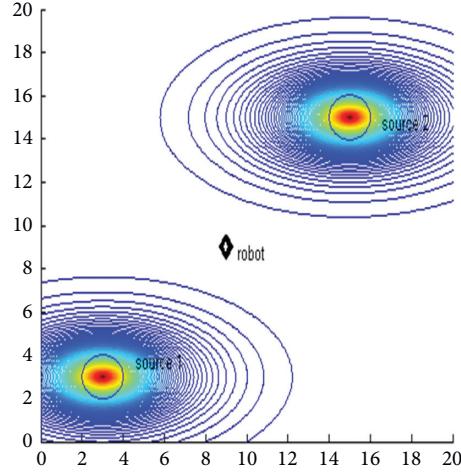


FIGURE 9: Falling the local optima trap with PD and PDS.

where  $N$  is the number of sources found.  $L$  is the trajectory length. Since the robot with constant speed  $\delta$ , equation (16) can be expressed as  $\eta = 1/(\delta \cdot \langle t_{se} \rangle)$ ,  $\langle t_{se} \rangle$  is the average time between successive encounters. This efficiency function does not depend on the number of sources but on the distribution of sources. Therefore, the evaluation function can reflect the efficiency of the algorithms, and the higher  $\eta$  value means the better performance.

The base reference is the Sweep strategy which requires a robot move from left to right and back and visit every point only once. The robot uses the Sweep strategy to carry out local environmental coverage and search for sources. The

robot stops when traverses all sources. We conducted related simulations for the comparison of statistical efficiency for PDS, the Sweep strategy, Yuragi-based strategy [14], and PS. For the Yuragi-based strategy, the noise parameter obeys  $\varepsilon_\omega(t) \sim N(0, 1)$  and Levy does not occur because the source particles completely cover the map.

For a scenario for specific number of sources, each algorithm is executed 30 times to reduce the influence of stochastic factors on the performance. The source locations are randomly generated each time.  $\bar{\eta}$  represents the mean statistical efficiency value. For all sources with central field intensity set to 1, 960 simulations are conducted: number of

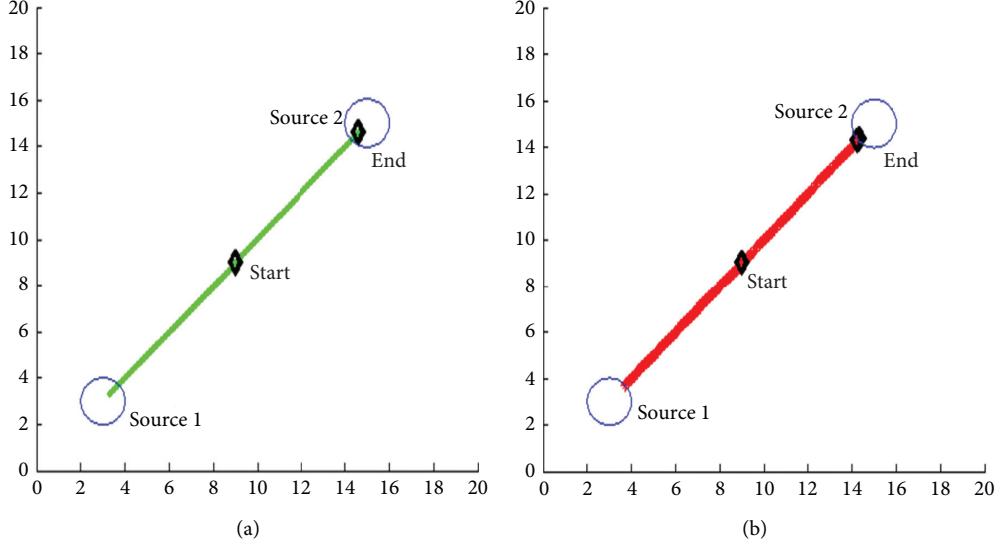


FIGURE 10: The trajectories for PD and PDS with escape strategy. (a) The trajectory for PS with escape strategy. (b) The trajectory for PDS with escape strategy.

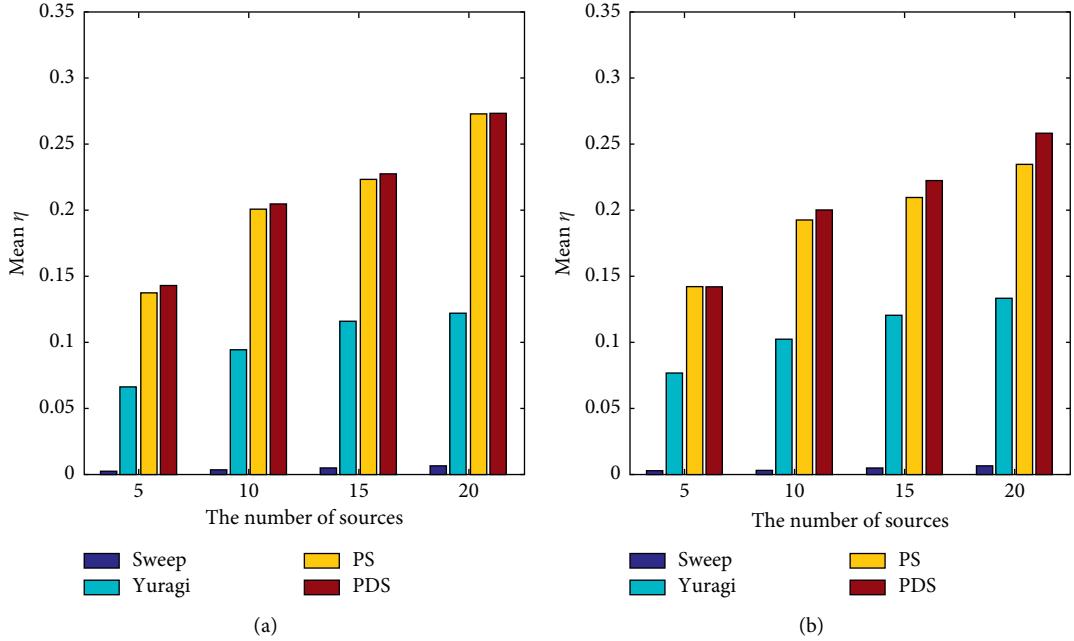


FIGURE 11: For the same central field intensity, comparison of  $\bar{\eta}$  for four algorithms. Blue, cyan, yellow, and red bars represent  $\bar{\eta}$  for the sweep, Yuragi, PS, and PDS, respectively. (a) The sources with the Gaussian-like shape field intensity distribution. (b) The light sources.

algorithms  $(4) \times$  number of the source number schemes  $(4) \times$  number of source types  $(2) \times$  number of simulation executions for a specific source number scheme  $(30)$ . The comparison results of  $\bar{\eta}$  are shown in Figure 11. For different source types, we can conclude that the  $\bar{\eta}$  values of PDS are higher than that of the Sweep strategy, Yuragi-based strategy, and PS, respectively. This means that the robot with decision-making ability performs the best. In Figures 11(a) and 11(b) is a common trend that the  $\bar{\eta}$  value increases with the number of sources. That is because  $\bar{\eta}$  value depends on the location distribution of sources. In a closed environment, all the source

locations are randomly generated and have no intersection area with each other. The greater number of sources along with the shorter mean time between successive encounters results in a larger  $\bar{\eta}$  value. For the sources with different central field intensities, the central field intensity values of 60%, 20%, and 20% of the total number of sources are set to 1, 1.2 and 1.5, respectively. 480 simulations are carried out: number of algorithms  $(4) \times$  number of the source number schemes  $(2) \times$  number of source types  $(2) \times$  number of simulation executions for a specific source number scheme  $(30)$ . The comparison results of  $\bar{\eta}$  are shown in Figure 12. Because

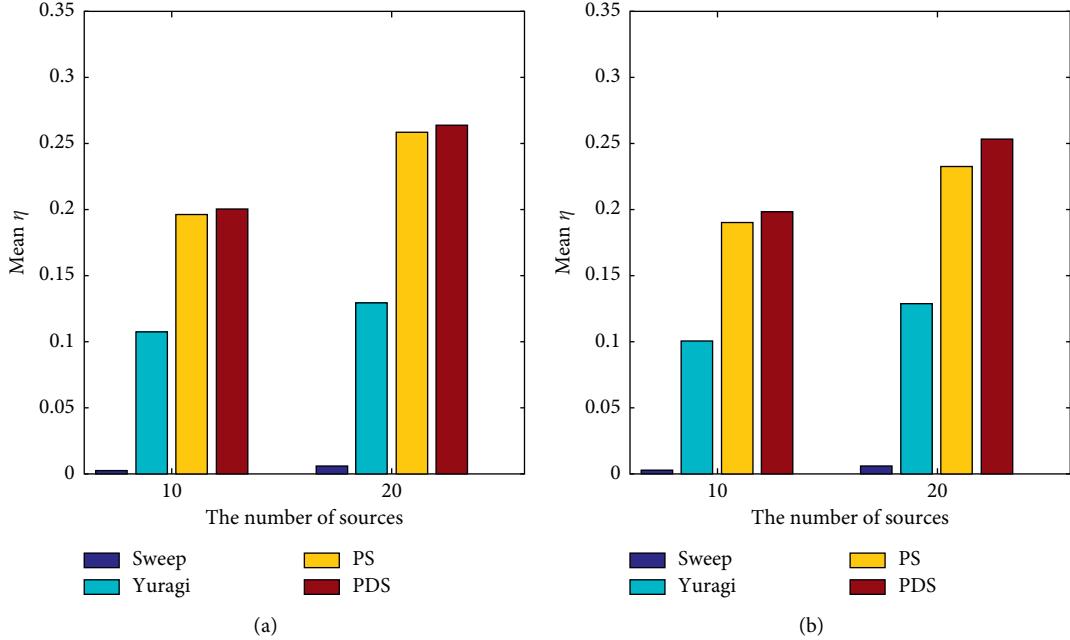


FIGURE 12: For different central field intensities, comparison of  $\bar{\eta}$  for four algorithms. Blue, cyan, yellow, and red bars represent the  $\bar{\eta}$  for the sweep, Yuragi, PS, and PDS, respectively. (a) The sources with the Gaussian-like shape field intensity distribution. (b) The light sources.

all center field intensities are not completely uniform, we consider two schemes with 10 and 20 sources for simplicity. In Figures 12(a) and 12(b), the change trend and comparison results of  $\bar{\eta}$  are similar to that described above. The PDS improves the search efficiency of the robot in various conditions by introducing decision-making factor in the synthetical field gradient. It should be noted that the evaluation function is no longer applicable when the central field intensity of the source changes with different magnitudes over time, which is beyond the scope of this paper.

## 5. Conclusions

In this paper, inspired by the foraging behavior of *Physarum polycephalus*, we establish two bio-inspired strategies for multi-source search task. The PS is proposed as a benchmark, which is efficient and easy to implement in an obstacle-free environment. The PDS based on PS has taken decision-making for each source and obstacle avoidance into account. The advantage of the two proposed algorithms are that the robot does not need to know the location of all sources, but only perceives the field intensity at the current position and estimates the gradient to determine the next action. The proposed algorithms prevent the robot from relocating the same source which has been located before. Simulations under different source types and central field intensity show that PS and PDS have certain versatility. Compared with PS, PDS can reduce redundant paths and endow the robot with obstacle avoidance capacity, improving greatly the search security. Furthermore, the related simulation results show that PDS has the best statistical efficiency value compared with other three methods.

Also, as future work, one research direction is multi-source searching task in such condition, and there is no

source signal in some areas of the environment. In addition, this paper focuses on a single robot, and the algorithms can be extended to swarm robotics system by increasing the co-operation between them.

## Data Availability

The simulation data and codes used to support this paper have not been made public since related information is related to personal privacy of academic achievements of each author.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

The authors thank Huankun Sheng for the suggestions and discussions for this work. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0102702.

## References

- [1] R. A. Russell, D. Thiel, R. Deveza, and A. Mackay-Sim, “A robotic system to locate hazardous chemical leaks,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 556–561, ICRA, Nagoya, Japan, May 1995.
- [2] R. A. Russell, “Robotic location of underground chemical sources,” *Robotica*, vol. 22, no. 1, pp. 109–115, 2004.
- [3] K. Nickels, H. Nguyen, D. Frasch, and T. Davison, “Effective exploration behavior for chemical-sensing robots,” *Biomimetics*, vol. 4, no. 4, p. 69, 2019.

- [4] V. San Juan, M. Santos, and J. M. Andújar, "Intelligent UAV map generation and discrete path planning for search and rescue operations," *Complexity*, vol. 2018, Article ID 6879419, 17 pages, 2018.
- [5] T. Kalmar-Nagy, G. Giardini, and B. D. Bak, "The multiagent planning problem," *Complexity*, vol. 2017, Article ID 3813912, 12 pages, 2017.
- [6] L. Marques and A. T. De Almeida, "Electronic nose-based odour source localization," in *Proceedings of the 6th International Workshop on Advanced Motion Control, AMC*, pp. 36–40, Nagoya, Japan, April 2000.
- [7] H. Ishida, H. Tanaka, H. Taniguchi, and T. Moriizumi, "Mobile robot navigation using vision and olfaction to search for a gas/odor source," *Autonomous Robots*, vol. 20, no. 3, pp. 231–238, 2006.
- [8] Y. Yang, Q. Feng, H. Cai et al., "Experimental study on three single-robot active olfaction algorithms for locating contaminant sources in indoor environments with no strong airflow," *Building and Environment*, vol. 155, pp. 320–333, 2019.
- [9] K. McGill and S. Taylor, "Robot algorithms for localization of multiple emission sources," *ACM Computing Surveys*, vol. 43, no. 3, p. 15, 2011.
- [10] M. E. Wosniack, E. P. Raposo, G. M. Viswanathan et al., "Efficient search of multiple types of targets," *Physical Review E*, vol. 92, no. 6, Article ID 062135, 2015.
- [11] X.-X. Chen and J. Huang, "Odor source localization algorithms on mobile robots: a review and future outlook," *Robotics and Autonomous Systems*, vol. 112, pp. 123–136, 2019.
- [12] J.-G. Li, J. Yang, J. Liu, G.-D. Lu, and L. Yang, "Odor-source searching using a mobile robot in time-variant airflow environments with obstacles," in *Proceedings of the 33rd Chinese Control Conference, CCC*, pp. 8559–8564, Nanjing, China, July 2014.
- [13] A. Dhariwal, G. S. Sukhatme, and A. A. G. Requicha, "Bacterium-inspired robots for environmental monitoring," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, pp. 1436–1443, New Orleans, LA, USA, May 2004.
- [14] S. G. Nurzaman, Y. Matsumoto, Y. Nakamura, S. Koizumi, and H. Ishiguro, "'Yuragi'-based adaptive mobile robot search with and without gradient sensing: from bacterial chemotaxis to a Levy walk," *Advanced Robotics*, vol. 25, no. 16, pp. 2019–2037, 2011.
- [15] S. G. Nurzaman, Y. Matsumoto, Y. Nakamura, K. Shirai, and H. Ishiguro, "Bacteria-inspired underactuated mobile robot based on a biological fluctuation," *Adaptive Behavior*, vol. 20, no. 4, pp. 225–236, 2012.
- [16] V. Gazi and K. M. Passino, "Stability analysis of social foraging swarms," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 539–557, 2004.
- [17] S. Xu, K. Dogancay, and H. Hmam, "3D pseudolinear Kalman filter with own-ship path optimization for AOA target tracking," in *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3136–3140, Shanghai, China, May 2016.
- [18] S. Xu, K. Dogancay, and H. Hmam, "Distributed pseudolinear estimation and UAV path optimization for 3D AOA target tracking," *Signal Processing*, vol. 133, pp. 64–78, 2017.
- [19] W. Jatmiko, P. Mursanto, and B. Kusumoputro, "Modified PSO algorithm based on flow of wind for odor source localization problems in dynamic environments," *Wseas Transaction on Systems*, vol. 7, no. 2, pp. 106–113, 2008.
- [20] W. Jatmiko, W. Pambuko, P. Mursanto et al., "Localizing multiple odor sources in dynamic environment using ranged subgroup PSO with flow of wind based on open dynamic engine library," in *Proceedings of the 2009 International Symposium on Micro-NanoMechatronics and Human Science*, pp. 602–607, Nagoya, Japan, November 2009.
- [21] W. Jatmiko, W. Pambuko, A. Febrian et al., "Ranged subgroup particle swarm optimization for localizing multiple odor sources," *International Journal on Smart Sensing and Intelligent Systems*, vol. 3, no. 3, pp. 411–442, 2010.
- [22] M. Dadgar, S. Jafari, and A. Hamzeh, "A PSO-based multi-robot cooperation method for target searching in unknown environments," *Neurocomputing*, vol. 177, pp. 62–74, 2016.
- [23] Y. Yan, R. Zhang, J. Wang, and J. Li, "Modified PSO algorithms with "request and reset" for leak source localization using multiple robots," *Neurocomputing*, vol. 292, pp. 82–90, 2018.
- [24] A. Tero, R. Kobayashi, and T. Nakagaki, "A mathematical model for adaptive transport network in path finding by true slime mold," *Journal of Theoretical Biology*, vol. 244, no. 4, pp. 553–564, 2007.
- [25] Y.-P. Gunji, T. Shirakawa, T. Niizato, and T. Haruna, "Minimal model of a cell connecting amoebic motion and adaptive transport networks," *Journal of Theoretical Biology*, vol. 253, no. 4, pp. 659–667, 2008.
- [26] A. Adamatzky, "If BZ medium did spanning trees these would be the same trees as *Physarum* built," *Physics Letters A*, vol. 373, no. 10, pp. 952–956, 2009.
- [27] J. Jones, "Approximating the behaviours of *Physarum polycephalum* for the construction and minimisation of synthetic transport networks," in *Lecture Notes in Computer Science*, vol. 5715, pp. 191–208, Springer, Berlin, Germany, 2009.
- [28] Y. Wu, Z. Zhang, Y. Deng, H. Zhou, and T. Qian, "A new model to imitate the foraging behavior of *Physarum polycephalum* on a nutrient-poor substrate," *Neurocomputing*, vol. 148, pp. 63–69, 2015.
- [29] Y. Liu, C. Gao, Z. Zhang et al., "A new multi-agent system to simulate the foraging behaviors of *physarum*," *Natural Computing*, vol. 16, no. 1, pp. 15–29, 2017.
- [30] A. Adamatzky, "Slime mold solves maze in one pass, assisted by gradient of chemo-attractants," *IEEE Transactions on NanoBioscience*, vol. 11, no. 2, pp. 131–134, 2012.
- [31] A. Adamatzky and J. Jones, "On using compressibility to detect when slime mould completed computation," *Complexity*, vol. 21, no. 5, pp. 162–175, 2016.
- [32] J. Jones, "Towards programmable smart materials: dynamical reconfiguration of emergent transport networks," *International Journal of Unconventional Computing*, vol. 7, no. 6, pp. 423–447, 2011.
- [33] X. Zhang, Y. Zhang, Z. Zhang, S. Mahadevan, A. Adamatzky, and Y. Deng, "Rapid physarum algorithm for shortest path problem," *Applied Soft Computing*, vol. 23, pp. 19–26, 2014.
- [34] X. G. Zhang, A. Adamatzky, F. T. S. Chan et al., "A biologically inspired network design model," *Scientific Reports*, vol. 5, p. 10794, 2015.
- [35] N. Pereira, A. F. Ribeiro, G. Lopes, and J. Lino, "Path planning towards non-compulsory multiple targets using TWIN-RRT," *Industrial Robot: An International Journal*, vol. 43, no. 4, pp. 370–379, 2016.