Wireless Pers Commun
DOI 10.1007/s11277-016-3607-z

Bio-inspired Patrolling Scheme Design in Wireless and Mobile Sensor and Robot Networks

Yanping Zhang[1] · Yang Xiao[2,3] · Yun Wang[4] · Patrick Mosca[1]

Abstract Environment monitoring is an essential application in wireless sensor networks. We consider a large area surveillance in which a monitored region is divided into small sub areas where sensors, sinks, and mobile robots are responsible for surveillance jobs. Sensors collect data from the environment for specific monitoring purposes. Sinks are usually stationary and responsible for collecting regional information, while mobile robots are always patrolling the monitored areas to handle events. Under the above scenario, we propose anti-inspired digital pheromone to classify priorities of events used in a proposed patrolling algorithm, in which, mobile robots are patrolling and handling events with priorities along the roads. Events with higher priorities are handled with higher priorities, while productive event handling, i.e., handling more events, is also the goal. Extensive simulations are conducted to study the efficiency of the proposed method.

Keywords Patrolling · Wireless sensor networks · Robot networks

✉ Yang Xiao
yangxiao@ieee.org

Yanping Zhang
zhangy@gonzaga.edu

Yun Wang
ywang2@fsmail.bradley.edu

Patrick Mosca
pmosca@zagmail.gonzaga.edu

1 Department of Computer Science, Gonzaga University, Spokane, WA 99258, USA

2 School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

3 Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487-0290, USA

4 Department of Computer Science and Information Systems, Bradley University, Peoria, IL 61625, USA

Published online: 17 August 2016

1 Introduction

Wireless sensor networks (WSNs) are useful for many applications including environment monitoring [1–4]. In [5, 6], we developed a divide-and-conquer based surveillance methodology, in which a monitored region is divided into small sub areas where sensors, sinks, and mobile robots are responsible for surveillance jobs. Static sensor nodes collect sensing data and send the data to the nearby sinks, while mobile robots are patrolling and handling events. Different sensors are responsible for different tasks or targets. Sinks are responsible for collecting and managing regional information. Robots are responsible for handling events collected from sensors and sinks in the monitored area [5, 6]. Sensors form connected or disconnected networks, while mobile robots can enhance connectivity as well as coverage. Disconnected or isolated sensors can buffer the sensing data until mobile robots passing by. Furthermore, mobile robots have sensors and Radio-frequency identification (RFID) tags carrying specific information for communication, which can be deployed by mobile robots to improve connectivity [7].

When there are more and more events, a mobile robot tries to figure out a path with the largest number of important events. Therefore, we propose a patrolling algorithm using bio-inspired digital pheromone in this paper. Patrolling problem research has many modern day applications such as improving the allocation and intelligence of drones or robots in military surveillance situations [8]. The taxi industry is also being revolutionized by services such as *Uber* and *Lyft* [9]. These services use a similar agent-event pattern to issue taxi cabs to nearby clients. In the past, taxis have been issued to customers manually, and it is often highly inefficient. Investigations into the patrolling problem can improve the pickup and drop off time of customers, as well as minimizing gas and miles wasted.

In our proposed scheme, we learn from ants' pheromone and propose the concept of digital pheromone and a patrolling algorithm for mobile robots using the digital pheromone. In our scheme, we differentiate events with different priorities using the different amounts of digital pheromone which evaporates over time; digital pheromones are formed among the sensors and the sinks when a sensor detects an event and sends a message to its nearby sink along with the relay sensors; several different events may be recorded in one sensor due to both detection and relaying, and they are added up to the total digital pheromone value in this sensor; the strongest digital pheromone path with a large number of high priority events will be chosen by a mobile robot for the coming patrolling path when the robot reaches an intersection.

Modeled as ants, robots do "release" materials. Although pheromones are generated and distributed by sensors, robots are able to deploy sensors and RFID tags [7]. RFID tags are deployed to assist the communication among robots, and robots can also deploy new sensors to improve network connectivity. As sensors are only originally deployed in critical areas for saving resources and energy, coverage and connectivity may be impacted. During patrolling, robots can pick up messages from isolated sensors or deploy extra sensors to make isolated sensors connected. Meanwhile, through the assistance of RFID tags [10], mobile robots are able to communicate with one another. By picking up the information carried by RFID tags, mobile robots could be able to get messages from others.

The remainder of the paper is organized as follows: In Sect. 2, we review related works of patrolling methods and bio-inspired research. Section 3 proposes the patrolling algorithm. Section 4 provides simulation results. Conclusions are presented in Sect. 5.

2 Related Works

Bio-inspired research has become a hot topic in computer science, especially in wireless networks. For example, scent-marking, a communication method among primates, are studied in [7, 9, 11], and in our previous work [7, 12], a primate-inspired method for mobile robot pursuing and tracking is proposed, while primate scent-marking is similar to ant pheromone communication, but it contains more information, much like a message.

Ant social behavior, such as leaving pheromone on the trail for other ants, inspires people who utilize a method called ant colony optimization to solve solving combinational optimization problems [13]. Applications include solutions for a traveling salesman problem to achieve the shortest path [14, 15].

Ant-like agents are adopted for routing information exchanges in wireless mobile ad hoc networks to reduce bandwidth overhead, to improve the connectivity of the network, and to make better routing decisions compared with proactive routing protocols [16].

In [17], sensor nodes are modeled as ants; the target-tracking task is modeled as ants' food locating task; pheromones will be released when food is found or a target exists; the diffusion, loss, and accumulation of pheromones are used to simulate the communication, invalidation, and fusion of target information during the target tracking; accumulated pheromones are used to measure the existence of a target, which is also used in the subsequent round as the determinant of the probability of ant-searching activity.

In [18], a dynamic itinerary planning for mobile agents in wireless sensor network is proposed for an agriculture application with a special concern of crop monitoring and frost prediction. Itinerary planning is quite a similar topic with the patrolling problem, and the itinerary of mobile agents is determined step by step based on interested sensing data from the mobile agents' sounding sensors.

In [19], the problem of planning patrol routes for police patrolling is studied, aiming to maximize coverage of important locations (called as "hot spots") with the minimum cost. The cost of patrolling is evaluated as the length of patrolling [19]. An edge-weighted graph is employed to model road networks, and each edge represents a road, while the edge weight is used to denote the importance of the road.

In [20], patrolling mechanisms for disconnected targets are studied and the main concern is to patrol efficiently in order to optimize the visiting intervals of all target points. Swarm-intelligence is used to improve quality of routes with only local communications for wireless ad hoc networks [13, 21].

In [22], a survey for bio-inspired visual attention in agile sensing for target detection is presented. In [23–25], target tracking methods of binary wireless sensor networks are proposed and studied. In [26], RFID tags are studied for tracking. In [27], construction of tree network for homogeneous wireless sensor networks is studied. In [28], a related routing method is presented.

Although there are some research efforts on bio-inspired methods as listed in the above, most of these methods are designed for different problems and not applicable in our research work. We do not target to handle all of events but efficiently handle more important events. We also consider the situation that a lot of events exist in the monitored field but not only several events happening occasionally. In our work, we design an innovative patrolling algorithm based on digital pheromone, which is established by sensor nodes and spread by network connections. The digital pheromone on a route is also being accumulated while different digital pheromones are spread on the same route. Digital pheromone is also evaporating according to time, and this is used to model the priority of events decreasing over time. Based on the evaluation of digital pheromone, robots choose their patrol route with the highest value of digital pheromone.

3 Digital Pheromone Based Patrolling Algorithm

In this section, we first present digital pheromone, and then design two digital pheromone-based patrolling algorithms: greedy patrolling algorithm and collaboration patrolling algorithm.

3.1 Digital Pheromone

Pheromone is the chemical material which can be used as a mean of communication for ants. Furthermore, pheromone evaporates over time. In this paper, we propose to use differentiated digital pheromones to classify priorities of events for communications and collaborations among mobile robots and sensors. Digital pheromone also evaporates with time to model the downgrade of events' priorities. A message is generated once an event is detected. A digital pheromone is created as an important component of the message and evaporates over time to reflect the event's real time priority. The value of digital pheromone is decreasing according to a function. Meanwhile, each message has a lifetime. When digital pheromone evaporates to zero, the corresponding message reaches its lifetime.

Digital pheromone generated by an original sensor node will be spread in a connected sensor network via relay nodes to the nearby sink, and all relay nodes keep a copy of the digital pheromone so that a path of digital pheromone is formed. Some sensors may be isolated and some networks can be disconnected. Digital pheromone will be kept in the isolated sensors or disconnected networks waiting mobile robots to collect them before they evaporate. If mobile robots collect the message, the digital pheromone will be cleared. Mobile robots plan their patrolling paths based on digital pheromones. If a mobile robot needs help to handle events, it will deploy RFID tags carrying messages for other robots.

3.2 Greedy Patrolling Algorithm

In this subsection, we propose two greedy algorithms: a local greedy algorithm and a global greedy algorithm.

Nearby sensors and sinks are contacted by the mobile robot for collecting digital pheromones on each road path when a mobile robot reaches an intersection of roads. The robot will choose a path with the largest value of digital pheromone among several paths if available. If some values of digital pheromone are not available, the robot only chooses roads with available values. The robot randomly chooses a road that not visited recently if digital pheromone is not available for all roads.

In the local greedy patrolling algorithm, the next patrolling path is $path = k$, where $P_k = \max(P_1, P_2, \ldots, P_n)$, and $P_1, P_2 \ldots P_n$ are digital pheromones if n paths are available. For example, in Fig. 1, "AB" will be chosen if "AC" has a lower digital pheromone value. The idea is similar to the notions established throughout the study [19].

In the global greedy patrolling algorithm, several segments of paths in terms of digital pheromone values are considered since important events may happen in several road segments away. The algorithm needs to compare digital pheromones of next several road

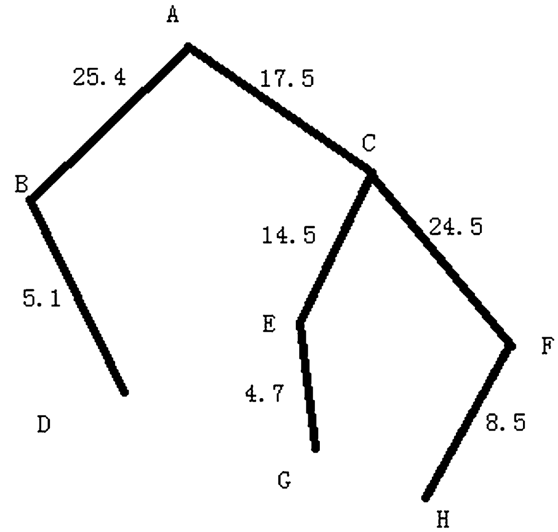


Fig. 1 Roads at intersection *A*

segments. An important consideration would be the evaporation of the pheromone. High priority but far away events may become less important when robots finally arrive the road. Therefore, robots should calculate events' final pheromone value in advance. We use a function $f(t) = a - b*t$ to model the evaporation of digital pheromones, where t is time, a is initial value of digital pheromone, and b is a rate of evaporation. Let $t_1, t_2, \ldots, t_i$ denote the traveling times required road segments #1, #2, …, #i in a path, respectively. The estimated pheromone is $E = \sum_{j=1}^{i} f\left(\sum_{k=1}^{j} t_k\right)$.

Figure 1 shows an example of a road topology with lengths. For simplicity, in Fig. 1, assume that robots move with a speed of $v = 1$, and the evaporation model of digital pheromone is $f(t) = 50 - t$. Therefore, estimated digital pheromones of routes is $E_{fAD} = 69.5$, $E_{fAG} = 63.8$, and $E_{fAH} = 43.8$. The best route is $A \geq B \geq D$.

Tables 1 and 2 show the patrolling algorithm design as well as path calculation.

3.3 Collaboration Patrolling Algorithm

Collaborated robots as a group can handle events in a collaborative way [29, 30]. There are two kinds of collaborations: (1) each mobile robot is responsible for a small specific region and robots collaborate on the boarders of the region; (2) mobile robots patrol the whole area.

In large area surveillance, we divide the monitored field into small subareas. Each subarea may have different concerns. Then, we need to employ robots with corresponding capabilities in each area.

Figure 2 illustrates two considerations that robots are patrolling: a pre-assigned responsible area (Fig. 2b) or without a responsible area (Fig. 2a).

Figure 2a describes the algorithm that robots are not pre-assigned responsible areas. Whenever robots arrive at an intersection, they will greedily choose the most efficient unpatrolled road with the largest digital pheromone. Figure 2b shows the case that every robot is responsible for one region and a patrolling robot will turn around when reaching the border.

Mobile robots initially randomly patrol in their regions and event locations are used to build a graph or topology based on which mobile robots collaborate with each other, assuming that robots have GPS to approximate exact locations. When robots need help,

Table 1 Patrolling algorithm

```
While Current_task[i]!=Null:
  Go on Patrolling
   If Event exists on current road
     CurrentTask[i]← event within the first RFID
   End
 End
 Call Path_Calc(loc[i], Loc_CurrentTask)
  Loc[i]←location of the event
  Time[i]← Time[i]+travel time +handle time
  Event_handled[i]++
     RFIDs related to the event←0
     Task[i]->currentevent ← "Finished"
Path_Calc(loc[i], Loc_CurrentTask)
  While(Loc[i]!= Loc_CurrentTask)
   If Itersection_flag= false  //patrol on a road
     Loc[i]← the other end of the road
   Else if Itersection_flag= true //arrive at intersection
    Next_Road← Road with max digital pheromone
     Loc[i]← the other end of the road
   End
```

Table 2 Algorithm of path calculation

```
Path_Calc(loc[i], Loc_CurrentTask)
  While(Loc[i]!= Loc_CurrentTask)
  Calculate for the direction through location differences
   If Itersection_flag= false //patrol on a road
     Loc[i]← the other end of the road
     Record the path in Path[i]
   Else if Itersection_flag= true //arrive at intersection
    Next_Road← Road with max digital pheromone
     Loc[i]← the other end of the road
     Record the path in Path[i]
   End
```

they can deploy RFID tags or sensors to leave messages. When receiving recourse messages from others, they will consequently try to do collaboration.

Mobile robots also communicate with sensor nodes and sinks to update information including events and topologies. Following assumptions are made: (1) a mobile robot will use the shortest path based on a topology and ignore all events alone the road if it is assigned to a region; (2) sensor nodes spread digital pheromones when detecting an event; the digital pheromones for the same event in all relay sensors keep consistent with the original one and evaporate over time synchronously; we use f(t) to denote the evaporating function of digital pheromones; the total digital pheromone on a road is the accumulation of all events relayed on the road; (3) messages will only be relayed to the nearest sink; however, sinks can communicate with each other to exchange information; (4) a robot will

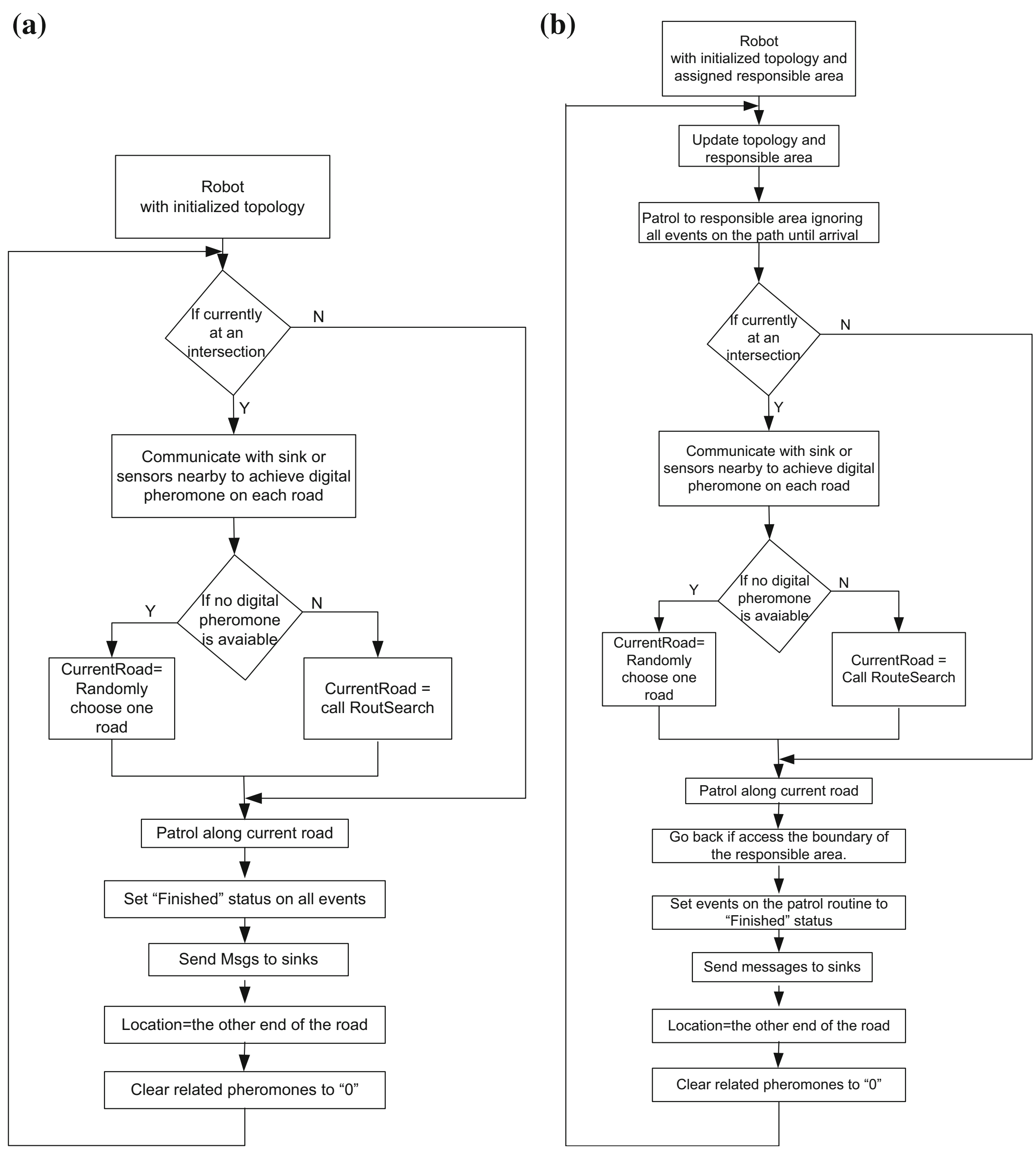


Fig. 2 Greedy patrolling algorithm. **a** Robots with no responsible area. **b** Robots with responsible area

plan its patrol route when it arrives at an intersection, and communicates with the nearby sink to obtain pheromone values; then it chooses one road based on the greedy patrolling algorithm; (5) a robot will collect all messages on its patrol route; once any event is handled, all pheromones related to it are cleared.

Tables 3 and 4 illustrate the collaboration algorithm and the collaboration patrolling algorithm, respectively, where Center_Calc_task is a management center function used to allocate tasks to each robot based on its location and events database and Center_Calc_loc is a management center function used to calculate the area for each robot and return the nearest location for the robots in the responsible area.

If robots are equipped with high transmission power, they can directly communicate with each other. Then they are able to plan their patrol for collaboration by themselves. Table 5 illustrates the collaboration patrolling algorithm based on direct communication assumption.

Table 3 The collaboration algorithm in management center

```
1.  Initialize the Topology, events and loc of robots
2.  Call Timer( Event_Update);
3.  For i 1 to n  //n is the number of robots
4.    Path[i] ← call Path_Calc(loc[i], Assign_loc[i])
5.  End
6.  For i 1 to n  //n is the number of robots
7.    For j 1to n (j<>i)
8.    Flag[i,j]← check Path[i] with all other Path[j] (j<>i)
9.      (Flag=1 if intersecting else Flag=0)
10.  If  Flag[i,j]=1
11.    Assign_loc[i], Assign_loc[j]← ReCalc_loc(i, j).
12.    Task[i],Task[j]←ReAssign(i,j)
13.  End
14.  Loc[i]←Assign_loc[i]
15.  Time[i]←travel time to destination of robot i
16.  Event_handled[i]← 0
17.End
18.Call Patrol_Algorithm
19.Timer_Expire_Function (Event_Update){
1) For i 1 to n
   If(Current_task[i]!=Null)
      Current_task[i]← "Finish"
      Time[i]← Time[i]+current task time
   end
  2) CenterEventBase[i]← "Finish" or not.
  3) Task[1 to n] ← Call Center_Calc_task
4) Loc[1 to n]← Call Center_Calc_loc}
```

3.4 Responsibilities of Sensors

Figure 3 illustrates the main responsibilities of sensors. The primary responsibility is to monitor the surrounding area. If a sensor node detects an event, it creates a message and corresponding digital pheromone which evaporates over time. It then spreads the event and pheromone values to the nearest sink. The second responsibility of sensors is to relay events and digital pheromones to sinks. The transmission delay is considered to keep pheromone value consistent with the original one. The 3rd responsibility is to collaborate with other mobile robots. A robot will collect information from sensors when passing by them.

A table is established in each sensor to record events and their digital pheromones. The table records all events detected or relayed by the sensor. Whenever a digital pheromone of an event is collected by a robot, the record will be cleared.

RFID tags are introduced into our design: (1) whenever a robot needs assistance, it will deploy RFID tags along its patrolling route; RFID tags carry information from the robot; (2) RFID tags are passive devices carrying information to be picked up, while sensors are proactive devices that actively monitor the surrounding area. Some specially-designed sensors can also achieve information from RFID tags.

Table 4 Collaboration patrolling algorithm

```
1. Communicate with sinks to update topology and responsible area
2. Calculate the quickest path to its responsible area and ignore events on the
road until arriving at its responsible area. Location of robot←the end of the
path.
3. WHILE (there are still events)
4.   If (currently not at an intersection)
5.   //Patrol along current road and handle events on it
6.   Events on the road←"Finished"
7.   Location of robot←the other end of the road
8.   All digital pheromone on the road←0
9.   Send MSG to sink about the handled events.
10.   END
11   //currently arrival at an intersection
12.  Communicate with sink or sensors nearby
13.   If (no digital pheromone is available)
14.    PatrolRoad←Randomly choose an unpatrolled road
15  ELSE
16.   PatrolRoad←The road with largest digital pheromone
17.   END
18.   Go on patrolling on the patrol road
19.   Events on the road←"Finished"
20.   Location of robot←the other end of the road
21.   All digital pheromone on the road←0
22.   Send MSG to sink about the handled events.
23. END
24.  //Update the topology and responsible area again.
25.  Call Patrolling Algorithm again.
```

4 Simulations

In this section, we design simulations using C++ and Java to study the digital pheromone based patrolling algorithms. Our simulation is designed based on event driven simulation design. Events are randomly generated. We study both greedy patrolling algorithms with a single robot and collaborative patrolling with multiple robots. For greedy algorithm, both local greedy patrolling algorithm and global greedy patrolling algorithm are studied. Two kinds of digital pheromones are used, constant model and decreasing model. To further evaluate the performance of our algorithm, we also compare our algorithm with some other patrolling methods.

4.1 Simulation for a Single Robot Patrol

In this subsection, we first use the topology as shown in Fig. 4 and make the following assumptions.

1. Sensor nodes are deployed along edges. Mobile robots also patrol following edges. A sink node is deployed at each vertex, which is also an intersection. Events are generated on edges or vertices following random distribution. A message is created when an event is detected and sent to the nearest sink via relay sensor nodes. Assume

Table 5 Collaboration patrolling algorithm based on direct communication

Collaboration Patrolling Algorithm:
1. Communicate with sinks to update topology and responsible area
2. Calculate the quickest path to its responsible area and ignore events on the road until arriving at its responsible area. Location of robot←the end of the path.
3. WHILE (there are still events)
4. If (currently not at an intersection)
5. //Patrol along current road and handle events on it
6. Events on the road←"Finished"
7. Location of robot←the other end of the road
8. All digital pheromone on the road←0
9. Send MSG to sink about the handled events.
10. END
11 //currently arrival at an intersection
12. Communicate with sink or sensors nearby
13. If (no digital pheromone is available)
14. PatrolRoad←Randomly choose an unpatrolled road
15 ELSE
16. PatrolRoad←The road with largest digital pheromone
17. END
18. Go on patrolling on the patrol road
19. Events on the road←"Finished"
20. Location of robot←the other end of the road
21. All digital pheromone on the road←0
22. Send MSG to sink about the handled events.
23. END
24. //Update the topology and responsible area again.
25. Call Patrolling Algorithm again.

that for an event, the corresponding message can be recorded by all of sensor nodes on the whole edge from one vertex to another.

2. For the same event, all sensors/RFID tags having a record of it keep consistent records. That is, all sensors/RFID tags related to the same event have the same digital pheromone all the time.
3. When a mobile robot reaches a sink, it communicates with surrounding sensors to collect the pheromone strength on each edge. According to the greedy patrolling algorithm, it will greedily choose the route with the highest pheromone to patrol.

Next, simulation results are presented. Service rate (service time) describes robot work time of handling each event. In most of simulations, the service time is chosen $f/10$, $f/20$, or $f/50$, where f denotes the digital pheromone. We use event rate to describe the speed of generating events.

4.1.1 Local Greedy Patrolling

Figure 5 shows percentage of handled events versus event rate under three scenarios with three service rates. The simulation finishes when 100,000 events occur. Figure 5 shows that a quick service can handle more events, while high handled percentage is related to a low event rate. But, when the service rate is very low (e.g., $f/10$), the influence of the event rate is not obvious.

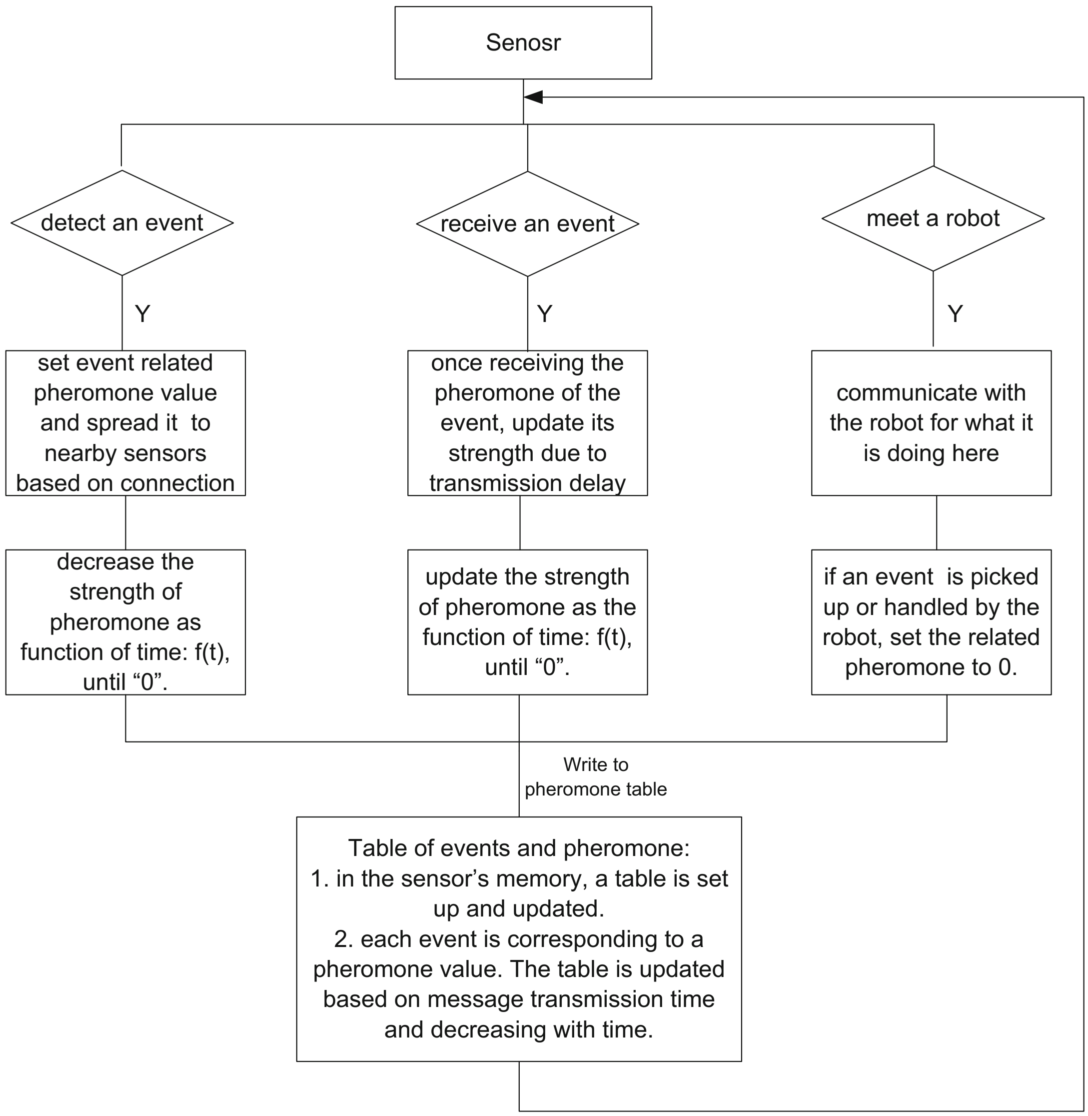
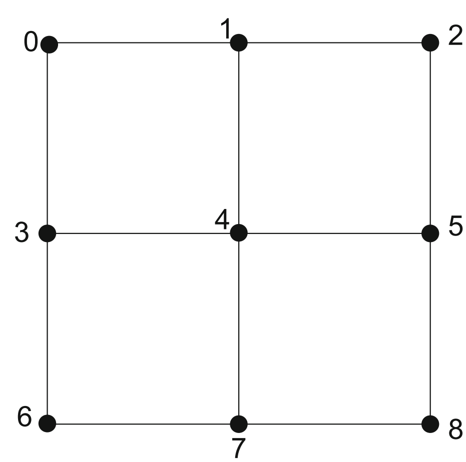


Fig. 3 Responsibilities of sensors

Fig. 4 A simply topology



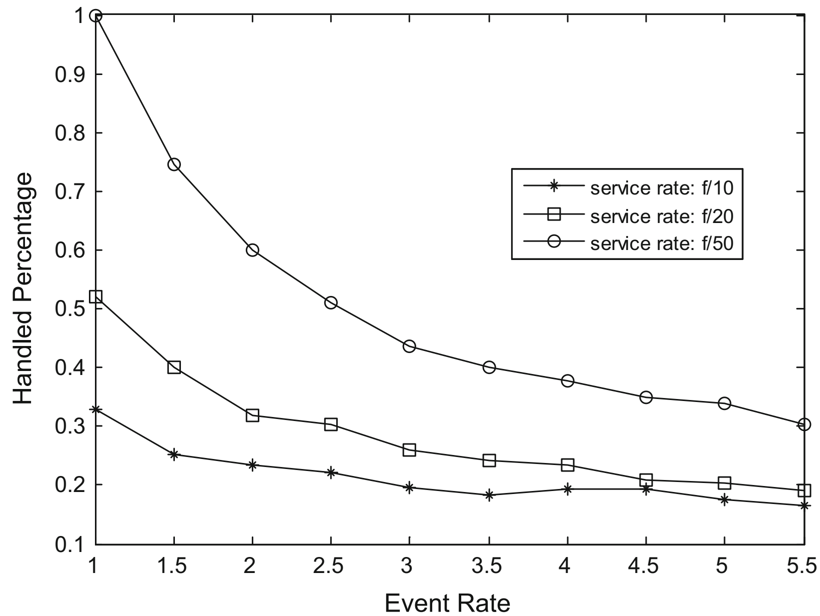


Fig. 5 Handled percentage versus event rate (when # of events = 100,000)

Figure 6 shows the performance of collected pheromones compared to event rate. Collected pheromone also decreases with the event rate.

Figures 7, 8 and 9 demonstrate the simulation results when considering the evaporation model of digital pheromones under three scenarios with three service rates. The evaporation speed is 0.5/s, that is, $f(t) = f_0 - 0.5*t$. Figure 7 shows the percentage of handled events compared to event rate. According to the figure, when events are handled quickly more events are handled. Meanwhile when events are slowly generated, the handled percentage is also larger. The results are reasonable. When events are being generated quickly, more events will become dead early. There would not be enough time for robots to handle a lot of events.

Figure 8 shows the performance collected pheromones. From the figure, we can learn that collected pheromones linearly increase with the events rate. Furthermore, with a quicker rate, more pheromones are collected.

As digital pheromones evaporate over time, events may become dead before a robot finally arrived. Figure 9 shows the percentage of dead events compared to event rate. Figure 9 shows that fewer events are cleared when events are handled quickly. When events are generated quickly, events dead percentage is high. The primary reason is that there would not be enough time for robots to handle events.

Figures 10 and 11 demonstrate the comparison of evaporation model of digital pheromones and constant model of digital pheromones under two service rates. The evaporating speed is 0.5/s, i.e., $f(t) = f_0 - 0.5*t$. Figure 10 shows the performance of collected pheromones. Figure 11 shows the performance of handled events. According to the figures, with evaporation model, more events are handled and more pheromones are collected. The reason is that with evaporation model, less time is spent on each event. More events could then be handled and more pheromones are collected.

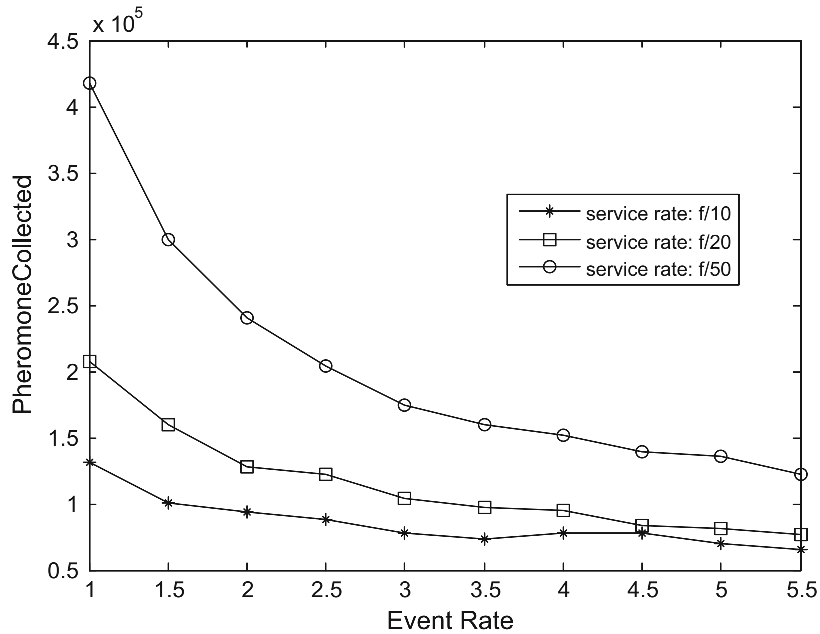


Fig. 6 Pheromone collected versus event rate (when # of events = 100,000)

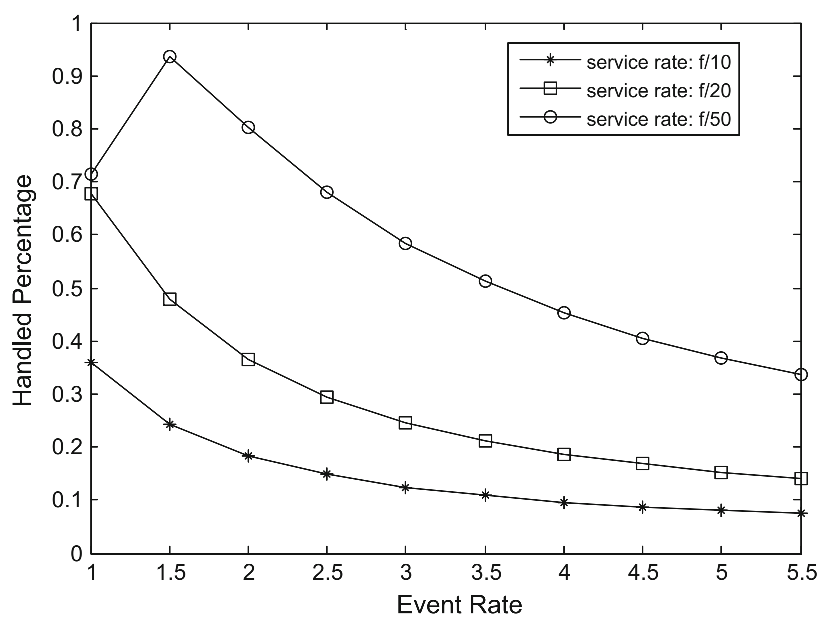


Fig. 7 Handled percentage versus event rate (when # of events = 100,000)

4.1.2 Global Patrolling Algorithm

In this subsection, we study the performance of global greedy patrolling algorithm. In our simulation, two steps of patrolling paths are considered.

Figures 12 and 13 demonstrate the simulation results with constant model of digital pheromones. Three experiments are conducted with three service rates. Figure 12 shows the performance of the total collected pheromones. According to the figure, collected pheromones decrease with event rate. Moreover, a quicker service (f/50) is corresponding

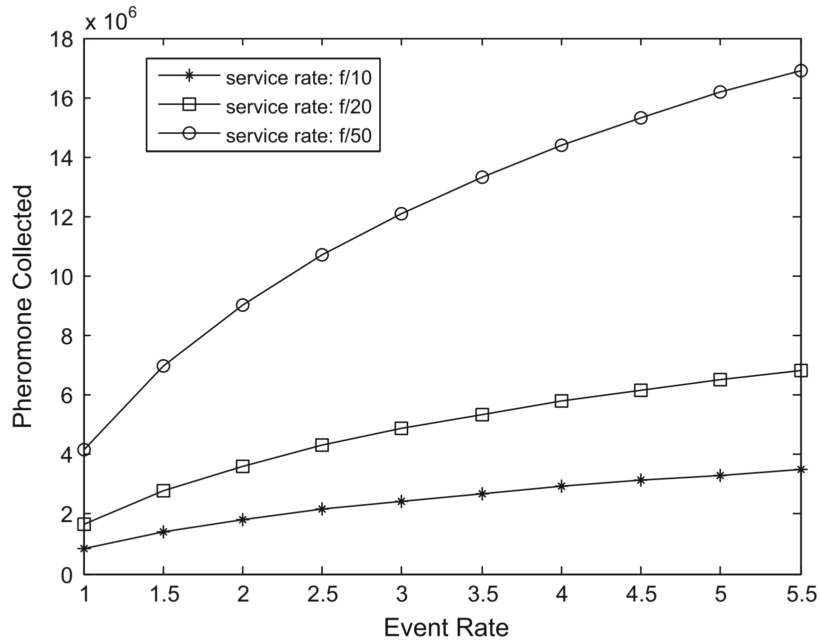


Fig. 8 Pheromone collected versus event rate (when # of events = 100,000)

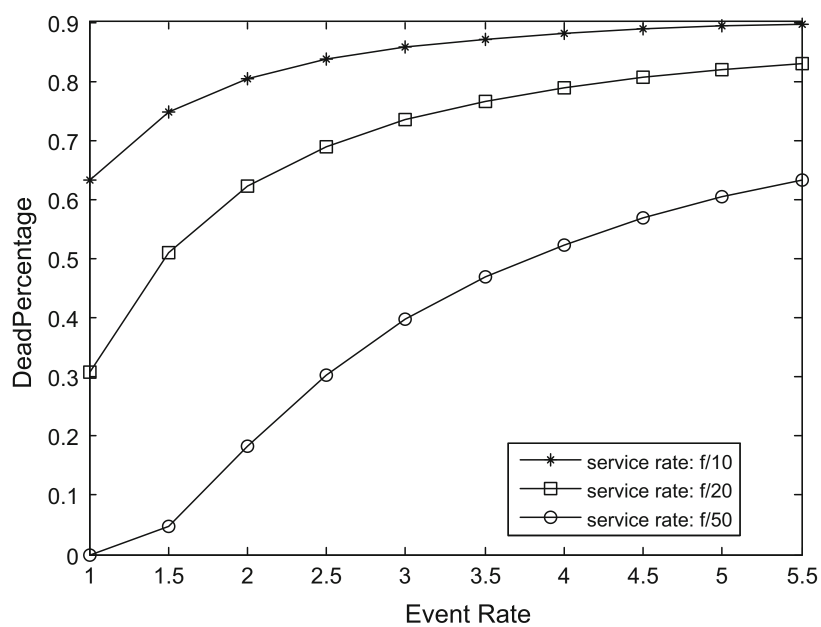


Fig. 9 Dead event rate versus event rate with the evaporation model of digital pheromone

to a larger collected pheromone value. Figure 13 indicates the performance of handled events. According to the figure, handled events do not regularly change either according to event rates or service rates.

Figures 14 and 16 demonstrate the simulation results with evaporation model of digital pheromones under three service rates. The descending speed is 0.5/s, that is, $f(t) = f_0 - 0.5*t$. Figure 14 shows the collected pheromones compared to event rate. From the figure, we can learn that collected pheromones linearly increase with event rate. Furthermore, with a higher service rate, more pheromones are collected.

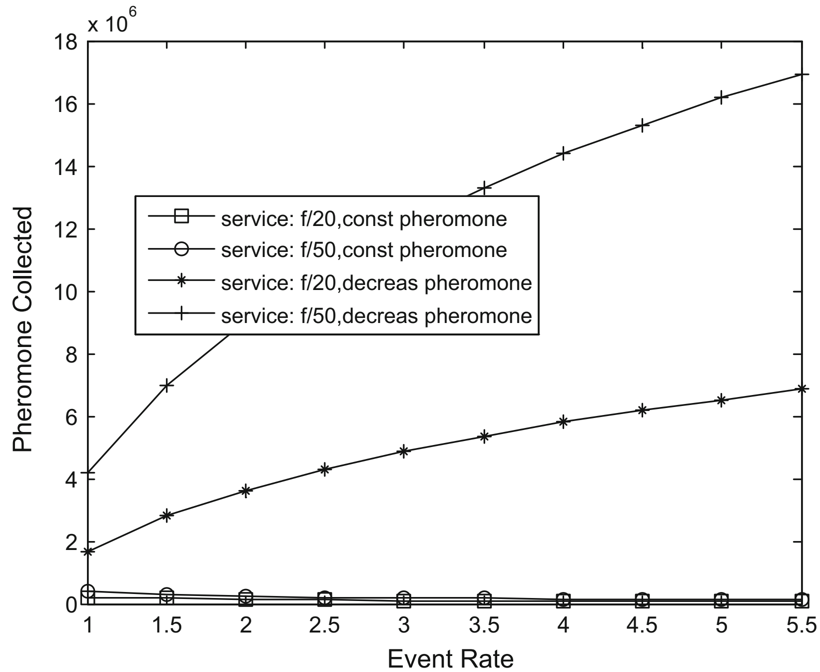


Fig. 10 Pheromone collected versus event rate

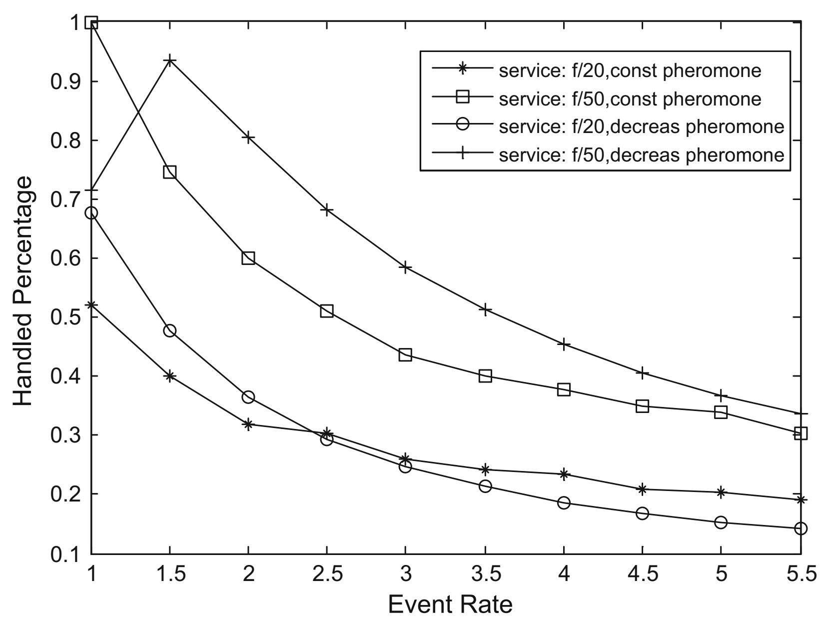


Fig. 11 Handled events versus event rate

Figure 15 shows performance of handled events versus event rate, and indicates that the percentage of events is higher when the event rate is lager. However, the percentage does not linearly increase or decrease with event rates. Figure 16 shows percentage of dead events versus event rate, and indicates that the dead event percentage is lower when the event rate is larger. However, the number of dead events does not obviously change with the event rate.

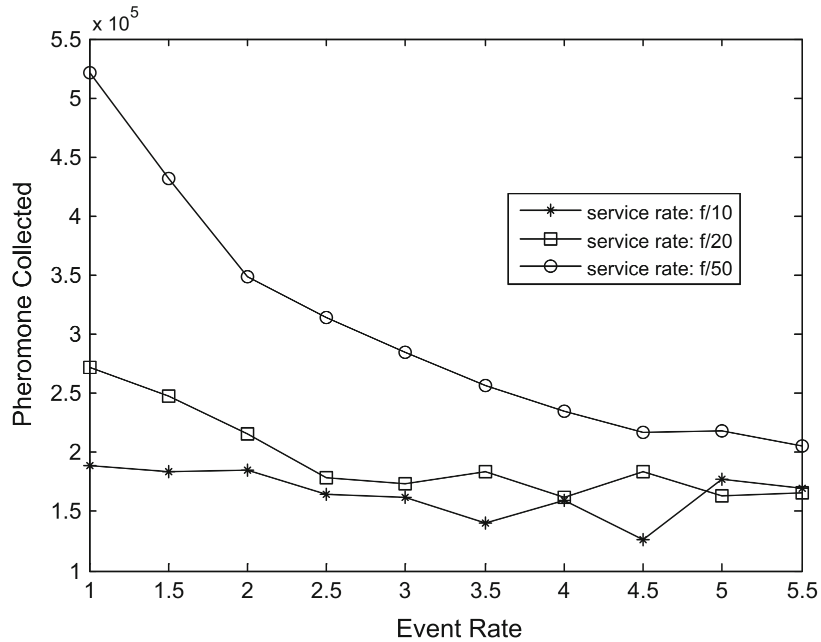


Fig. 12 Pheromone collected versus event rate

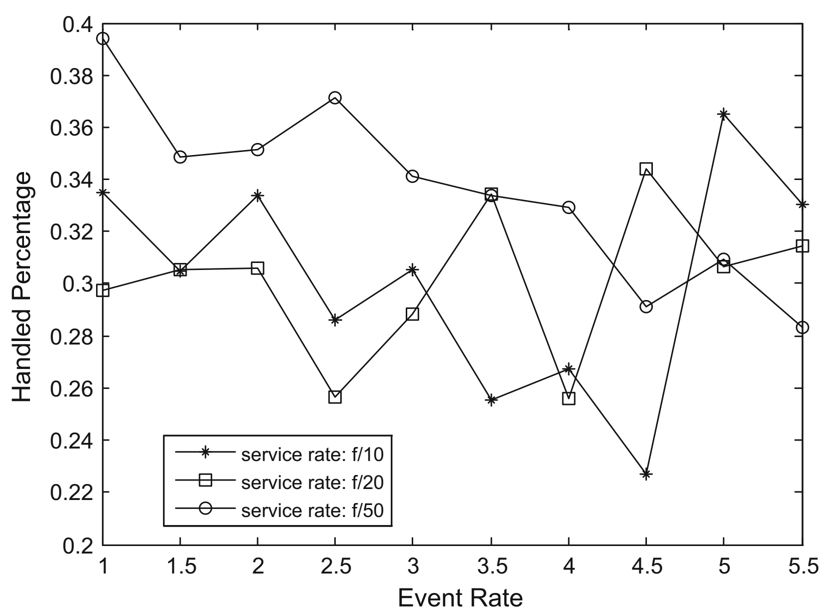


Fig. 13 Handled events versus event rate

Figures 17 and 18 demonstrate the comparison of evaporation model of digital pheromones and constant model of digital pheromones. The pheromone is evaporating with a speed of 0.5/s, that is $f(t) = f_0 - 0.5*t$. Two service rates are considered: f/20 and f/50. Figure 17 demonstrates the performance of collected pheromones. Figure 18 shows the performance of handled events. Both figures demonstrate that with the evaporation model, better performance is achieved. The reason should be that with descending pheromones, less time is required to handle each event. More events could then be handled and more pheromones are collected.

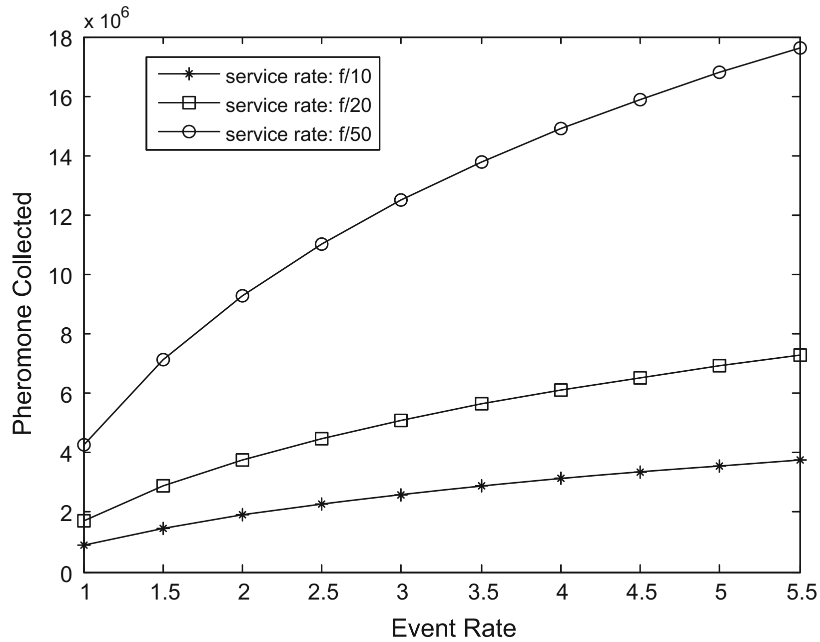


Fig. 14 Pheromone collected versus event rate

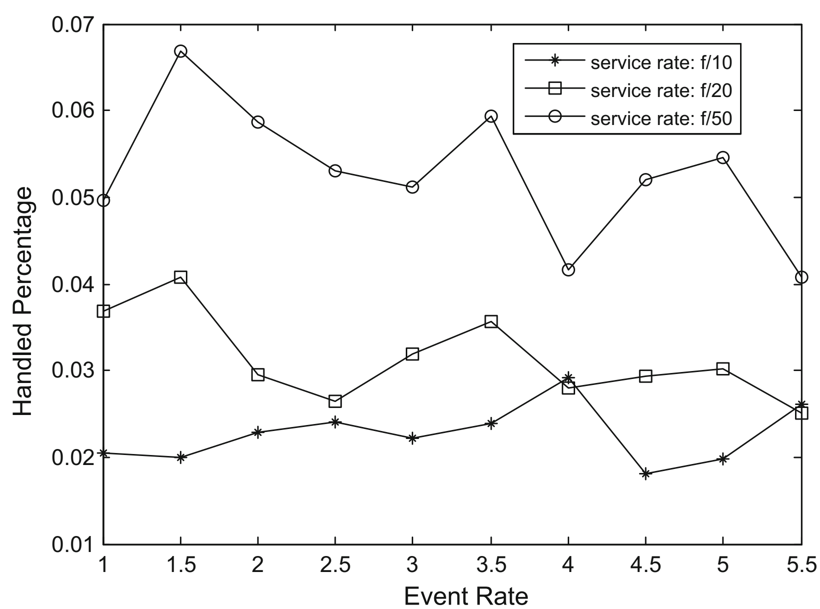


Fig. 15 Handled events versus event rate

4.1.3 Global Greedy Patrolling Versus Local Greedy Patrolling

Figures 19, 20, 21, 22 and 23 compare global greedy patrolling algorithm and local greedy patrolling algorithm. Figures 19 and 20 demonstrate the simulation results with constant pheromone model. According to Fig. 19, more pheromones are collected with global greedy patrolling algorithm. However, in Fig. 20, more events are handled with local greedy patrolling algorithm. The results are reasonable because when applying global

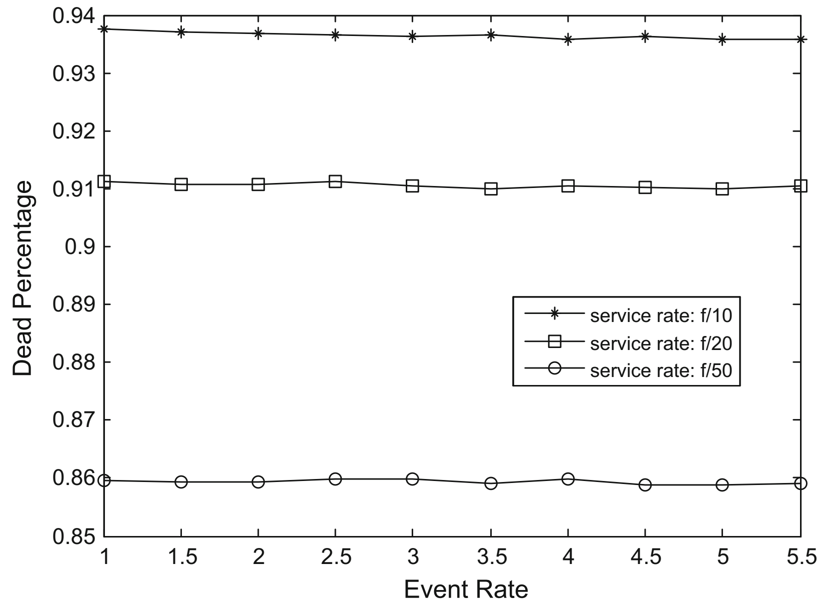


Fig. 16 Dead events versus event rate

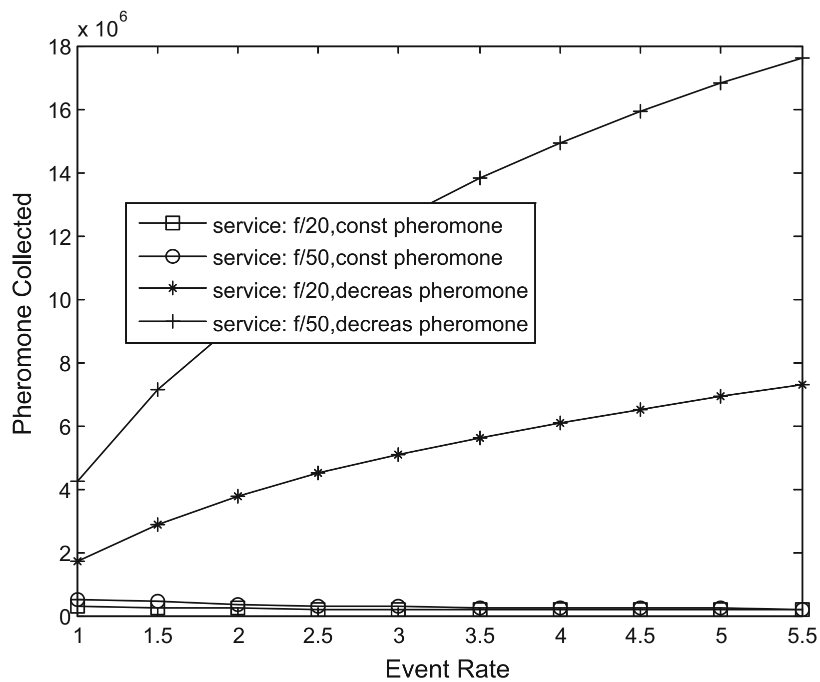


Fig. 17 Pheromone collected versus event rate

greedy patrolling, robots always choose events with a larger pheromone value, which correspondingly cost more time. Then, the total number of handled events is less than the number when applying local greedy patrolling algorithm.

Figures 21, 22, and 23 demonstrate the simulation results with evaporation model of digital pheromone. According to Fig. 21, more pheromones are collected when applying global greedy patrolling algorithm, although the simulation results are very close. Figures 22 and 23 show the

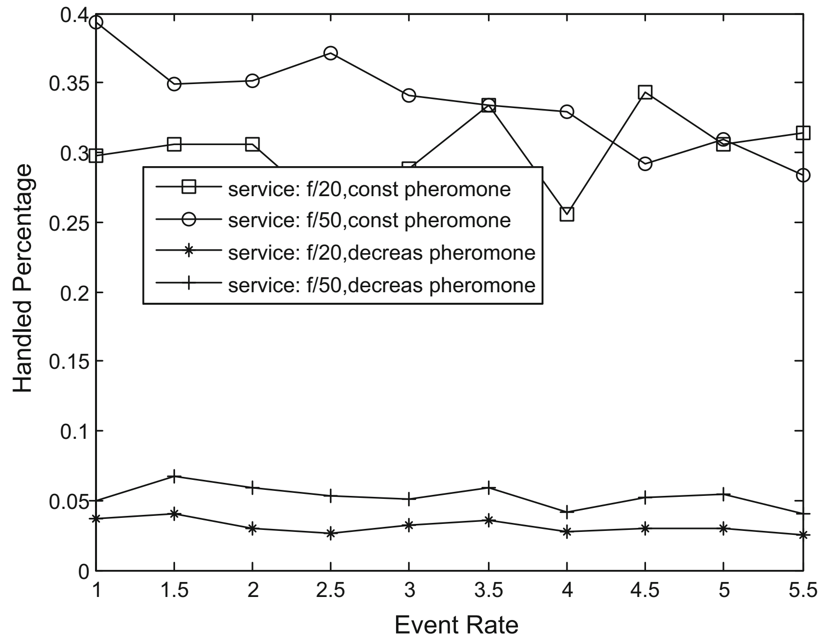


Fig. 18 Pheromone collected versus event rate

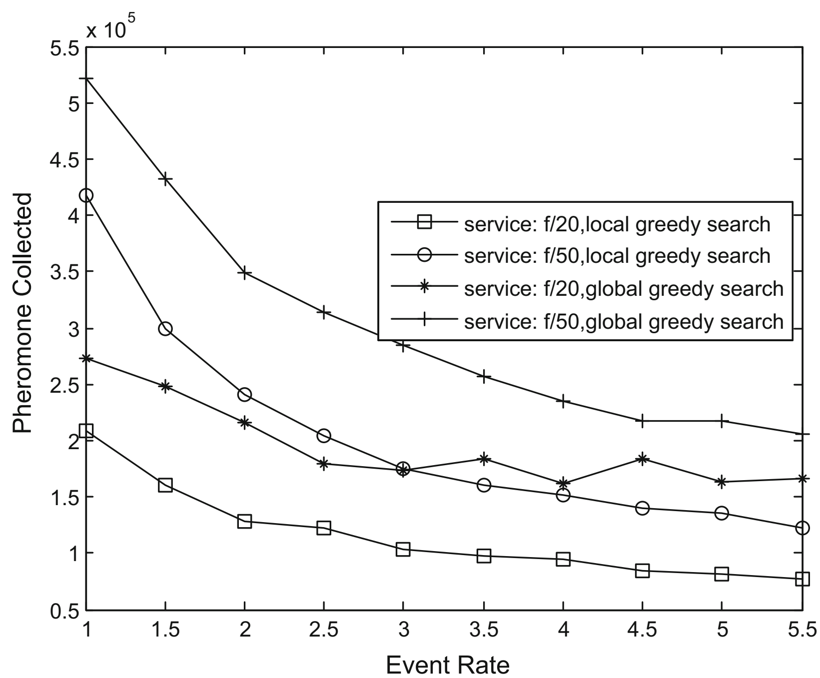


Fig. 19 Pheromone collected versus event rate

performances of handled events and dead events. More events are handled when applying local greedy patrolling algorithm. The results are similar as in Fig. 20. Correspondingly, dead events percentage is higher when applying global patrolling algorithm.

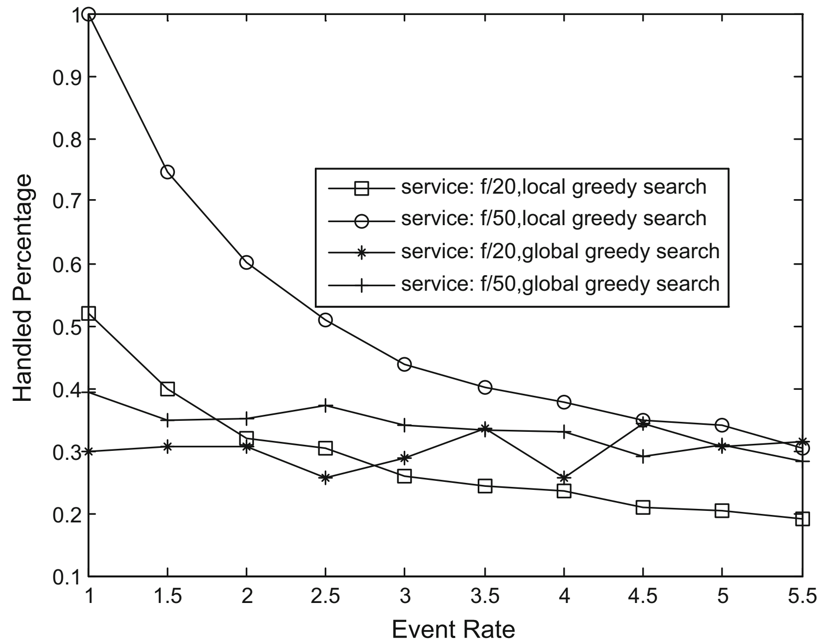


Fig. 20 Handled events versus event rate

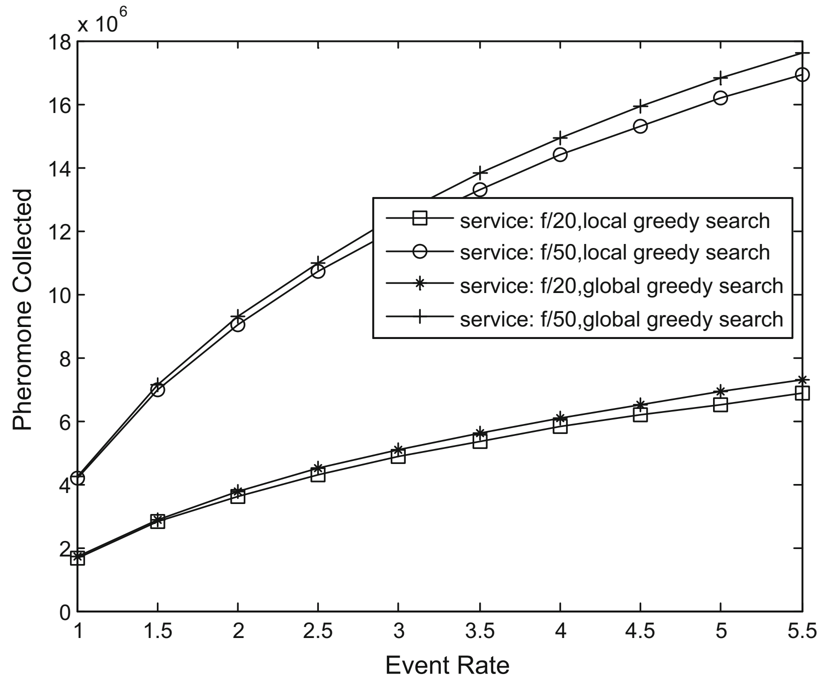


Fig. 21 Pheromone collected versus event rate

4.1.4 Comparison with Traditional Patrolling Algorithm

In order to better study the performance of our patrolling design. We also compare our algorithm with some traditional design without biological considerations. As most designs are different with our design, we have to tailor their algorithm to our problem and simulation design. We design the simulation in a more flexible topology which is randomly

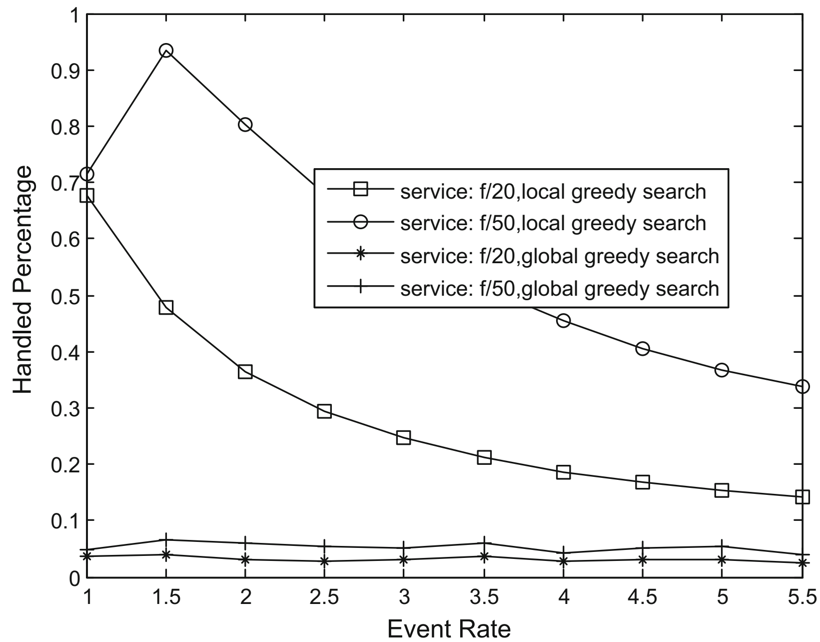


Fig. 22 Handled events versus event rate

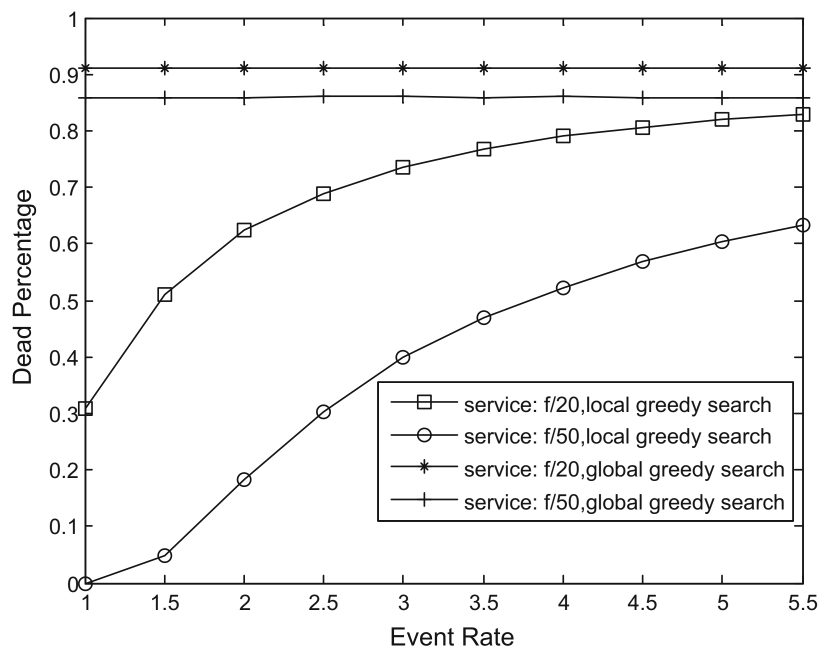


Fig. 23 Dead events versus event rate

generated instead of the simple topology as shown in Fig. 4. Figures 24, 25 and 26 compare our digital pheromone based patrolling algorithm and a traditional patrolling algorithm, through which robots patrol randomly in the monitored field. The performances are evaluated according to the mean value of generating events. Events are generated randomly in both time and locations, which is following Gaussian distribution. Figures 24 and 25 demonstrate the simulation results of dead events and handled events. According to the figs, more events are handled with our digital pheromone based algorithms. Figure 26

Fig. 24 Dead evens versus event mean value

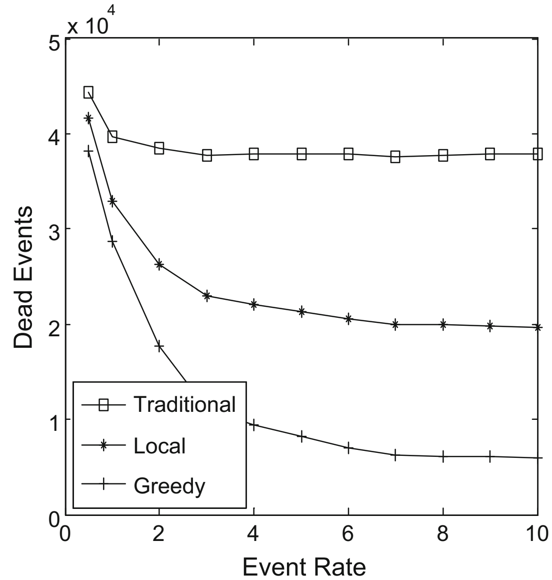


Fig. 25 Handled evens versus event mean value

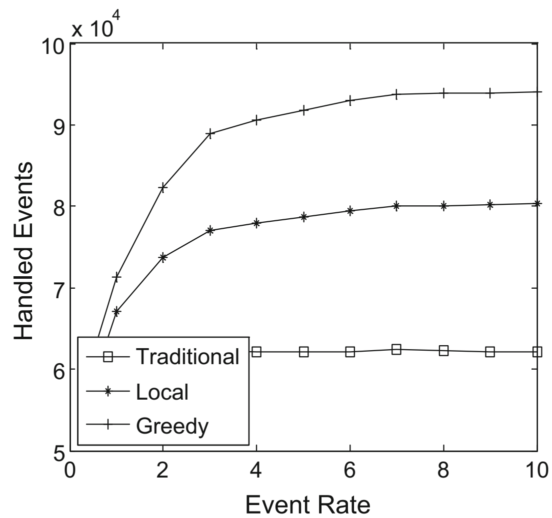
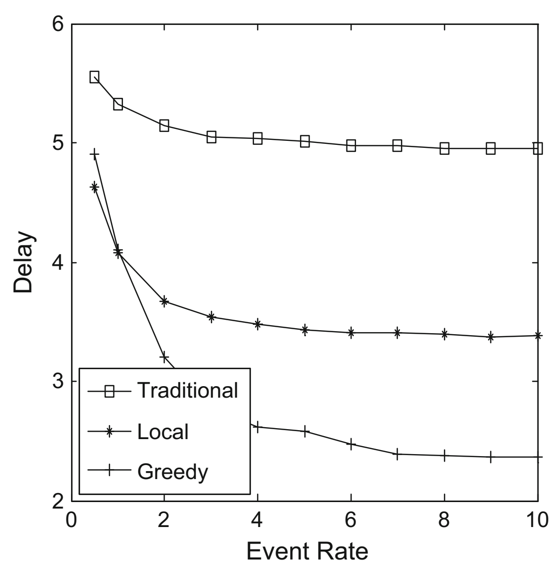


Fig. 26 Delay versus event mean value



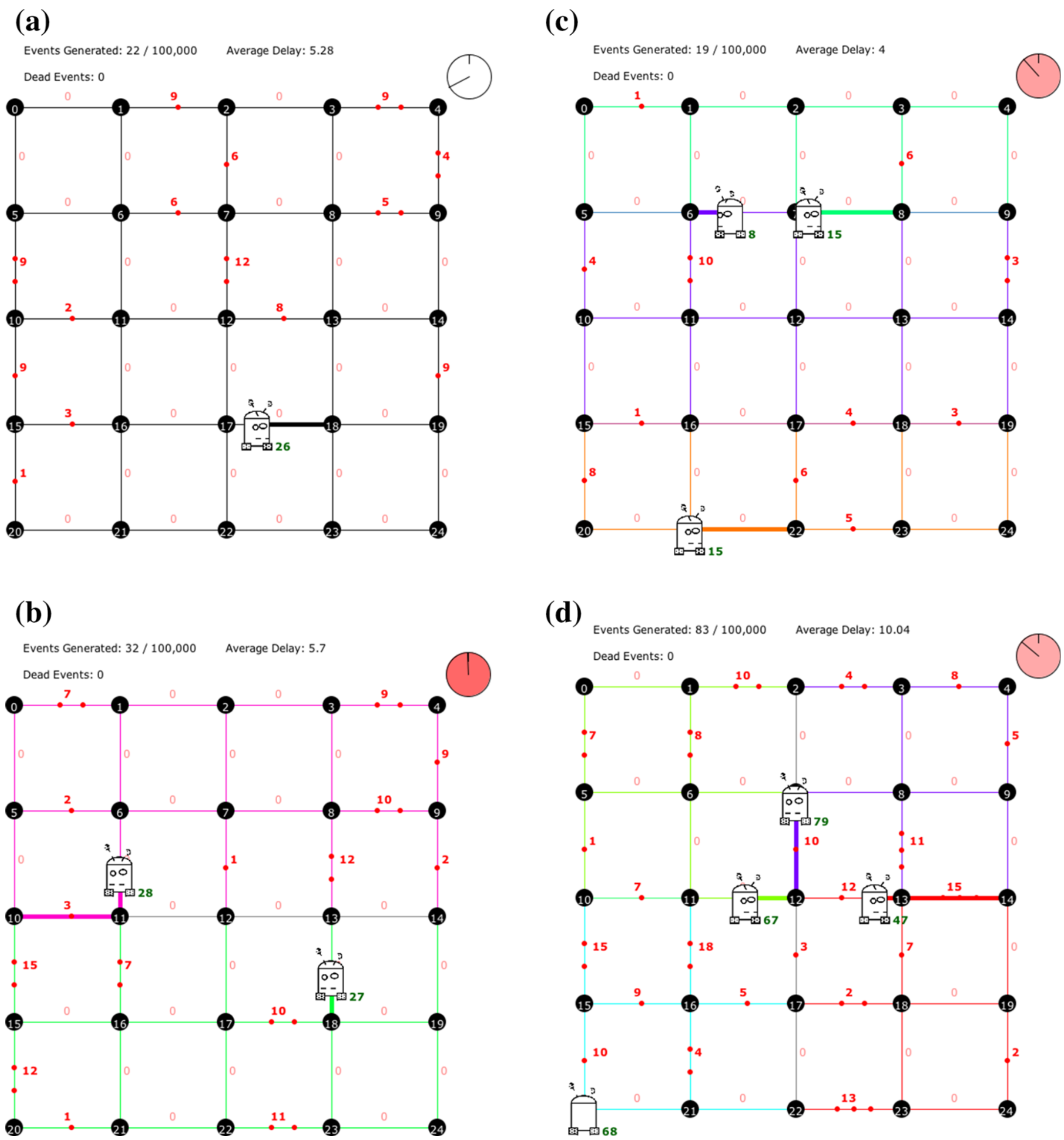


Fig. 27 Simulation topology with multiple robots. **a** One robot with the whole area as its responsible area. **b** Two robots with each one's responsible area and sharing a boundary. **c** Three robots with each one's responsible area and sharing boundaries. **d** Four robots with each one's responsible area and sharing boundaries

shows the performance of delay. The simulation results show that our digital pheromone based patrolling algorithms demonstrate better performance than classic patrolling algorithm.

4.2 Simulation for Collaborative Patrolling Algorithms

Multiple robot collaborations are studied in this subsection. Each robot is assigned a responsible area as described in Fig. 2b. We use a more complicated topology for our simulation and keep the assumptions described in subsection A. Figure 27 shows our simulation designs with one, two, three, or four robot(s), with their responsible areas shown

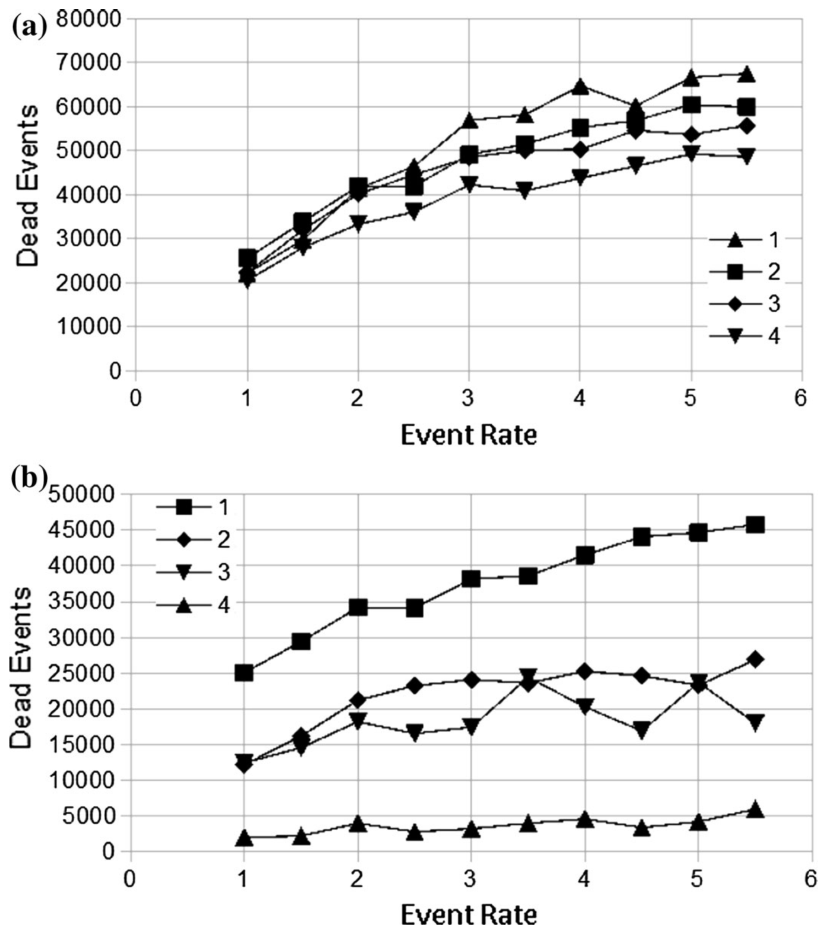


Fig. 28 Number of dead events versus event rate, with 1, 2, 3, and 4 robot(s). **a** Local greedy algorithm. **b** Global greedy algorithm with 2 steps for each path calculation

in each figure. For example, in Fig. 27b, two robots are patrolling in the field. Two different colors identify the responsible area of each robot and a middle boundary is shared between these two robots. Each sink node is identified with a number. Numbers on edges denotes current digital pheromone value on it.

There are multiple parameters involved in each Simulation design. Events are generated at an exponential rate. Exponential event generation is based on an exponential distribution with mean value ranging from 1 to 5.5. When an event is generated, the pheromone value (or priority value) of that event is randomly chosen from 1 to 10.

The pheromone value of an event will either remain constant or evaporate over time. We focus on evaporation model of events. In evaporation model, the pheromone value decreases by 1 every two time units.

For each simulation, all robots will share a service rate, meaning the speed at which a robot deals with events in the network. The service rate decides the speed at which an agent traverses an edge. The traversal speed is directly proportional to the value of an edge. For our simulations, the service rates vary from f/2, f/4, and f/8, where f denotes the value of digital pheromone on an edge. For example, if the service rate is f/2, and the edge has a value of 15, then the robot will spend 7.5 time units traversing that edge.

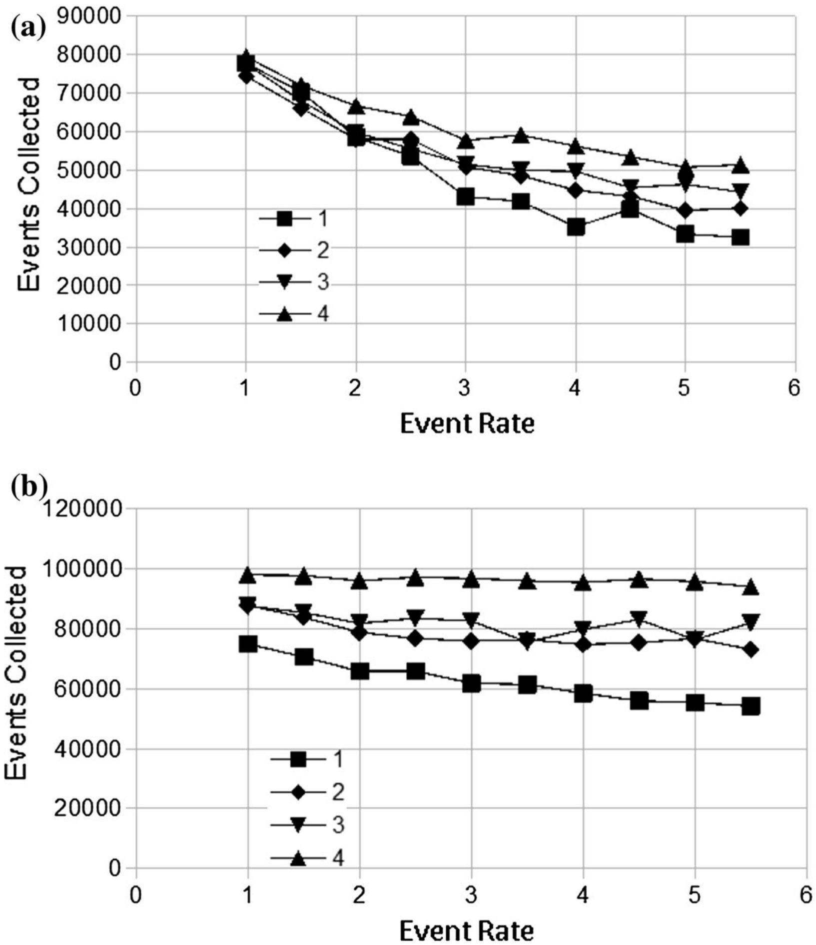


Fig. 29 Number of events collected versus event rate, with 1, 2, 3, and 4 robot(s). **a** Local greedy algorithm. **b** Global greedy algorithm: 2 steps

An edge with digital pheromone value 0 has no active events on it. When a robot comes to such an edge, it will 'idle' across the edge. Idle time is equal to the traversal time of an edge, a constant value of 1 time-unit.

We have four topology scenarios with different numbers of robots, from 1 to 4. Both local greedy algorithm and global greedy algorithm are studied. A simulation will end after 100,000 events have been generated. Multiple repeats with the same setting of parameters are collected to get a more general result.

Simulation results are demonstrated as follows with service times $f/2$, $f/4$, or $f/8$, where f is the digital pheromone. We still use event rate to describe the speed of generating events. A simulation will end after 100,000 events have been generated.

Since digital pheromone evaporates over time, if no robot stops by before it decreases to zero, we believe the event is dead and all sensors will clear the record of the event. Figure 28 shows the performance of dead events with different event generation rates, or the mean value of the exponential distribution.

As the mean of event generation increases, the number of dead events increases proportionally to this. This can be clearly seen in the graph of dead events versus mean with local greedy algorithm, as in Fig. 28a. The number of robots has a small improvement in the number of dead events. Naturally, more agents result in fewer dead events. This improvement can be better seen in the case of global greedy algorithm

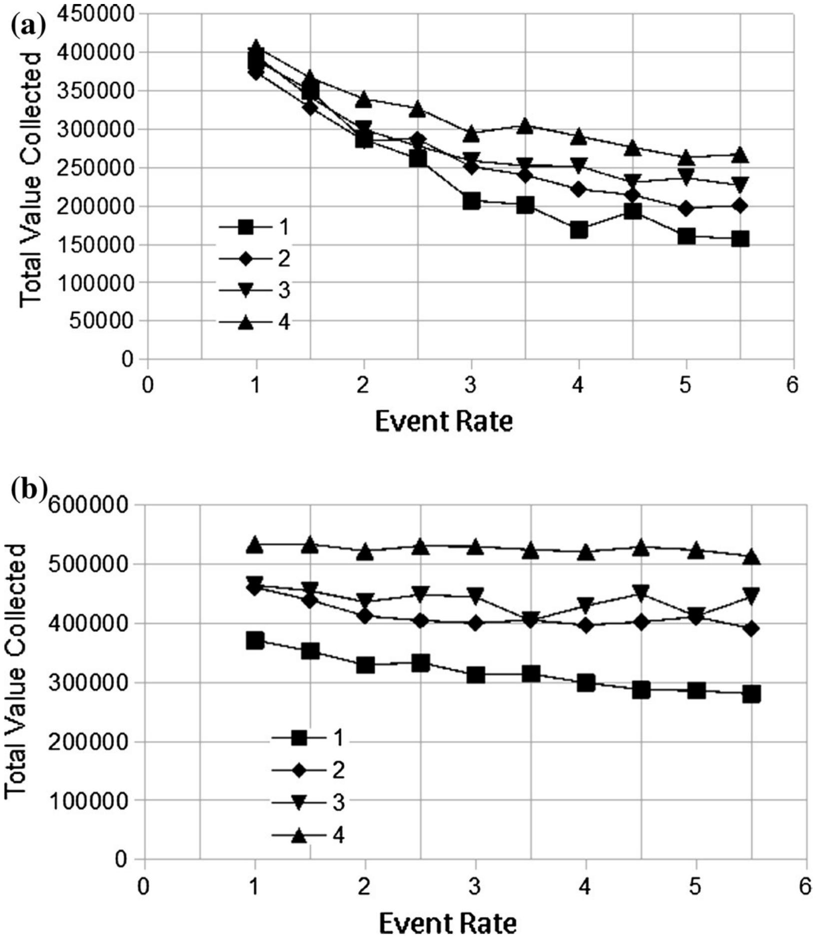


Fig. 30 Total value collected versus event rate, with 1, 2, 3, and 4 robot(s). **a** Local greedy algorithm. **b** Global greedy algorithm: 2 steps

with two steps look ahead. Because agents are able to look farther ahead, they can coordinate to collect more important events and avoid traversing the same edges, which results in fewer dead events.

We care about events collection, since it is the key task of robots' patrol. Figure 29 demonstrates the performance of collected events according to the events generation rate.

In local greedy algorithm, the number of events collected tends to decrease as the event generation rate increases. The number of collected events decreases because when events are generated faster, more of them die before collected. Robots do not have enough time to deal with so many active events at the same time. More robots help with the performance, since each of them has their responsible area and collaboration helps achieve more events collection. Meanwhile, global greedy search helps collect more events, since it enables robots to find optimal paths covering more important events.

We also study the performance of total digital pheromone collected, which is shown in Fig. 30. The number of robots help improve the performance of total pheromone collected. Meanwhile, total pheromone collected tends to decrease with the events generation rate. Since our simulation stops when there are 100,000 events, a faster generation of events leads to shorter simulation time, which is why less events collected.

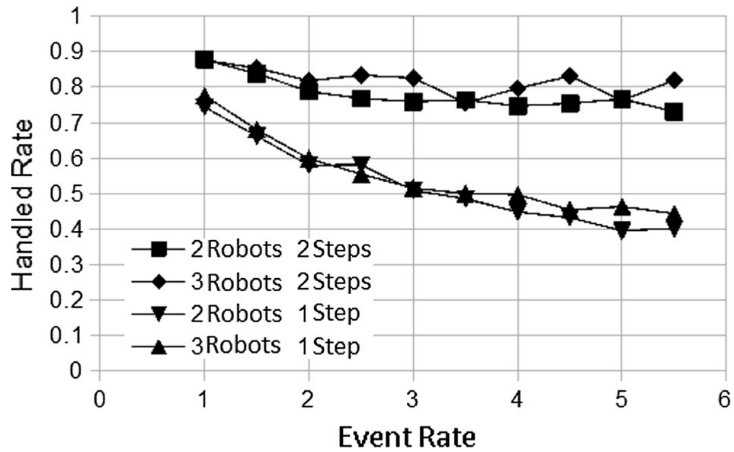


Fig. 31 Handled rate versus event rate

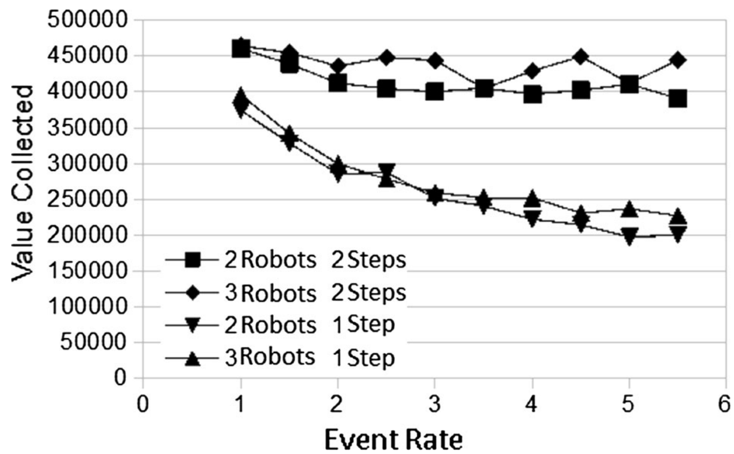


Fig. 32 Total pheromone value collected versus event rate

Figures 31 and 32 compare the local greedy algorithm and the global greedy algorithm. Handled rate is the percentage of events that have been collected, excluding all dead events.

As the event generation increases, the handled rate tends to decrease. This decrease in handled rate can be attributed to the fact that robots are unable to traverse the edges fast enough to handle the increased number of events. They do not have enough time to handle so many active events. The global greedy algorithm looks two step ahead and results in a higher handled rate because the robots are able to find more efficient paths.

Similarly, more digital pheromone is collected with global greedy algorithm, which provides patrol path with more important events. The number of robots also impacts the performance of digital pheromone collection. While it may appear that the efficiency of the robots can be increased by increasing the number of look-ahead steps, the computational difficulty grows exponentially with every additional step. For this reason, 'greedy' algorithms have new challenges at higher look-ahead numbers.

5 Conclusion

In this paper, we have designed a digital pheromone-based patrol schemes for mobile robots, which choose patrolling routes with more important events to efficiently handle events.

We simulate the impact of event rates and the robot's handling time. Two models of digital pheromones are considered based on whether to consider the evaporation of digital pheromones. That is, the priority of an event decreases with time.

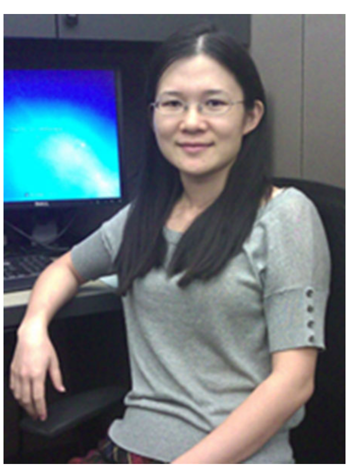
Two patrolling algorithms are studied: greedy patrol algorithm and collaborative patrol algorithm. Multiple robots are considered in our simulation design. We study the performance of different metrics with a signal robot or multiple robots. More robots will achieve better patrol and monitoring performance. Global greedy algorithm offers more obvious improvement compared to local greedy algorithm. Meanwhile, our simulation results also demonstrate the advantage of our proposed greedy patrolling algorithms compared to traditional patrolling algorithms. Our simulations show how certain parameters affect the outcome of robot efficiency. Future work may involve searching for more efficient search algorithms. Besides implementing search algorithms that operate independently on each robot, centralized and decentralized algorithms could be written to create communication between the agents, results in a higher overall system efficiency.

Acknowledgments This work was supported in part by the US National Science Foundation (NSF) under grants CNS-0737325, CNS-0716211, CCF-0829827, and CNS-1059265.

References

1. Wang, L., & Xiao, Y. (2006). A survey of energy-efficient scheduling mechanisms in sensor networks. *ACM/Springer Mobile Networks and Applications (MONET), 11*(5), 723–740.
2. Liu, C., Wu, K., Xiao, Y., & Sun, B. (2006). Random coverage with guaranteed connectivity: Joint scheduling for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems, 17*(6), 562–575.
3. Wu, K., Gao, Y., Li, F., & Xiao, Y. (2005). Lightweight deployment-aware scheduling for wireless sensor networks. *ACM/Springer Mobile Networks and Applications (MONET) Journal, 10*(6), 837–852.
4. Ozdemir, S., & Xiao, Y. (2009). Secure data aggregation in wireless sensor networks: A comprehensive overview. *Computer Networks, 53*(12), 2022–2037.
5. Xiao, Y., & Zhang, Y. (2011). Divide- and conquer-based surveillance framework using robots, sensor nodes, and RFID tags. *Wireless Communications and Mobile Computing (WCMC), 11*(7), 964–979.
6. Xiao, Y., & Zhang, Y. (2013). A critical line based boundary surveillance strategy in wireless sensor networks. *Telecommunication Systems Journal, 52*(2), 423–434.
7. Xiao, Y., Zhang, Y., & Liang, X. (2011). Primate-inspired communication methods for mobile and static sensors and RFID tags. *ACM Transactions on Autonomous and Adaptive Systems, 6*(4), 26. doi:10.1145/2019591.2019595.
8. History of US Drones: The rise of the predator empire: Tracing the history of US Drones. https://understandingempire.wordpress.com/2-0-a-brief-history-of-u-s-drones/.
9. http://www.lyftvsuber.com/.
10. He, J., & Xu, Z. (2013). Authentication and search mechanism for diffusing RFID-sensor networks. *International Journal of Sensor Networks, 14*(4), 211–217.
11. Zhang, Y., Xiao, Y., & Bales, K. L. (2009). Primate social systems, scent-marking, and their applications in mobile and static sensor networks. *International Journal of Sensor Networks, 5*(4), 210–222.
12. Zhang, Y., & Xiao, Y. (2009). Primate-inspired scent marking for mobile and static sensors and RFID tags. In *Proceedings of 18th international conference on computer communications and networks, 2009 (ICCCN 2009)*.
13. Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the traveling salesman problem. *BioSystems, 43*, 73–81.
14. Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. Cambridge: MIT Press.
15. Hinchey, M. G., Rash, J. L., Truszkowski, W. F., Rouff, C. A., & Sterritt, R. (2005). Autonomous and autonomic swarms. *Proceedings of the autonomic and autonomous space exploration systems (A&A-*

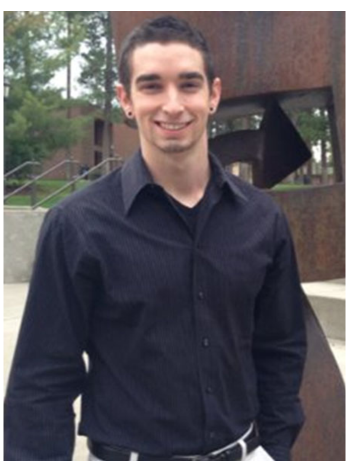
SES-1) at 2005 international conference software engineering research and practice (SERP'05) (pp. 36–42). Las Vegas, NV: CREA Press.
16. Marwaha, S., Indulska, J., & Portmann, M. (2009). Biologically inspired ant-based routing in mobile ad hoc networks (MANET): A survey. In The sixth international conference on ubiquitous intelligence and computing.
17. Liang, Y., Cao, J., Zhang, L., Wang, R., & Pan, Q. (2010). A biologically inspired sensor wakeup control method for wireless sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews, 4*(5), 525–538.
18. Ota, K., Dong, M., Wang, J., Guo, S., Cheng, Z., & Guo, M. (2010). Dynamic itinerary planning for mobile agents with a content-specific approach in wireless sensor networks. In *Proceedings of 2010 IEEE 72nd vehicular technology conference*.
19. Chawathe, S. (2007). Organizing hot-spot police patrol routes. In *Proceedings of IEEE 2007 intelligence and security informatics*.
20. Chang, C., Lin, C., Hsieh, C., & Ho, Y. (2011). Patrolling mechanisms for disconnected targets in wireless mobile data mules networks. In *The 2011 IEEE international conference on parallel processing (IEEE ICPP)*, Taiwan.
21. Liu, Z., Kwiatkowska M.Z., & Constantinou, C. (2005). A biologically inspired congestion control routing algorithm for MANETs. In *Third IEEE international conference on pervasive computing and communications workshops, 2005. PerCom 2005 workshops* (pp. 226–231).
22. Liu, J., Xiao, Y., Hao, Q., & Ghaboosi, K. (2009). Bio-inspired visual attention in agile sensing for target detection. *International Journal of Sensor Networks, 5*(2), 98–111.
23. Ling-Dong, S., Ming-Yue, Z., & Qi-Lin, M. (2015). Target tracking of binary wireless sensor networks in the domain of medicine and healthcare. *International Journal of Sensor Networks, 17*(3), 188–194.
24. Shi, L., & Tan, J. (2014). Two-tier target tracking framework in distributed sensor networks. *International Journal of Sensor Networks, 16*(1), 32–40.
25. Huang, Y., Liang, W., Yu, H.-B., & Xiao, Y. (2008). Target tracking based on a distributed particle filter in underwater sensor networks. *Wireless Communications and Mobile Computing (WCMC), 8*(8), 1011–1022.
26. Grau, D., Zeng, L., & Xiao, Y. (2012). Automatically tracking engineered components through shipping and receiving processes with passive identification technologies. *Automation in Construction, 28*, 36–44.
27. Xie, S., & Wang, Y. (2014). Construction of tree network with limited delivery latency in homogeneous wireless sensor networks. *Wireless Personal Communications, 78*(1), 231–246.
28. Shen, J., Tan, H., Wang, J., Wang, J., & Lee, S. (2015). A novel routing protocol providing good transmission reliability in underwater sensor networks. *Journal of Internet Technology, 16*(1), 171–178.
29. Fu, B., Xiao, Y., Liang, X., & Chen, C. L. P. (2015). Bio-inspired group modeling and analysis for intruder detection in mobile sensor/robotic networks. *IEEE Transactions on Cybernetics, 45*(1), 103–115.
30. Liang, X., & Xiao, Y. (2010). Studying bio-inspired coalition formation of robots for detecting intrusions using game theory. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, Special Issue on Game Theory, 40*(3), 683–693.



Yanping Zhang is an Assistant Professor in the department of Computer Science at Gonzaga University. She got the PhD in Computer Science in 2012 at the University of Alabama. Her research interests include Medium Access Control (MAC) layer optimization in wireless sensor networks (WSNs), intrusion detection in WSNs, wireless network security, protection from cybercrime, bio-computing, and telemedicine.

Yang Xiao currently is a Professor of both School of Computer and Software at Nanjing University of Information Science and Technology, Nanjing, China and Department of Computer Science at the University of Alabama, Tuscaloosa, AL, USA. His current research interests include networking and computer/network security. He has published over 200 journal papers and over 200 conference papers. Dr. Xiao was a Voting Member of IEEE 802.11 Working Group from 2001 to 2004, involving IEEE 802.11 (WIFI) standardization work. He is a Fellow of IET. He currently serves as Editor-in-Chief for International Journal of Security and Networks, International Journal of Sensor Networks, and Journal of Communications. He had (s) been an Editorial Board or Associate Editor for 17 international journals. He served (s) as a Guest Editor for over 20 times for different international journals. Dr. Xiao has delivered over 30 keynote speeches at international conferences around the world and gave more than 60 invited talks at different international institutes.

Yun Wang joined Bradley University as an assistant Professor in the Department of Computer Science and Information Systems in Fall 2011. From August 2008, she was an assistant professor in the Department of Computer Science at Southern Illinois University—Edwardsville, IL. She received the Ph.D. degree in computer science and engineering in 2008 at University of Cincinnati, Ohio. She received the B.S. degree in computer science and engineering in 2001 at Wuhan University, Hubei, China.



Patrick Mosca got his Bachelor's degree in Computer Science in 2014. He is now a software engineer at Apple Inc.