

Primate-Inspired Communication Methods for Mobile and Static Sensors and RFID Tags

YANG XIAO, YANPING ZHANG, and XIANNUAN LIANG, The University of Alabama

Although previous bio-inspired models have concentrated on invertebrates, such as ants, mammals, such as primates with higher cognitive function, are valuable for modeling the increasingly complex problems in engineering. Understanding primates' social and communication systems and applying what is learned from them to engineering domains will likely lead to solutions to a number of problems. Scent-marking is an important behavior among primates and many other mammals. In this article, inspired by primates' scent-marking activity, we propose and study a collaboration strategy for mobile and static sensors with RFID tags, where mobile sensors can be treated as robots or mobile actuators and can leave information to direct others to find them. Mobile sensors are equipped with RFID tags (or sensors) that can be deployed whenever needed, and RFID tags (or sensors) carry related information for other robots to pick up. We propose several primate-inspired communication mechanisms, including delayed-and-relayed and scent-trail communication among robots. We analytically model and simulate scent-trail communication. We also study a tracking and pursuing scheme of mobile sensors using simulations in terms of robot speeds, searching function, deployment density, turning function, and so on. We assume that robots (mobile sensors or mobile actuators) are capable of deploying/throwing-out sensors/RFID tags.

Categories and Subject Descriptors: C.2.2 [Computer-Communication Networks]: Network Protocols

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Primate-inspired communication methods, robotic, mobile and static sensors, RFID, monkeys

ACM Reference Format:

Xiao, Y., Zhang, Y., and Liang, X. 2011. Primate-inspired communication methods for mobile and static sensors and RFID tags. *ACM Trans. Auton. Adapt. Syst.* 6, 4, Article 26 (October 2011), 37 pages. DOI = 10.1145/2019591.2019595 <http://doi.acm.org/10.1145/2019591.2019595>

1. INTRODUCTION

Primates such as coppery titi monkeys engage in both vocal and chemical communications (scent-marking) in which pheromone traces are left to relay messages to other animals [Snowdon et al. 1985]. This species provides an excellent example of communication. Although previous bio-inspired models have concentrated on invertebrates, such as ants, mammals with higher cognitive function are valuable for modeling the increasingly complex problems in heterogeneous mobile and static sensor networks. In this work, we propose to study and model communication modes of the coppery titi monkeys and to apply the results to communications for collaborations in the distributed control of heterogeneous mobile and static sensor networks.

This work is supported in part by the U.S. National Science Foundation (NSF) under grant numbers CNS-0737325, CNS-0716211, CCF-0829827, and CNS-1059265.

Authors' address: Y. Xiao, Y. Zhang, and X. Liang, Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487-0290; email: yangxiao@ieee.org.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permission may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701, USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2011 ACM 1556-4665/2011/10-ART26 \$10.00

DOI 10.1145/2019591.2019595 <http://doi.acm.org/10.1145/2019591.2019595>

Animals have inspired many inventions that help human beings. Examples of such inventions include radar, which is manufactured according to the principle of how bats identify directions, airplanes, which were invented based on dragonfly wings, and submarines, which are applied in the military based on the physical structure of fish. People can always learn something practical from the physiological characteristics of animals and adopt them for their own use.

Scent-marking [Lewis 2006] is an important communication behavior of many different mammals, especially primates. Through scent-marking, primates leave information demonstrating their purpose, showing their aggression, and establishing and maintaining their social rank [Lewis 2006]. Inspired by scent-marking, we propose a collaboration strategy in mobile and static sensor networks with RFID tags.

In wireless sensor networks, two kinds of sensor nodes are important for study: static and mobile sensors. After deployment, static sensors can only passively sense and collect data from the environment [Abrams et al. 2004; Amundson et al. 2011; Arienzo and Longo 2011; Azad and Chockalingam 2011; Boubiche and Bilami 2011; Brass 2007; Gupta et al. 2003; Jaggi and Kar 2011; Jedermann et al. 2011; Kafetzoglou and Papavassiliou 2011; Krontiris and Dimitriou 2011; Li 2011; Liu et al. 2006; Lu and Suda 2007; Lu et al. 2011; Majumdar and Ward 2011; Miao et al. 2005; Morreale et al. 2011; Olteanu et al. 2010; Peng et al. 2009; Poornima and Amberker 2011; Rao and Kesidis 2004; Shakkottai et al. 2003; Shen et al. 2011; Soliman and Al-Otaibi 2011; Taniguchi et al. 2011; Tran et al. 2011; Tseng et al. 2011; Wang and Xiao 2006; Wodczak 2011; Wu et al. 2005; Xiao et al. 2010]. On the other hand, mobile sensors are resource-rich sensor nodes with more energy, higher power, and better processing capabilities. All these features give mobile sensors more capabilities, such as more computation and storage, which help to make decisions and execute appropriate actions according to the information gathered from sensors. There is a vast amount of literature on multirobot collaboration and surveillance [Dias and Stentz 2000; Simmons et al. 2000; Yamauchi 1998; Zlot et al. 2002]. A detailed survey of animal behavior on scent-marking and social behavior can be found in Zhang et al. [2009].

In our study, robots function as mobile sensors that patrol the entire network, communicate with sensors, collect information from sensors, and when necessary, leave RFID tags or sensors to convey data for other robots to pick up. For some collaboration work, robots can leave RFID tags or sensors, a behavior very similar to the communication of primates using scent-marking.

In this article, inspired by the primates' scent-marking activity, we propose primate-inspired message-based communications. We propose delayed-and-relayed and scent-trail approaches to achieve mobile and static sensor communications. Both approaches are very similar to pheromone traces left by New World monkeys. For bio-inspired message communication based on pheromonal information trails, we design ways to relay messages to static sensor nodes, including delayed-and-relayed and scent-trail communication. Robots (mobile sensors or mobile actuators) can also communicate and collaborate with each other. A single robot is only capable of achieving simple targets. Once robots meet some complicated task, they need to communicate with other robots and collaborate to complete it. In this situation, the former robots need to leave as many RFID tags as needed for other robots to catch up. When the pursuing robots meet RFID tags, they track these to find the route of the former robots. In this article, we analytically model and simulate scent-trail communication. In our analytical model, we solve the following problems.

- What is the probability that the second robot can capture the first robot?
- What is the conditional expected time that the second robot takes to capture the first robot with the condition that the second robot can capture the first robot?

- What is the conditional expected distance required for the second robot to capture the first robot?
- What is the conditional number of RFID tags that the first robot must drop in order for the second robot to be able to capture the first.

We propose and study a collaboration strategy for mobile and static sensors with RFID tags, where mobile sensors can be treated as robots or mobile actuators and can leave information to direct others to find them. Mobile sensors are equipped with RFID tags (or sensors) that can be deployed whenever needed, and RFID tags (or sensors) carry related information for other robots to pick up. We also study a tracking and pursuing scheme of mobile sensors using simulated robot speeds, searching functions, deployment densities, turning functions, and so on. We study two cases, with and without considering the lifetime of sensors and RFID tags. We study the following problems in the scent-marking scheme:

- the performance of, (a) the minimum requirement of a follower robot, the maximum requirement of the first robot, (b) the delay time for a follower robot to catch up to the first robot, and (c) the catch probability;
- the performance of delay with the speed of the follower robot;
- the performance of delay with the speed of the first robot;
- the probability that the follower robot catches up to the first robot, and so on.

The rest of the article is organized as follows. Section 2 presents some background information including a brief survey of scent-marking activities among primates, optimal forging theory versus our approaches, and other related work. Section 3 proposes primate-inspired communication methods. Section 4 gives an analytical model for the scent-trail mechanism. Section 5 presents evaluations for the analytical model of the scent-trail mechanism. Section 6 provides further analysis of primate-inspired scent-marking strategy. Sections 7 and 8 provide simulation results with and without consideration of the lifetime of sensors and RFID tags, respectively. Section 9 provides future research directions. Finally, we conclude the article in Section 10.

2. BACKGROUND

In this section, we provide some background information including a brief survey of scent-marking activities among primates, optimal forging theory vs. our approaches, and other related work.

2.1. A Brief Review of Scent-Marking of Primates

Scent-marking¹ is an important activity among primates, which is accomplished by deploying chemicals with irritating odors, such as urine, at prominent locations. In recent years, there has been a great deal of research on primate society, communication, sexuality, and so on. All these studies are closely related to the scent-marking behavior of primates. As mentioned in Lewis [2006], scent-marking of primates falls under one or more of the following five categories: territorial demarcation, ownership of resources, mate attraction, noncombative fighting, and self-advertisement. In Dixson and Brancoft [1998] and Smuts et al. [1987], primate sexuality is studied

¹<http://en.wikipedia.org/wiki/Scent-marking>

in depth The studies show that indirect communication by means of scent-marking plays the most important role in the social and sexual behaviors of primates. In Smuts et al. [1987], primate society is studied and scent-marking is shown as a sign of the boundary of a territory, of defense, of alarm, and of individual recognition. It is also proposed in Snowdon et al. [1985] that scent and scent-marking behavior appear to be important in many areas of primate life, especially in communication. Scent and scent-marking are even used to demonstrate aggressive motivation and the establishment and maintenance of high social rank.

Scent-marking [Fiorie et al. 2006] (or olfactory signaling) is commonly used among mammals for communication both within and between social groups. Scent marks are regarded as a way of conveying information not only about individual identity, group membership, and sex, but also about social rank and reproductive state [Fiorie et al. 2006]. The active compounds in these marks that convey the effective information about the signaler will remain volatile for a relatively long period [Fiorie et al. 2006]. Furthermore, the signaler is capable of sharing its information without being present.

There have already been a number of nonexclusive hypotheses concerning the functions of scent-marking in primates, most of which fall into three basic categories [Fiorie et al. 2006]. First, scent-marking functions in a variety of sociosexual and reproductive contexts, both within and between groups [Fiorie et al. 2006]. Such functions include both intersex and intrasex purposes (such as advertising a signaler's sexual status or receptivity to members of the opposite sex), advertising or reinforcing a signaler's dominance over same-sexed conspecifics, competing with same-sexed individuals for reproductive opportunities, and attracting opposite-sexed individuals [Fiorie et al. 2006]. Second, scent-marking may demonstrate the signaler's ownership, occupancy, or use of an area, and has thus been proposed as a potentially efficient means of defending a territory or home range [Fiorie et al. 2006]. Finally, it was recently suggested that scent-marking may have other functions in intragroup communication that are related to ecological information, such as aiding the orientation of group members within the home range or demonstrating the locations of food sources to group mates [Fiorie et al. 2006].

So far, most of the studies related to scent-marking in primates have been focused on the callitrichines [Fiorie et al. 2006]. Such behavior comprises a rich part of their behavioral repertoire, and some researchers believe that callitrichine scent-marking serves several of the functions discussed in the preceding in different ecological and social environments [Fiorie et al. 2006]. In the case of golden lion tamarins, their scent-marking activities are more frequently on feeding trees than nonfeeding trees and marks are more frequently deposited by dominant males than subordinate males [Fiorie et al. 2006]. This not only suggests that these signals communicate the location of important food sources to group mates, but that they also communicate male social status [Fiorie et al. 2006]. In several species of Amazonian tamarins (*Saguinus* spp.) and common marmosets (*Callithrix jacchus*) [Fiorie et al. 2006], pheromones are often detected in the scent marks of dominant females and appear to be important cues that prompt physiological suppression of reproductive cycles in subordinates [Fiorie et al. 2006].

Some studies reported that scent-marking is more frequently discovered in peripheral areas or in regions of home-range overlap than in the interior portions of the range or areas of exclusive use [Fiorie et al. 2006]. Some researchers have also found that there is no distinct difference in scent-marking rates in these various environments [Fiorie et al. 2006]. In addition to the callitrichines, scent-marking behavior in anthropoids is also a research focus [Fiorie et al. 2006]. This is true for many other remaining New World monkeys, except that many of them have well developed skin glands on the

sternum and perineum and routinely engage in a variety of scent-marking behaviors [Fiorie et al. 2006].

Within large-bodied individuals, scent-marking appears to be relatively common [Fiorie et al. 2006], although few related studies have been conducted.

2.2. Optimal Foraging Theory (OFT) vs. Our Approaches

Foraging is searching for food.² Foraging theory studies the relative payoffs of different foraging alternatives. Optimal Foraging Theory (OFT) maximizes the energy intake per unit of time. OFT normally either achieves the greatest amount of energy possible in a given environment or requires the least possible amount of time for a fixed amount of energy.³

Inspired by OFT, we have designed robot patrolling schemes to handle events in a surveillance area. Robots travel around the monitored area to handle detected events in a wireless sensor network. Each event has a priority, which is very similar to the calories contained in a prey in OFT. Similar to the foraging activities of animals, robots patrol the surveillance area to handle events. As in the different optimal considerations in the OFT problem, robots are designed to be able to choose from different patrolling paths, that may target high priority events, as large a number of events as possible, or more high priority events per unit of time. The choice of paths is based on the target of the surveillance system.

2.3. Other Related Work

Much research has focused on sensor networks in which a large number of sensor nodes is densely deployed over a region [Abrams et al. 2004; Brass 2007; Liu et al. 2006; Wu et al. 2005]. A number of studies focus on network coverage. In Brass [2007], the coverage capability of the network is studied with sensor nodes deployed regularly, or with Poisson distribution with the assumption that there are countless sensor nodes that can be deployed all over the region. Recently, the joint problem of coverage and connectivity was considered [Gupta et al. 2003; Liu et al. 2006; Shakkottai et al. 2003]. Meanwhile, many research efforts have also focused on surveillance applications for sensor networks with static and mobile sensors [Lu and Suda 2007; Miao et al. 2005; Rao and Kesidis 2004]. The deployment problem in mobile sensor networks is studied in Miao et al. [2005], including the optimal placement of mobile sensor networks and their self-organizable deployment. In Rao and Kesidis [2004], the Enhanced Differentiated Surveillance scheme is proposed, to maintain a required coverage while saving energy. In Rao and Kesidis [2004], a mobile algorithm is developed and surveillance in mobile sensor networks is studied. All of these studies consider the mobility feature of the sensor network as well as the surveillance application. There is a vast amount of literature about multirobot collaboration and surveillance [Dias and Stentz 2000; Simmons et al. 2000; Yamauchi 1998; Zlot et al. 2002]. A detailed survey of animal behaviors on scent marking and social behaviors can be found in Zhang et al. [2009]. There are other related works in Xiao and Zhang [2009, 2011a, 2011b], Zhang and Xiao [2009a, 2009b], and Xiao and Liang [2009].

²http://wikipedia.org/wiki/Optimal_foraging_theory

³<http://www.thewildclassroom.com/biodiversity/birds/aviantopics/optimalforagingtheory.html>

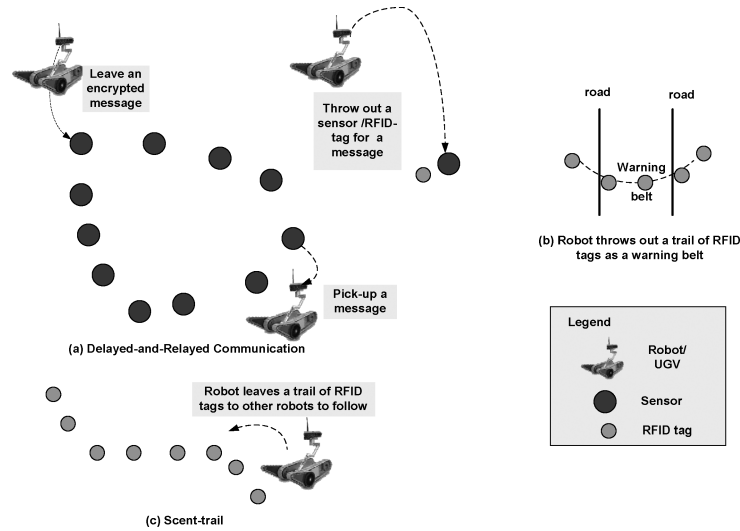


Fig. 1. Bio-inspired communications.

3. PRIMATE-INSPIRED COMMUNICATION METHODS

Bio-inspired communication mechanisms are now proposed as follows. Typically, chemicals can be left by animals on food and sleeping sites to mark scent trails. An animal can leave a scent-trail to help others locate the food source. Inspired by these scent-trails, we propose two bio-inspired communication approaches: delayed-and-relayed and scent-trail, as shown in Figures 1(a) and 1(c), respectively. Both methods are inspired by animal scent-trails. Delayed-and-relayed is used to leave a message for a sensor node or to leave a sensor node/RFID-tag to be picked up by other robots at a later time. Scent-trail leaves a trail of sensor nodes/RFID-tags to let other robots know the route of this robot, if necessary, and is used when the route of this robot cannot be known beforehand and the delayed-and-relayed approach cannot achieve its goal, and when a route/trail cannot be expressed well enough to leave it as a message. We can use sensor nodes to hold messages to be picked up at a later time. A robot can also leave one or multiple sensor nodes/RFID-tags on the ground to form a pheromone trace/trail to help relay messages in large and complex areas. If there are sensor nodes available, a trail can also be achieved by leaving messages for passing sensor nodes. Another function of robots is sending danger alerts. Once serious events, such as forest fires, chemical weapons, bombs, or other incidents, are detected, robots need to warn other robots of the danger. They deploy RFID tags to blockade the event area, as shown in Figure 1(b). If other robots patrol nearby, they will know the danger and stop to help with the blockade work. They may either wait until humans come to handle the events or for human commands.

After robots pick up a message, the related memory space is released. We assume that such messages are encrypted and that only authorized robots can decrypt these messages.

Robots may communicate with each other when they meet or are within each other's range. We assume that the surveillance area is so large that central control of robots is impossible. Robots are distributed and do their own jobs. Therefore, robots may or may not communicate with each other. In our surveillance framework, we assume robots communicate with each other using the following three methods.

(1) When two robots meet, they exchange their information and update their own information to themselves. They function as mobile sinks—each maintains partial information, which is updated through communications. (2) Robots can use these bio-inspired approaches to communicate. (3) When robots detect some emergency, such as a forest fire, chemical bomb, or intrusion tank, robots can send emergency messages to sensor nodes and sinks; then each message is broadcast to other robots in the network. However, this method is only used in emergencies. All three methods help robots exchange information, balance their loads, and update their own information.

4. ANALYTICAL MODEL FOR PRIMATE-INSPIRED SCENT-TRAIL COMMUNICATION

In this section, we propose an analytical model for the scent-trail mechanism.

We assume that R_1 and R_2 are two robots. R_2 is tracking R_1 in a road. R_1 and R_2 move at constant velocities of V_1 and V_2 , respectively. Initially, R_1 is D_0 units ahead of R_2 . R_1 is expected to drop at most N signals called RFID tags, randomly when it moves. Initially, R_1 drops the first RFID tag; we denote it as F_0 . The arbitrary distance between two adjacent RFID tags is subject to an exponential distribution with mean $1/\lambda$. R_2 captures R_1 by tracking the RFID tags. We now denote the i th RFID as F_{i-1} , $0 \leq i \leq N-1$; the distance between F_{i-1} and F_i is denoted as D_i . Here, D_1, D_2, \dots, D_{N-1} are independent and have the same exponential distribution with mean $1/\lambda$. We make two assumptions: (1) R_2 is sure to find the RFID tags; the time interval between the instant it finds F_{i-1} and the instant that it finds F_i is denoted as T_i , $1 \leq i \leq N-1$, and the time R_2 spends to find F_0 is T_0 . The expression of T_i is as follows: $T_i = G(V_2, D_i)$, if there is no turn in the path between F_{i-1} and F_i and otherwise $T_i = H(V_2, D_i)$, if there is any turn in the path between F_{i-1} and F_i . Here, we assume that the probability of there being no turn in the path between F_{i-1} and F_i is P_0 . Here G and H have the following property: $G(v_1, d) \leq d/v_2$ and $H(v_1, d) \leq d/v_2$ for any pair of v_1 and v_2 which satisfy $v_1 \geq v_2 > 0$ and any $d, d > 0$. (2) R_2 can capture R_1 if and only if R_2 can capture R_1 before R_2 finds the N th RFID tag. We will solve the following problems.

- *Problem 1.* What is the probability that R_2 captures R_1 ?
- *Problem 2.* What is the expected time that R_2 takes to capture R_1 , under the condition that R_2 can capture R_1 , where we denote the conditional expected time as T ?
- *Problem 3.* What is the expected distance R_2 travels to capture R_1 ?
- *Problem 4.* How many RFID tags will R_1 drop, under the condition that R_2 can capture R_1 ?

Before we discuss these problems, we denote the probability that R_2 can capture R_1 as P . Then the probability that R_2 can not capture R_1 is $1 - P$.

For Problem 1, we first denote the time from the beginning to the instant that R_1 drops F_i as t_i and the time from the beginning to the instant that R_2 finds F_i as \bar{T}_i . We then have

$$t_i = \frac{1}{V_1} \sum_{j=1}^i D_j, i = 1, \dots, n-1.$$

$$\bar{T}_i = \sum_{j=0}^i T_j, i = 1, \dots, n-1.$$

We can make the following statement.

$$\begin{array}{l}
 R_2 \text{ can capture } R_1 \text{ if and only if} \\
 \left\{ \begin{array}{l}
 R_2 \text{ can capture } R_1 \text{ between } F_0 \text{ and } F_1; \text{ or} \\
 R_2 \text{ can capture } R_1 \text{ between } F_1 \text{ and } F_2, \\
 \text{but cannot capture } R_1 \text{ between } F_0 \text{ and } F_1; \text{ or} \\
 \dots \\
 R_2 \text{ can capture } R_1 \text{ between } F_{i-1} \text{ and } F_i, \\
 \text{but cannot capture } R_1 \text{ between } F_0 \text{ and } F_{i-1}; \text{ or} \\
 \dots \\
 R_2 \text{ can capture } R_1 \text{ between } F_{n-1} \text{ and } F_n, \\
 \text{but cannot capture } R_1 \text{ between } F_0 \text{ and } F_{n-1}.
 \end{array} \right. \quad (1)
 \end{array}$$

Namely, R_2 cannot capture R_1 if and only if R_2 cannot capture R_1 between F_0 and F_n .

We can say that

$$\begin{array}{l}
 R_2 \text{ cannot capture } R_1 \\
 \Downarrow \\
 \left\{ \begin{array}{l}
 \bar{T}_1 = \sum_{j=0}^1 T_j > t_1 = \frac{D_1}{V_1} \\
 \bar{T}_2 = \sum_{j=0}^2 T_j > t_2 = \frac{1}{V_1} \sum_{j=1}^2 D_j \\
 \dots \\
 \bar{T}_{N-1} = \sum_{j=0}^{N-1} T_j > t_{N-1} = \frac{1}{V_1} \sum_{j=1}^{N-1} D_j.
 \end{array} \right.
 \end{array}$$

We now bring in a variable U_i , where $i = 1, 2, \dots, N-1$, to indicate whether there is any turn between F_{i-1} and F_i . U_i has two values: 0 and 1. If there is any turn between F_{i-1} and F_i , $U_i = 1$; otherwise, $U_i = 0$. Thus, there are permutations of the values of U_i . The i th element of a permutation is the value of U_i . We define a variable M , the values of which are the possible permutations. There are 2^{N-1} different permutations, so M has 2^{N-1} different values. Since U_i affects the expression of T_i , if we want to calculate $1 - P$, we should divide the capture into 2^{N-1} different subcases. As an example of calculating the subcases, we now calculate the probability of the case that R_2 cannot capture R_1 and the permutation is $(0, 0, \dots, 0)$, the probability of which we denote as p_0 . Also, we denote the probabilities that R_2 cannot capture R_1 in the other subcases as $p_1, \dots, p_{2^{N-1}-1}$. We then have

$$\begin{aligned}
 p_0 &= \text{Pr ob}\{R_2 \text{ cannot capture } R_1, M = (0, 0, \dots, 0)\} \\
 &= \text{Pr ob}\{R_2 \text{ cannot capture } R_1 | M = (0, 0, \dots, 0)\} \cdot \text{Pr ob}\{M = (0, 0, \dots, 0)\} \quad (2)
 \end{aligned}$$

$$\begin{aligned}
 &\text{Pr ob}\{M = (0, 0, \dots, 0)\} \\
 &= (P_0)^{N-1} \quad (3)
 \end{aligned}$$

$$\begin{aligned}
& \Pr \text{ob}\{R_2 \text{ cannot capture } R_1 | M = (0, 0, \dots, 0)\} \\
&= \Pr \text{ob}\{\overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots, \overline{T}_{N-1} > t_{N-1} | M = (0, 0, \dots, 0)\} \\
&= \Pr \text{ob}\left\{\sum_{j=0}^1 G(V_2, D_j) > t_1, \sum_{j=0}^2 G(V_2, D_j) > \frac{1}{V_1} \sum_{j=1}^2 D_j, \dots, \sum_{j=0}^{N-1} G(V_2, D_j) > \frac{1}{V_1} \sum_{j=1}^{N-1} D_j\right\} \quad (4) \\
&= \iint_{\substack{\overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots \\ \overline{T}_{N-1} > t_{N-1}}} \lambda^{N-1} \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j\right)} dD_1 \dots dD_{N-1}.
\end{aligned}$$

Here $\overline{T}_i = \sum_{j=0}^i G(V_2, D_j)$, $t_i = \frac{1}{V_1} \sum_{j=1}^i D_j$, $i = 1, \dots, N-1$.

Then from (3) and (4) we have

$$\begin{aligned}
p_0 &= \Pr \text{ob}\{R_2 \text{ cannot capture } R_1, M = (0, 0, \dots, 0)\} \\
&= (P_0)^{N-1} \cdot \iint_{\substack{\overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots \\ \overline{T}_{N-1} > t_{N-1}}} \lambda^{N-1} \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j\right)} dD_1 \dots dD_{N-1}. \quad (5)
\end{aligned}$$

Here $\overline{T}_i = \sum_{j=0}^i G(V_2, D_j)$, $t_i = \frac{1}{V_1} \sum_{j=1}^i D_j$, $i = 1, \dots, N-1$.

Similar to (5), we can calculate $p_1, \dots, p_{2^{N-1}-1}$. Then $1 - P = p_0 + p_1 + \dots + p_{2^{N-1}-1}$.

We continue our derivation in Problem 2. Note that if we only calculate the expected time for R_2 to capture R_1 by tracking the RFID tags, we cannot obtain an accurate expected time, but only its upper and lower bounds.

From the preceding discussion, we can rephrase (1) as

$$\begin{aligned}
& R_2 \text{ can capture } R_1 \\
& \quad \Updownarrow \\
& \left\{ \begin{array}{l} \overline{T}_1 \leq t_1; \text{ or} \\ \overline{T}_1 > t_1, \overline{T}_2 \leq t_2; \text{ or} \\ \dots \\ \overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots, \overline{T}_{i-1} > t_{i-1}, \overline{T}_i \leq t_i; \text{ or} \\ \dots \\ \overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots, \overline{T}_{N-2} > t_{N-2}, \overline{T}_{N-1} \leq t_{N-1}. \end{array} \right. \quad (6)
\end{aligned}$$

Here, we know that the $N-1$ events in (6) are mutually exclusive. Also from the discussion of U_i , $i = 1, 2, \dots, N-1$. Similarly, we should consider 2^{N-1} subcases in order to calculate the upper (denoted as T_u) and lower (denoted as T_l) bounds of the conditional expected time that R_2 takes to capture R_1 .

As an example, in order to calculate T_u and T_l , we now consider the subcase in which R_2 can capture R_1 and the permutation is $(0, 0, \dots, 0)$. We denote the upper and lower bounds of the conditional capture time as T_u^0 and T_l^0 , respectively. We also denote

the upper and lower bounds of the conditional capture time in the other subcases as $T_u^1, \dots, T_u^{2^{N-1}-1}$, and $T_l^1, \dots, T_l^{2^{N-1}-1}$, respectively. We then have

$$\begin{aligned}
T_u^0 &= (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\overline{T}_1 \leq t_1} \lambda^{N-1} \left(\sum_{j=0}^1 T_j \right) \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1} \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\overline{T}_1 > t_1, \overline{T}_2 \leq t_2} \lambda^{N-1} \left(\sum_{j=0}^2 T_j \right) \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1} \\
&+ \dots \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\substack{\overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots \\ \overline{T}_{i-1} > t_{i-1}, \overline{T}_i \leq t_i}} \lambda^{N-1} \left(\sum_{j=0}^i T_j \right) \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1} \\
&+ \dots \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\substack{\overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots \\ \overline{T}_{N-1} \leq t_{N-1}}} \lambda^{N-1} \left(\sum_{j=0}^{N-1} T_j \right) \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1}
\end{aligned} \tag{7}$$

$$\begin{aligned}
T_l^0 &= (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\overline{T}_1 \leq t_1} \lambda^{N-1} T_0 \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1} \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\overline{T}_1 > t_1, \overline{T}_2 \leq t_2} \lambda^{N-1} \left(\sum_{j=0}^1 T_j \right) \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1} \\
&+ \dots \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\substack{\overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots \\ \overline{T}_{i-1} > t_{i-1}, \overline{T}_i \leq t_i}} \lambda^{N-1} \left(\sum_{j=0}^{i-1} T_j \right) \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1} \\
&+ \dots \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\substack{\overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots \\ \overline{T}_{N-1} \leq t_{N-1}}} \lambda^{N-1} \left(\sum_{j=0}^{N-2} T_j \right) \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1}.
\end{aligned} \tag{8}$$

Here $\overline{T}_i = \sum_{j=0}^i G(V_2, D_j)$, $t_i = \frac{1}{V_1} \sum_{j=1}^i D_j$, $i = 1, \dots, N-1$, and $\sum_{j=0}^i T_j = \sum_{j=0}^i G(V_2, D_j)$, $i = 1, \dots, N-1$.

As with the T_u^0 and T_l^0 , we can obtain $T_u^1, \dots, T_u^{2^{N-1}-1}$, and $T_l^1, \dots, T_l^{2^{N-1}-1}$. Then we have $T_u = \sum_{j=0}^{2^{N-1}-1} T_u^j$ and $T_l = \sum_{j=0}^{2^{N-1}-1} T_l^j$.

Now we solve Problem 3. As in the discussion of the conditional expected time for R_2 to capture R_1 , assuming that it can capture R_1 , we cannot obtain the conditional accurate length of the path that R_2 takes to capture R_1 but only the upper and lower bounds.

From the discussion of U_i , in which $i = 1, 2, \dots, N-1$, in order to calculate the upper (denoted as D_u) and lower (denoted as D_l) bounds of the conditional expected length

of the path that R_2 follows to capture R_1 , we should consider 2^{N-1} different subcases. As an example, to calculate the upper and lower bounds, we now consider the upper and lower bounds of the conditional moving lengths of the path of R_2 in the subcase in which R_2 can capture R_1 and the permutation is $(0, 0, \dots, 0)$. We denote the upper and lower bounds of the conditional capture moving length of the path in that case as D_u^0 and D_l^0 , respectively. Also, we denote the upper and lower bounds of the conditional capture moving lengths of the path in the other subcases as $D_u^1, \dots, D_u^{2^{N-1}-1}$ and $D_l^1, \dots, D_l^{2^{N-1}-1}$, respectively. Then we have

$$\begin{aligned}
D_u^0 &= (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\overline{T}_1 \leq t_1} \lambda^{N-1} \left(\sum_{j=0}^1 D_j \right) \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1} \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\overline{T}_1 > t_1, \overline{T}_2 \leq t_2} \lambda^{N-1} \left(\sum_{j=0}^2 D_j \right) \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1} \\
&+ \dots \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\substack{\overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots \\ \overline{T}_{i-1} > t_{i-1}, \overline{T}_i \leq t_i}} \lambda^{N-1} \left(\sum_{j=0}^i D_j \right) \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1} \quad (9) \\
&+ \dots
\end{aligned}$$

$$\begin{aligned}
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\substack{\overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots \\ \overline{T}_{N-2} \leq t_{N-2} \\ \overline{T}_{N-1} \leq t_{N-1}}} \lambda^{N-1} \left(\sum_{j=0}^{N-1} D_j \right) \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1} \\
D_l^0 &= (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\overline{T}_1 \leq t_1} \lambda^{N-1} (D_0) \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1} \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\overline{T}_1 > t_1, \overline{T}_2 \leq t_2} \lambda^{N-1} \left(\sum_{j=0}^1 D_j \right) \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1} \\
&+ \dots \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\substack{\overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots \\ \overline{T}_{i-1} > t_{i-1}, \overline{T}_i \leq t_i}} \lambda^{N-1} \left(\sum_{j=0}^{i-1} D_j \right) \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1} \quad (10) \\
&+ \dots
\end{aligned}$$

$$\begin{aligned}
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\substack{\overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots \\ \overline{T}_{N-2} \leq t_{N-2} \\ \overline{T}_{N-1} \leq t_{N-1}}} \lambda^{N-1} \left(\sum_{j=0}^{N-2} D_j \right) \cdot e^{-\lambda \left(\sum_{j=1}^{N-1} D_j \right)} dD_1 \dots dD_{N-1}.
\end{aligned}$$

Here, $\overline{T}_i = \sum_{j=0}^i G(V_2, D_j)$, $t_i = \frac{1}{V_1} \sum_{j=1}^i D_j$, $i = 1, \dots, N-1$.

Then, similar to (9) and (10), we can calculate $D_u^1, \dots, D_u^{2^{N-1}-1}$ and $D_l^1, \dots, D_l^{2^{N-1}-1}$. Then we have $D_u = \sum_{j=0}^{2^{N-1}-1} D_u^j$ and $D_l = \sum_{j=0}^{2^{N-1}-1} D_l^j$. We have now solved Problem 3.

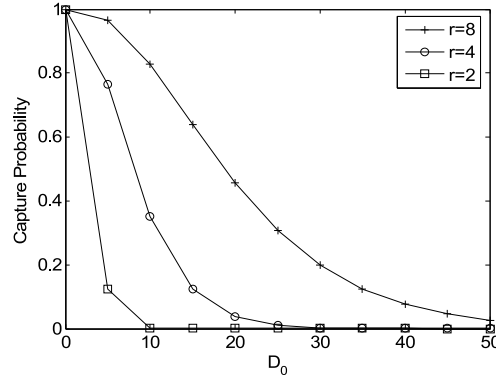
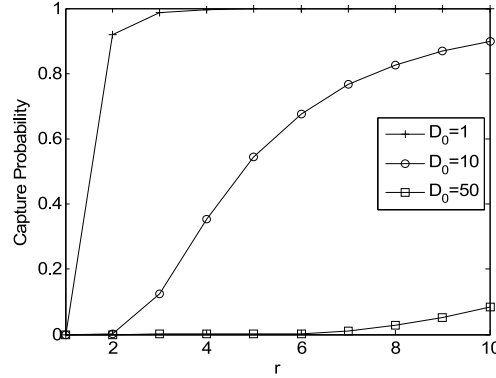
We now solve Problem 4. By focusing on only the RFID tags, we cannot obtain an accurate number of RFID tags that R_1 drops in the case that R_2 can capture R_1 . But we can obtain the upper (denoted as N_u) and lower (denoted as N_l) bounds of the conditional number.

Based on U_i , where $i = 1, 2, \dots, N - 1$, we can obtain N_u and N_l by considering 2^N subcases. For example, we consider the subcase in which R_2 can capture R_1 and the permutation is $(0, 0, \dots, 0)$. We denote the upper and lower bounds of the conditional number of RFID tags that R_1 drops in this case as N_u^0 and N_l^0 , respectively. We also denote the upper and lower bounds of the conditional capture time in the other subcases as $N_u^1 \dots N_u^{2^{N-1}-1}$ and $N_l^1 \dots N_l^{2^{N-1}-1}$, respectively. Then, we have

$$\begin{aligned}
N_u^0 &= (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\overline{T}_1 \leq t_1} \lambda^{N-1} \cdot (2) \cdot e^{-\lambda(\sum_{j=1}^{N-1} D_j)} dD_1 \dots dD_{N-1} \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\overline{T}_1 > t_1, \overline{T}_2 \leq t_2} \lambda^{N-1} \cdot (3) \cdot e^{-\lambda(\sum_{j=1}^{N-1} D_j)} dD_1 \dots dD_{N-1} \\
&+ \dots \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\substack{\overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots \\ \overline{T}_{i-1} > t_{i-1}, \overline{T}_i \leq t_i}} \lambda^{N-1} \cdot (i+1) \cdot e^{-\lambda(\sum_{j=1}^{N-1} D_j)} dD_1 \dots dD_{N-1} \quad (11) \\
&+ \dots \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\substack{\overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots \\ \overline{T}_{N-2} \leq t_{N-2} \\ \overline{T}_{N-1} \leq t_{N-1}}} \lambda^{N-1} \cdot (N) \cdot e^{-\lambda(\sum_{j=1}^{N-1} D_j)} dD_1 \dots dD_{N-1}
\end{aligned}$$

$$\begin{aligned}
N_l^0 &= (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\overline{T}_1 \leq t_1} \lambda^{N-1} \cdot (1) \cdot e^{-\lambda(\sum_{j=1}^{N-1} D_j)} dD_1 \dots dD_{N-1} \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\overline{T}_1 > t_1, \overline{T}_2 \leq t_2} \lambda^{N-1} \cdot (2) \cdot e^{-\lambda(\sum_{j=1}^{N-1} D_j)} dD_1 \dots dD_{N-1} \\
&+ \dots \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\substack{\overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots \\ \overline{T}_{i-1} > t_{i-1}, \overline{T}_i \leq t_i}} \lambda^{N-1} \cdot (i) \cdot e^{-\lambda(\sum_{j=1}^{N-1} D_j)} dD_1 \dots dD_{N-1} \quad (12) \\
&+ \dots \\
&+ (P_0)^{N-1} \cdot \frac{1}{P} \cdot \iint_{\substack{\overline{T}_1 > t_1, \overline{T}_2 > t_2, \dots \\ \overline{T}_{N-2} \leq t_{N-2} \\ \overline{T}_{N-1} \leq t_{N-1}}} \lambda^{N-1} \cdot (N-1) \cdot e^{-\lambda(\sum_{j=1}^{N-1} D_j)} dD_1 \dots dD_{N-1}.
\end{aligned}$$

Here, $\overline{T}_i = \sum_{j=0}^i G(V_2, D_j)$, $t_i = \frac{1}{V_1} \sum_{j=1}^i D_j$, $i = 1, \dots, N - 1$.

Fig. 2. Capture probability versus D_0 .Fig. 3. Capture probability versus r .

Then, similar to (11) and (12), we can calculate $N_u^1, \dots, N_u^{2^{N-1}-1}$ and $N_l^1, \dots, N_l^{2^{N-1}-1}$.

Then we have $N_u = \sum_{j=0}^{2^{N-1}-1} N_u^j$ and $N_l = \sum_{j=0}^{2^{N-1}-1} N_l^j$. We have now solved Problem 4.

5. ANALYTICAL RESULTS OF PRIMATE-INSPIRED SCENT-TRAIL COMMUNICATION

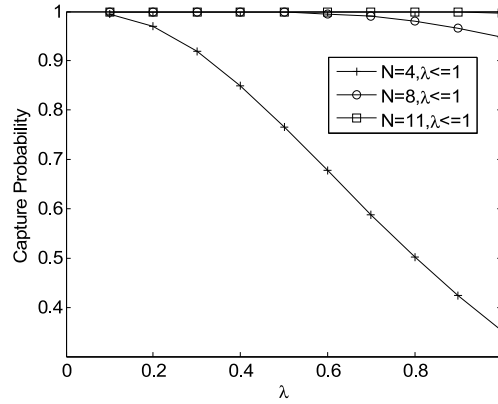
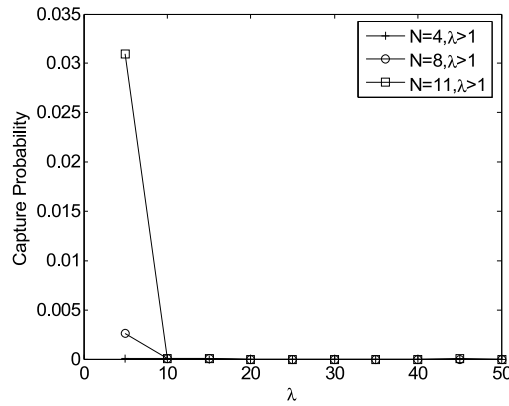
In this section, we obtain some numerical results for the performance of the scent-trail mechanism under different parameters.

In this section, we assume that (1) there is no turn in the road, and that (2) $G(V_2, D_i) = D_i/V_2$ and $H(V_2, D_i) = D_i/aV_2$, $aV_2 \geq V_1$, where $i = 1, 2, \dots, n$. We now denote the ratio of V_2 to V_1 as r . Then, from the first assumption, we know that $1 - P = p_0$, $T_u = T_u^0$, $T_l = T_l^0$, $D_u = D_u^0$, $D_l = D_l^0$, $N_u = N_u^0$, and $N_l = N_l^0$.

5.1. The Probability that R_2 Can Capture R_1

In Figure 2, we fix $\lambda = 1$, $N = 4$, $r = 2, 4$, and 8 , and let D_0 increase from 0 to 50 with an increment of 5 . Figure 2 shows the capture probability P with three different r values over the distance D_0 between R_1 and R_2 . As illustrated in the figure, given a fixed r , P decreases as D_0 increases. With each fixed D_0 , P increases as r increases.

In Figure 3, we fix $\lambda = 1$, $N = 4$, $D_0 = 1, 10$, and 50 , and let r increase from 1 to 10 in increments of 1 . Figure 3 shows the capture probability P with three different D_0

Fig. 4. Capture probability versus λ .Fig. 5. Capture probability versus λ .

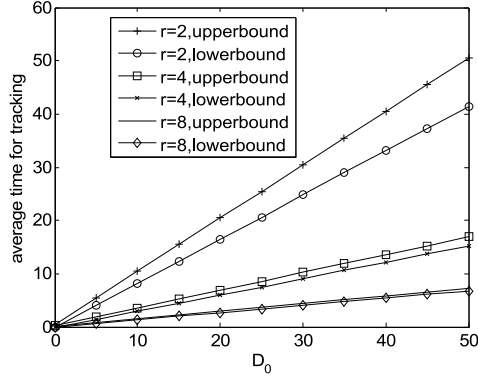
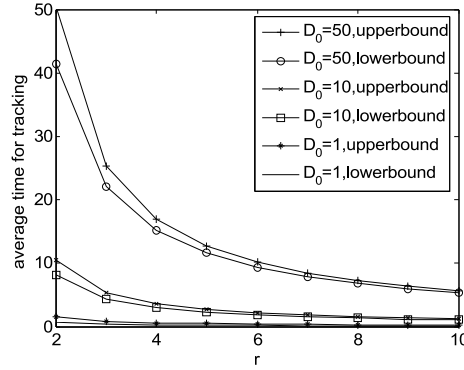
values over the ratio of V_2 to $V_1 - r$. As illustrated in the figure, given a fixed D_0 , P increases as r increases. With each fixed r , P decreases as D_0 increases.

In Figure 4, we fix $D_0 = 10$, $N = 4, 8$, and 11 , and let λ increase from 0.1 to 1 in increments of 0.1 . Figure 4 shows the capture probability P with three different N values over λ . As illustrated in the figure, given a fixed N , P decreases as λ increases. With each fixed λ , P increases as N increases.

In Figure 5, we fix $D_0 = 10$, $N = 4, 8$, and 11 , and let λ increase from 5 to 50 in an increments of 5 . Figure 5 shows the capture probability P with three different N values over λ . As illustrated in the figure, given a fixed N , P decreases as λ increases. With each fixed λ , P increases as N increases.

5.2. The Conditional Average Time that R_2 Takes to Capture R_1 with the Condition that R_2 Can Capture R_1

In Figure 6, we fix $\lambda = 1$, $N = 4, r = 2, 4$, and 8 , $V_1 = 1$, and let D_0 increase from 0 to 50 in increments of 5 . Figure 6 shows the upper and lower bounds of the conditional average time that R_2 takes to capture R_1 in the case that it can capture R_1 . It shows the upper and lower bounds of the conditional average time over D_0 with three different values of r . As illustrated in the figure, with each fixed r , the difference between the upper and

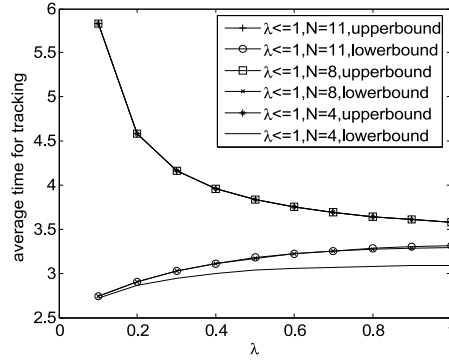
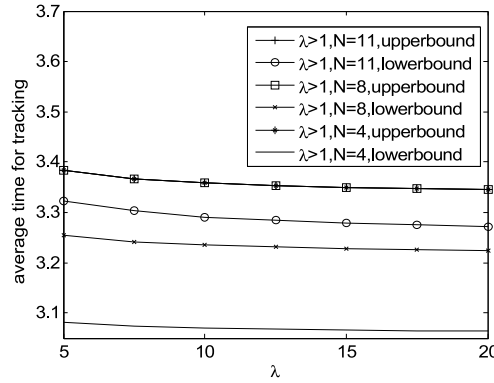
Fig. 6. Average time for tracking when capturing versus D_0 .Fig. 7. Average time for tracking when capturing versus r .

lower bounds increases as D_0 increases. With each fixed D_0 , the difference between the upper and lower bounds decreases as r increases.

In Figure 7, we fix $\lambda = 1$, $N = 4$, $D_0 = 1, 10$, and 50 , and $V_1 = 1$, and let r increase from 2 to 10 in increments of 1. Figure 7 shows the upper and lower bounds of the conditional average time that R_2 takes to capture R_1 in the case that it can capture R_1 . It shows the upper and lower bounds of the conditional average time over r with three different values of D_0 . As illustrated in the figure, with each fixed r , the difference between the upper and lower bounds increases as D_0 increases. With each fixed D_0 , the difference between the upper and lower bounds decreases as r increases.

In Figure 8, we fix $D_0 = 10$, $N = 4, 8$, and 11 , $V_1 = 1$, and $r = 4$, and let λ increase from 0.1 to 1 in increments of 0.1. Figure 8 shows the upper and lower bounds of the conditional average time that R_2 takes to capture R_1 in the case that it can capture R_1 . It shows the upper and lower bounds of the conditional average time over λ with three different values of N . As illustrated in the figure, with each fixed N , the difference between the upper and lower bounds decreases as λ increases.

In Figure 9, we fix $D_0 = 10$, $N = 4, 8$, and 11 , $V_1 = 1$, and $r = 4$, and we let λ increase from 5 to 20 in increments of 2.5. Figure 9 shows the upper and lower bounds of the conditional average time that R_2 takes to capture R_1 in the case that it can capture R_1 . It shows the upper and lower bounds of the conditional average time over λ with three different values of N . As illustrated in the figure, with each fixed N , the difference between the upper and lower bounds decreases as λ increases.

Fig. 8. Average time for tracking when capturing versus λ .Fig. 9. Average time for tracking when capturing versus λ .

5.3. The Conditional Average Length of the Path that R_2 Takes to Capture R_1 With the Condition that R_2 Can Capture R_1

In Figure 10, we fix $r = 2, 4, \text{ and } 8$, and let D_0 increase from 0 to 50 in increments of 5. Figure 10 shows the upper and lower bounds of the conditional average length of the path that R_2 takes to capture R_1 in the case that it can capture R_1 . It shows the upper and lower bounds of the conditional average length of the path that R_2 takes over D_0 with three different values of r . As illustrated in the figure, with each fixed r , the difference between the upper and lower bounds increases as D_0 increases. With each fixed D_0 , the difference between the upper and lower bounds increases as r increases.

In Figure 11, we fix $D_0 = 1, 10, \text{ and } 50$, and let r increase from 2 to 10 in increments of 1. Figure 11 shows the upper and lower bounds of the conditional average length of the path that R_2 takes to capture R_1 in the case that it can capture R_1 . It shows the upper and lower bounds of the conditional average length of the path that R_2 takes over r with three different values of D_0 . As illustrated in the figure, with each fixed D_0 , the difference between the upper and lower bounds decreases as r increases. With each fixed r , the difference between the upper and lower bounds increases as D_0 increases.

In Figure 12, we fix $D_0 = 10, N = 4, 8, \text{ and } 11, V_1 = 1$, and $r = 4$, and let λ increase from 0.1 to 1 in increments of 0.1. Figure 12 shows the upper and lower bounds of the conditional average length of the path that R_2 takes to capture R_1 in the case that it can capture R_1 . It shows the upper and lower bounds of the conditional average length of the path that R_2 takes over λ with three different values of N . As illustrated

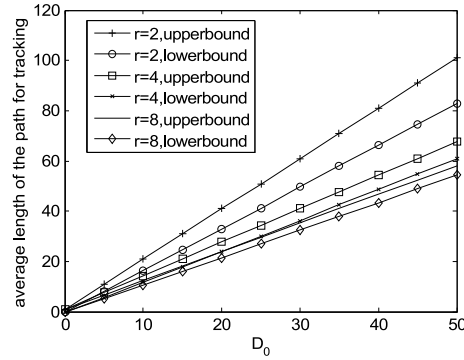


Fig. 10. Average length of path for tracking when capturing versus D_0 .

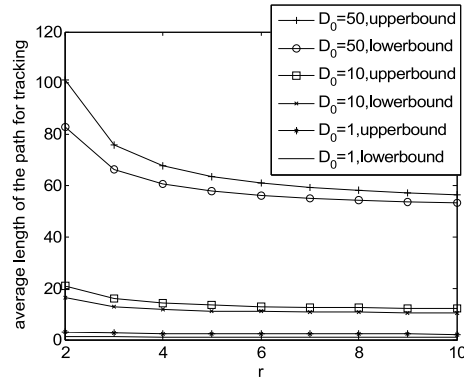


Fig. 11. Average length of path for tracking when capturing versus r .

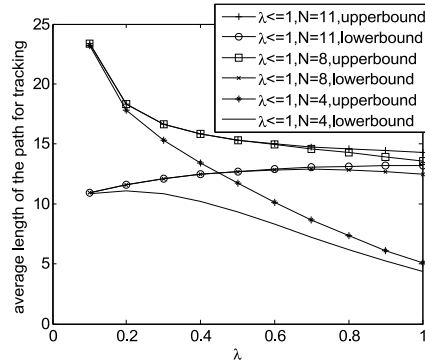


Fig. 12. Average length of path for tracking when capturing versus λ .

in the figure, with each fixed N , the difference between the upper and lower bounds decreases as λ increases.

In Figure 13, we fix $D_0 = 10$, $N = 4, 8$, and 11 , $V_1 = 1$, and $r = 4$, and let λ increase from 5 to 20 in increments of 2.5. Figure 13 shows the upper and lower bounds of the conditional average time that R_2 takes to capture R_1 in the case that it can capture R_1 . It shows the upper and lower bounds of the conditional average length of the path

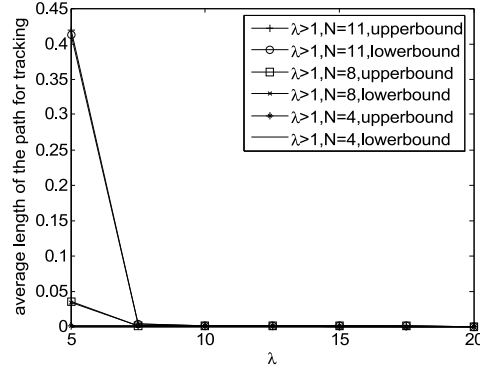


Fig. 13. Average length of path for tracking when capturing versus λ .

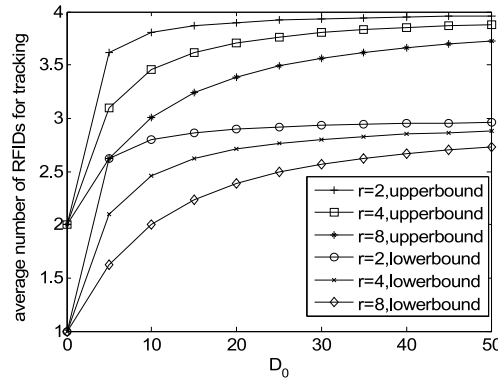


Fig. 14. Average number of RFIDs for tracking when capturing versus D_0 .

that R_2 takes over λ with three different values of N . As illustrated in the figure, with each fixed N , the difference between the upper and lower bounds is so slight that it is approximated to zero.

5.4. The Conditional Average Number of RFID Tags that R_1 Drops With the Condition that R_2 Can Capture R_1

In Figure 14, we fix $r = 2, 4, \text{ and } 8$, and let D_0 increase from 0 to 50 in increments of 5. Figure 14 shows the upper and lower bounds of the conditional average number of RFID tags that R_1 drops in the case that R_2 can capture R_1 . It shows the upper and lower bounds of the conditional average number of RFID tags that R_1 drops over D_0 with three different values of r . As illustrated in the figure, with each fixed r , both the upper and lower bounds increase when D_0 increases. With a fixed D_0 , both the upper and lower bounds decrease as r increases.

In Figure 15, we fix $D_0 = 1, 10, \text{ and } 50$, and let r increase from 2 to 10 in increments of 1. Figure 15 shows the upper and lower bounds of the conditional average number of RFID tags that R_1 drops in the case that R_2 can capture R_1 . It shows the upper and lower bounds of the conditional average number of RFID tags that R_1 drops over r with three different values of D_0 . As illustrated in the figure, with each fixed D_0 , both the upper and lower bounds decrease as r increases. With a fixed r , both the upper and lower bounds decrease as D_0 increases.

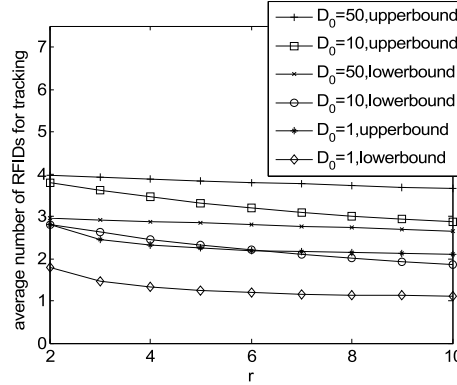


Fig. 15. Average number of RFID tags for tracking when capturing versus r .

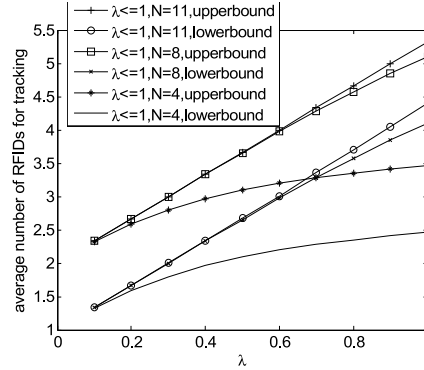


Fig. 16. Average number of RFID tags for tracking when capturing versus λ .

In Figure 16, we fix $D_0 = 10$, $N = 4, 8$, and 11 , $V_1 = 1$, and $r = 4$ and let λ increase from 0.1 to 1 in increments of 0.1 . Figure 16 shows the upper and lower bounds of the conditional average number of RFID tags that R_1 drops in the case that R_2 can capture R_1 . It shows the upper and lower bounds of the conditional average number of RFID tags that R_1 drops over λ with three different values of N . As illustrated in the figure, with each fixed N , the difference between the upper and lower bounds is a constant 1 and both the upper and lower bounds increase as λ increases.

In Figure 17, we fix $D_0 = 10$, $N = 4, 8$, and 11 , $V_1 = 1$, and $r = 4$, and let λ increase from 5 to 20 in increments of 2.5 . Figure 17 shows the upper and lower bounds of the conditional average number of RFID tags that R_1 drops in the case that R_2 can capture R_1 . It shows the upper and lower bounds of the conditional average number of RFID tags that R_1 drops over λ with three different values of N . As illustrated in the figure, with each fixed N , the difference between the upper and lower bounds is a constant 1 and both the upper and lower bounds increase as λ increases.

6. FURTHER STUDIES ON THE PRIMATE-INSPIRED SCENT-MARKING STRATEGY

In this section, we first provide further studies on our primate-inspired strategy for robot collaboration in mobile and static sensors. Similar to primates, whose glands or urine can be utilized for scent-marking, robots are equipped with RFID tags or sensors that can be deployed with information for others to pick up. We then provide some analysis of this strategy.

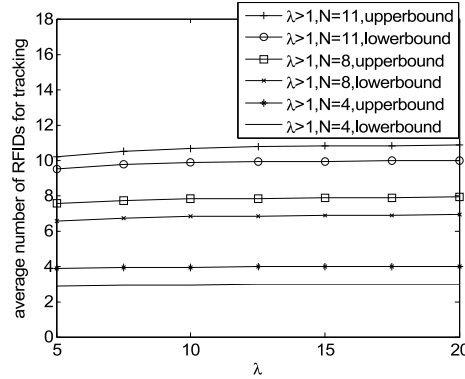


Fig. 17. Average number of RFID tags for tracking when capturing versus λ .

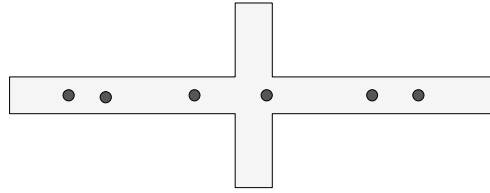


Fig. 18. RFID tags or sensors.

6.1. The Strategy

In our strategy, robots are not only responsible for collecting information and saving energy for the whole network but are also critical for detecting and handling events. In the network, several robots are normally deployed to patrol their areas of monitoring responsibility. Once they detect an event, they try to address it. Sometimes, robots need help. Upon detecting such an event, they deploy RFID tags or sensors that carry information necessary to attract other robots passing by to collaborate to handle the problem. For example, when detecting pollution or radiation, robots remain at the polluted area waiting for other robots. For some other events, such as detecting enemy traffic, robots need to track the traffic and deploy RFID tags and sensors on their way to provide directions for other robots. They need to leave enough information to guide other robots to catch up and collaborate.

Patrolling robots are responsible for collecting data from sensors, detecting events in time, and deploying an adequate number of RFID tags or sensors to inform other robots when collaboration is necessary. When a robot discovers an event, it will go to the site and deploy RFID tags or sensors on its way. If the robot is near the site, it will simply deploy RFID tags or sensors in order to attract nearby robots. Directed by the information conveyed by those deployed RFID tags or sensors, the followers are thus informed about events and try to catch up to the former robot for cooperation.

Figure 18 shows an example of the deployment of RFID tags. However, only a limited number of RFID tags or sensors can be carried by each robot. Carrying too many RFID tags or sensors is not only a waste of resources, but it is also unrealistic. RFID tags and sensors have their own lifetimes and communication ranges. An RFID tag will run out of power when its energy lifetime expires.

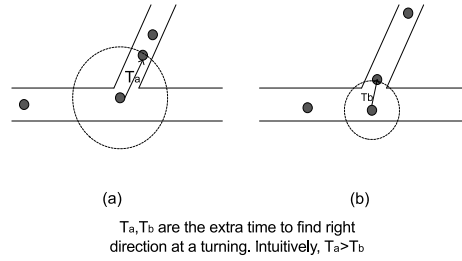


Fig. 19. Extra time for finding the right direction at a turn.

6.2. Analysis of Collaborations Between Robots

We assume that robots move with a fixed speed along a line/curve unless they make turns. Therefore, between each pair of RFID tags, the trajectory of the robot is the segment between the two tags. When robots make turns, they drop a tag at the corner to clearly indicate their routes. It is quite possible that several robots may detect tags from the same robot. As robots are all independent from each other, we only need to study the performance of a single follower. In the following discussions, we use Robot1 to denote the lead robot and Robot2 to denote the follower.

6.2.1. Searching Function. When Robot2 discovers an RFID tag, it will try to find other RFID tags one by one until it arrives at the site of the event. As RFID tags are randomly distributed, the distance between each pair of tags is a random variable. Correspondingly, the time for Robot2 to find each RFID tag is also a random variable. We use $s(d, v)$ to describe the searching function for Robot2 to find the next RFID, where d is the distance between two tags and v is the speed of the follower. In our study, we will employ several models of the searching process.

6.2.2. Turning Function. From a realistic point of view, robots do not always travel in an unvarying direction. They make turns whenever roads turn. In our model, we assume that Robot1 drops at least one RFID tag at the corner of each turn so that the route of Robot1 is clearly indicated.

Robot2's searching work will not always follow its current path. It may also take some extra time to determine the direction whenever there is a turn on Robot1's route. We use another function $f(d, v)$ to calculate the extra time taken by Robot2 to find the correct direction when Robot1 has made a turn, where d is the distance between two tags and v is the speed of Robot2. Figure 19 illustrates two turning examples in which the short distance costs less time to find the right direction. In the following study, we also employ a function to model the time cost of turning.

6.2.3. Tradeoff. Whether RFID tags or sensors are deployed sparsely or densely can critically impact whether and when, Robot2 can catch up to Robot1. On the one hand, if RFID tags are densely distributed, Robot2 can find the next RFID tag easily, which saves searching time. The total distance covered by the tags is short, and Robot2 may not catch up to Robot1 within the distance since the number of RFID tags is limited and the tags can be used up within a time period. On the other hand, having the RFID tags sparsely distributed causes Robot2 to take longer to find the next RFID tag but also saves tags. The total distance covered by RFID tags is extended so that Robot2 has a greater probability of catching up to Robot1. Therefore, there is a tradeoff between the density of deployed tags and the delay of catching up to Robot2.

Meanwhile, RFID tags have a set lifetime. To save RFID tags, robots may deploy them sparsely. Consider the mean value of the distance between two RFID tags. With

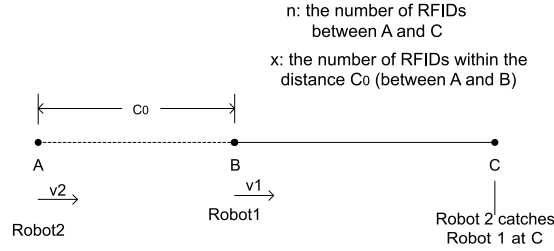


Fig. 20. Robot 2 catches up to Robot1 at C before RFID tags are consumed.

a larger mean value, it will take longer for followers to find the next tag. If these RFID tags are not found during their lifetime, they will die. These RFID tags are then wasted. The lifetime consideration complicates the pursuing problem.

As RFID tags are randomly deployed, we evaluate the RFID tags' density by the mean value of their distribution. A large mean value corresponds to a sparse deployment, and vice versa.

6.2.4. Pursuing Problem. We use V_1 and V_2 to describe the speeds of Robot1 and Robot2, respectively. In order to catch up to Robot1, Robot2 must have a high enough V_2 .

Figure 20 shows Robot2 catching up to Robot1 at location C, where Robot2 detected the first RFID tag at location A while Robot1 was already at location B.

Robot1 has a limited number of RFID tags, denoted as N . The number of RFID tags between B and C is x . We use n to denote the number of RFID tags between locations A and B. Then we have $x + n \leq N$.

We use "delay" to describe the time required for Robot2 to catch up to Robot1 between A and C. If Robot2 cannot catch up to Robot1, we consider the delay to be infinite.

$$\text{delay} = \begin{cases} c_0/v_2 + \sum_{i=1}^{x-1} s(d_i, v_2) + \sum_{i=1}^{x-1} k_i \cdot f(d_i, v_2), & (\text{catches up}) \\ \infty, & (\text{cannot catch up}) \end{cases};$$

$$k_i = \begin{cases} 0, & \text{if Robot1 turns at tag } i \\ 1, & \text{if Robot1 doesn't turn at tag } i \end{cases}$$

As in our assumption, Robot1 makes a turn with a probability of p when it drops an RFID tag. d_i is the distance between tag i and tag $(i + 1)$. The condition that Robot2 can catch up to Robot1 is

$$\text{delay} \leq \sum_{i=1}^{N-n-1} d_i/v_1.$$

6.3. Metrics

In the collaboration scheme, whether Robot2 can catch up to Robot1 is the main focus. We study the catch probability and delay between Robot1 and Robot2, which are impacted by many factors, such as the speeds of the two robots, the initial distance, and so on.

For the pursuing problem, the speeds of the two robots are the most critical factors. We also study the speeds required for Robot1 and Robot2 to meet each other. It is easy to understand that the faster the following robot and the slower the former robot, the

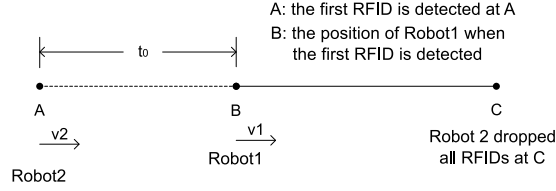
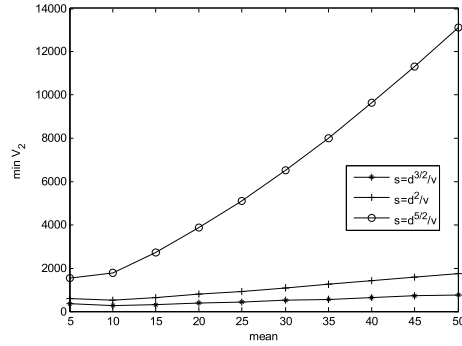


Fig. 21. The pursuing problem of two Robots.

Fig. 22. Minimum V_2 vs. mean (with different searching functions).

easier it becomes for the two robots to meet each other. Therefore, we study the ranges of the two robots' speeds (the maximum and minimum values of Robot1).

Meanwhile, we also study the impact of other factors, such as the mean value of the deployment of RFID tags, the searching function, the turning function, and so on. In order to show their roles in the problem, we assume some models of searching and turning functions for our purposes; however, the models may not exactly correspond with real situations.

7. SIMULATION WITHOUT CONSIDERING LIFETIME OF RFID TAGS/ SENSORS

In this section, we study the pursuing problem without considering the lifetimes of RFID tags/sensors. The simulations are conducted using a discrete event simulation that uses C++ language.

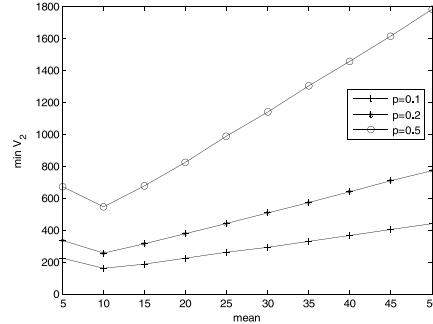
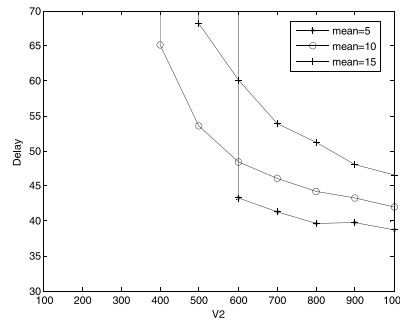
We assume that RFID tags are deployed in an exponential distribution with a mean value. We also assume that, when Robot1 drops an RFID tag, it makes a turn with probability p . We assume that Robot1 can deploy no more than $x = 100$ RFID tags. As shown in Figure 21, Robot2 detects the first RFID tag at location A while Robot1 is already at location B. After that, Robot1 can drop at most 100 RFID tags. Robot2 must catch up to Robot1 before location C, where Robot1 uses up 100 RFID tags. We use t_0 to describe the time required for Robot2 to reach location B.

In the following simulations, we study the performance of the minimum requirement of V_2 , the maximum requirement of V_1 , the delay time for Robot2 to catch up to Robot1, and the catch probability.

7.1. Min (V_2) vs. Mean:

Figures 22 and 23 show the performance of the minimum requirement of V_2 .

Figure 22 shows the performance of the minimum V_2 with different searching functions and 20% turning probability. The searching functions are $s(d, v) = d^{5/2}/v$, $s(d, v) = d^2/v$, and $s(d, v) = d^{3/2}/v$. The turning function is $f(d, v) = \pi d^2/v$. Figure 23

Fig. 23. Minimum V_2 vs. mean (with different turning probabilities).Fig. 24. Delay vs. V_2 .

shows the performance of the minimum V_2 with turning probabilities of 10%, 20%, and 50%. The searching function is the same: $s(d, v) = d^{3/2}/v$.

From these two figures, we can learn that the minimum requirement of V_2 is not linear with the mean. In Figure 22, from the upper line or with the searching function $s(d, v) = d^{5/2}/v$, the minimum requirement of V_2 increases with the mean value. More detailed information about the lower lines is shown in Figure 23.

In Figure 23, when the mean value is very small, Robot1 will drop all its RFID tags in a short time. Then Robot2 needs to move at a high speed (V_2) in order to catch up to Robot1 before all its RFID tags are dropped. When the mean value is very large, Robot2 needs to search for the next RFID tag over a much larger distance. Therefore, Robot2 should also move at a high speed in order to catch up.

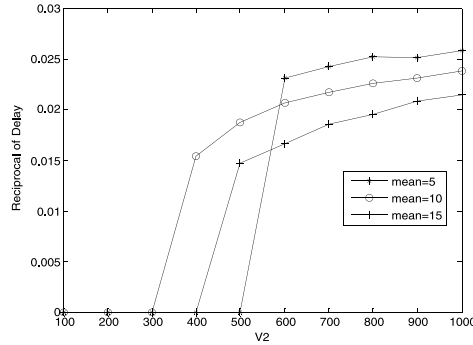
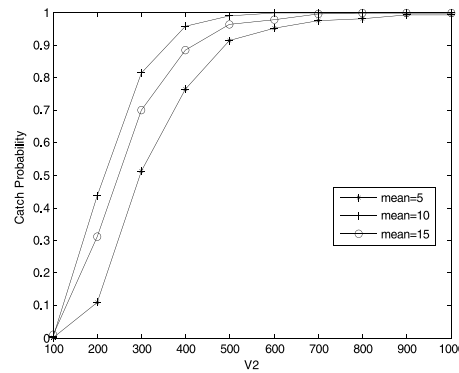
The minimum requirement of V_2 is related not only to the mean value and the probability of a turn, but also to the choice of searching function and t_0 .

7.2. Delay vs. V_2

In this subsection, we study the delay for different speeds of Robot2 (V_2). Figure 24 shows the delay when $V_1 = 10$ and $t_0 = 35$; we tried the simulation thousands of times to achieve more general results.

From the figure, we can learn that the delay decreases with the speed of Robot2. However, for lower speeds, the figure shows some unusual results. The reason is that, with a lower speed, Robot2 is less likely to catch up to Robot1.

During the following simulations, we provide a threshold to determine whether Robot2 can catch up to Robot1. We simulate each speed 10000 times. If Robot2 does not catch up to Robot1 more than 100 times, we say that Robot2 cannot catch up to Robot1.

Fig. 25. Delay reciprocal vs. v_2 .Fig. 26. Catch probability vs. V_2 .

The threshold denotes the number of times Robot2 fails to catch up to Robot1. It can be any reasonable data in the simulation. Figure 24 shows the result.

From Figure 24, we can learn that the delay starts from an infinite value at the beginning of the V_2 axis, which means that Robot2 cannot catch up to Robot1 at this speed. Furthermore, the delay decreases with V_2 . With a larger mean value, the delay is also larger.

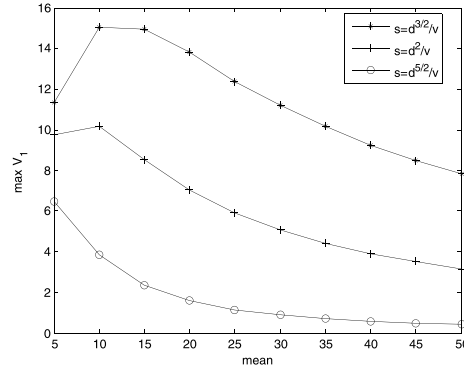
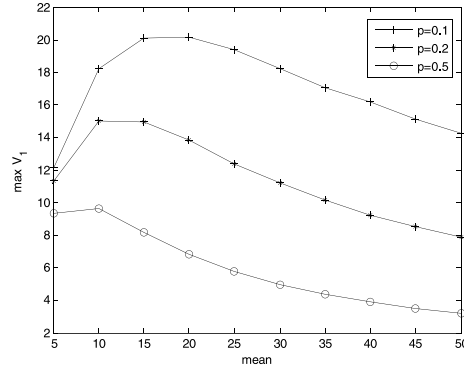
In order to express the simulation results more clearly, especially for the infinite value, we draw another figure, which shows the reciprocal value of the delays. The reciprocal value of the infinite result is zero. The result is shown in Figure 25.

From Figure 25, we can learn that the reciprocal value of the delay increases with V_2 . In other words, the delay decreases with the speed of Robot2. The beginning points are on the V_2 axis, which means the delay is infinite and Robot2 cannot catch up to Robot1 at this speed. For a different mean, the reciprocal value of the delays starts from a different speed. This is because we cannot do an infinite number of simulations. For another simulation, the value may start at another speed. However, the figure still shows the performance of the delay, which decreases with the speed of Robot2.

7.3. Catch Probability vs. V_2

In this subsection, we study the probability that Robot2 catches up to Robot1. The simulation is conducted with $V_1 = 10$ and $t_0 = 35$. The result are shown in Figure 26.

From the figure, we can learn that the probability increases with the speed of Robot2. When the speed is very low, the catch probability is nearly zero. That is, at such

Fig. 27. Maximum V_1 vs. mean.Fig. 28. Maximum V_1 . vs. mean.

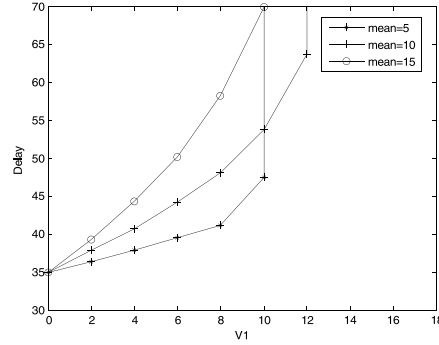
a low speed, Robot2 cannot catch up to Robot1. When Robot2 moves at a higher speed, the probability of catching up to Robot1 increases. However, even when the speed is very high, such as 700 or 800, it is still possible that Robot2 cannot catch up to Robot1. This is due to the deployment of RFID tags, which follows an exponential distribution. The distance between each pair of RFID tags is random. Therefore, the time Robot2 takes for each distance is also random. Then, within the 10000 simulations, it is possible that the distance between these RFID tags takes Robot2 a great deal of time to cover and that it cannot even catch up to Robot1.

However, from the figure, we cannot learn the exact performance of catch probability for different mean values. When mean = 10, the catch probability is greater than when mean = 5 or 15. Such a performance is consistent with Figure 23, in which we analyzed the impact of the mean.

7.4. Max V_1 vs. Mean

In this subsection, we study the performance of the maximum value of V_1 . Intuitively, the higher the speed of Robot1 (V_1), the more difficult it is for Robot2 to catch up to Robot1. Therefore, it is of great importance to study the maximum requirement of V_1 . In this subsection, we assume that $V_2 = 500$ and $t_0 = 35$.

Figures 27 and 28 show the performance of the maximum requirement of V_1 . Figure 27 shows the performance of the maximum of V_1 with different searching functions and 20% turning probability. The searching functions are $s(d, v) = d^{5/2}/v$,

Fig. 29. Delay vs. V_1 .

$s(d, v) = d^2/v$, and $s(d, v) = d^{3/2}/v$. The turning function is $f(d, v) = \pi d^2/v$. Figure 28 shows the performance of the maximum V_1 with turning probabilities of 10%, 20%, and 50%. The searching function is the same, $s(d, v) = d^{3/2}/v$, and the turning function is still $f(d, v) = \pi d^2/v$.

From Figure 27, we can learn that larger power searching functions correspond to a smaller V_1 . This is because, the more complex the searching function, the more difficult it is for Robot2 to find the next RFID tag and therefore to catch up to Robot1. The lowest line with the searching function $s(d, v) = d^{5/2}/v$ shows that the maximum requirement of V_1 decreases with the mean value. But, from the upper two lines, we can learn that the maximum requirement of V_1 is not linear with the mean value. More detailed information on upper lines is shown in Figure 28.

Figure 28 shows the detailed performance of the maximum V_1 with searching function $s(d, v) = d^{3/2}/v$. Three different turning probabilities are considered. When the mean is very small, RFID tags are more easily consumed. Thus the speed of Robot1 should not be so high that Robot2 is more likely to catch up to Robot1 before all RFID tags are dropped. When the mean value is very large, Robot2 needs more time to search for the next RFID tag. Robot1 should therefore move slowly to offer Robot2 more time to catch up.

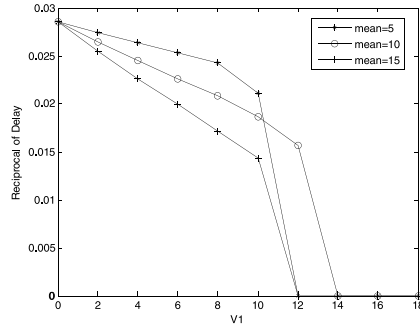
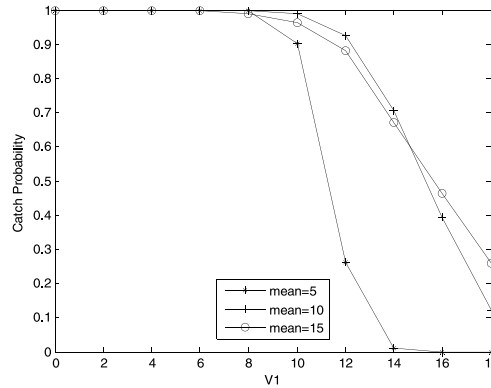
The maximum requirement of V_1 is not only related to the mean value and the probability of turning, but also to the choice of searching function and t_0 . In this work, we choose some power functions of the distance between each pair of RFID tags to simulate the searching process of Robot2.

7.5. Delay vs. V_1

We now study the performance of the delay vs. the speed of Robot1 (V_1). We still have $V_2 = 500$ and $t_0 = 35$.

In our simulation, we simulate the deployment thousands of times. We still use the same threshold as in Subsection B. Figure 29 shows the results of the delay. From the figure, we can learn that delay increases with the speed of Robot1. The results are consistent with our intuition. The faster Robot1 moves, the more difficult it is for Robot2 to catch up.

In Figure 29, V_1 starts at 0 and the delay starts at t_0 . This means that Robot1 does not move, but stays there and waits for Robot2. Then the delay increases with V_2 . Larger mean values, correspond to larger delays. When V_1 is as large as 12 or 14, all delays become infinite (Robot2 cannot catch up to Robot1 before all RFID tags are dropped).

Fig. 30. Reciprocal of delay vs. V_1 .Fig. 31. Catch probability vs. V_1 .

In order to express the simulation results more clearly, we draw another figure representing the reciprocal value of delays in Figure 30. The reciprocal value of the infinite result then is zero.

From Figure 30, we can learn that the reciprocal value of the delay decreases with V_1 . That is, the delay increases with the speed of Robot1. With a larger V_1 , the result is zero. This means that the delay is infinite and that Robot2 cannot catch up to Robot1 at such a speed. For different means, the reciprocal values of delays end at different speeds. This is because we cannot do an infinite number of simulations. For another simulation, the value may end at another speed. However, the figure still shows the performance of the delay, which increases with the speed of Robot1.

7.6. Catch Probability vs. V_1

In this subsection, we study the probability that Robot2 catches up to Robot1. The simulation is conducted with $V_2 = 500$ and $t_0 = 35$. The results are shown in Figure 31.

From the figure, we can learn that the probability decreases with the speed of Robot1. When the speed is very low, the catch probability is nearly 1. In other words, with a low speed, Robot2 can more easily catch up to Robot1. When Robot1 moves at a higher speed, the probability of being caught decreases. In fact, even when the speed of Robot1 is very low, it is still possible that Robot2 cannot catch up to Robot1. This is due to the deployment of RFID tags in an exponential distribution. The distance between each pair of RFID tags is random. Therefore, the time Robot2 takes to cover each distance is also random. Then, within the 10000 simulations, it is possible that

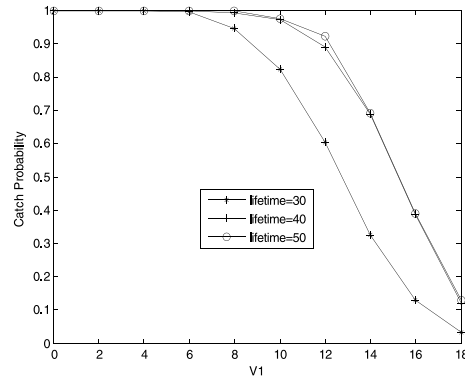


Fig. 32. Catch probability vs. V_1 with considering the lifetime of RFID.

the distance between these RFID tags costs Robot2 a great deal of time to cover and that it cannot even catch up to Robot1.

However, we cannot learn the exact performance of catch probabilities with different mean values from the figure. At some speeds, such as from 10 to 14, with $mean = 10$, the catch probability is higher than the performance with $mean = 5$ or 15 . Such a performance is consistent with Figure 23, in which we analyzed the impact of the mean.

8. SIMULATION CONSIDERING THE LIFETIMES OF RFID TAGS/ SENSORS

When we studied the deployment of the RFID tags of robots, we learned that intensity greatly impacts performance. On one hand, if RFID tags are densely deployed, the former robot will soon use up its RFID tags. The follower then has to catch up to the former over a short distance, which means that the required speed is high. On the other hand, if RFID tags are sparsely deployed, the follower will spend more time searching for the next tag. In order to catch up to the former robot, the follower then needs to move at a high speed.

However, when we analyzed this problem in the previous section, we assumed that RFID tags/sensors were on at all times. We know that RFID tags/sensors run out of power at the end of their lifetime. In this section, we study the impact of the lifetime of RFID tags/sensors.

8.1. Catch Probability vs. V_1

In this subsection, we study the probability of Robot2 catching up to Robot1 while considering the lifetime of RFID tags /sensors. The simulation is conducted with $V_2 = 500$, $t_0 = 35$, and $mean = 10$. Three lifetime values are chosen for study 30, 40, and 50. The results are shown in Figure 32.

From the figure, we can learn that the probability decreases with the speed of Robot1, which is consistent with the results of our simulations in the previous section. With a longer lifetime, the catch probability is greater than it is with a shorter lifetime. This is also consistent with our intuition. With a longer lifetime, RFID tags/ sensors would remain for a longer time and therefore would help Robot2 catch up to Robot1 more easily and quickly.

8.2. Catch Probability vs. V_2

In this subsection, we continue to study the probability that Robot2 catches up to Robot1, while considering the lifetime of RFID tags/sensors. The simulation is conducted

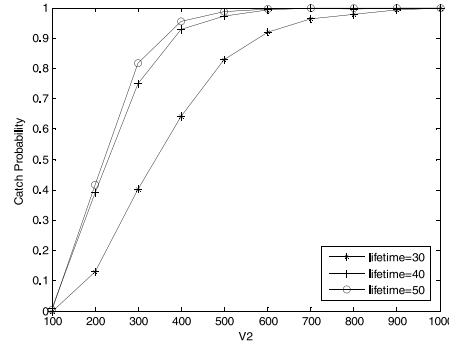


Fig. 33. Catch probability vs. V_2 with considering the lifetime of RFID.

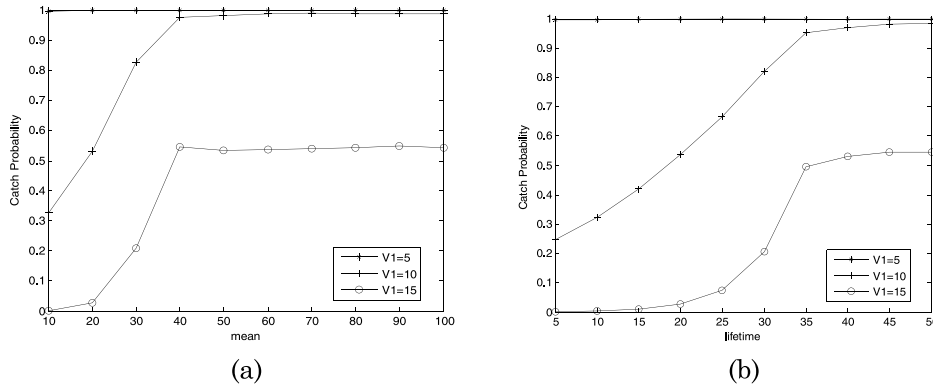


Fig. 34. Catch probability vs. the lifetime of RFID.

with $V_1 = 10$, $t_0 = 35$, and $mean=10$. Three lifetime values are chosen for study: 30, 40, and 50. The results are shown in Figure 33.

From the figure, we can learn that the probability increases with the speed of Robot2, which also achieves similar results as our simulations in the last section. With a longer lifetime of RFID tags/sensors, the catch probability is greater than it is with a shorter lifetime. This is consistent with our intuition. With a longer lifetime, RFID tags/sensors would remain for a longer time and therefore would help Robot2 catch up to Robot1 more easily and quickly.

8.3. Catch Probability vs. Lifetime

In this subsection, we continue to study the probability that Robot2 catches up to Robot1, while considering the lifetime of RFID tags/sensors. The simulation is conducted with $V_2 = 500$, $t_0 = 35$, and $mean = 10$. Three values of V_1 are chosen for study: 5, 10, and 15. The results are shown in Figure 34.

From the figure, we can learn that the probability increases with the lifetime of RFID tags/sensors. With a longer lifetime of RFID tags/sensors, the whole searching progress of Robot2 will receive more guidance. Then the catch probability is higher than it is with a shorter lifetime. This is also consistent with our intuition. With a longer lifetime, RFID tags/sensors would be able to remain for a longer time, which would help Robot2 catch up to Robot1 more easily and quickly.

Meanwhile, for a larger value of V_1 , the corresponding value of the catch probability is lower. The faster Robot1 moves, the more difficult it is for Robot2 to catch up.

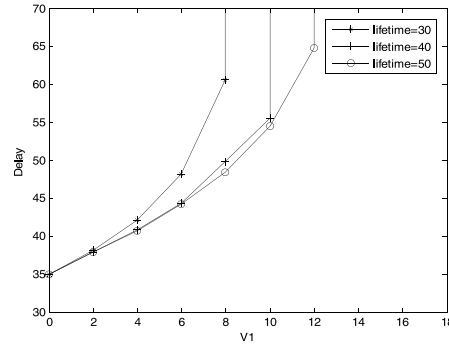


Fig. 35. Delay vs. V_1 with considering the lifetime of RFID.

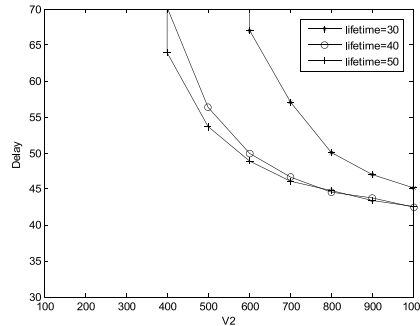


Fig. 36. Delay vs. V_2 with considering the lifetime of RFID.

8.4. Delay vs. V_1

Now we study the performance of delay vs. the speed of Robot1 (V_1), while considering the lifetime of RFID tags/sensors. We still have $V_2 = 500$, $t_0 = 35$, and $lifetime = 30, 40$, and 50.

We employ a similar method in Subsection 7.5 as in the last section. The results of the simulation are shown in Figure 35, and they are similar to our previous simulations. The results are consistent with our intuition. The faster Robot1 moves, the more difficult it is for Robot2 to catch up. Therefore, the delay is longer.

For RFID tags with a longer lifetime, the delay is shorter. This is because, the longer the RFID tag is active, the more guidance Robot2 will receive.

8.5. Delay vs. V_2

In this subsection, we continue to study the performance of delay vs. the speed of Robot2 (V_2), while considering the lifetime of RFID tags/sensors. We still have $V_1 = 10$, $t_0 = 35$, and $lifetime = 30, 40, 50$.

We employ a similar method as in the simulation of Subsection 7.2. The results of the simulation are shown in Figure 36, and they are similar to our former simulations. The results are consistent with our intuition. The faster Robot2 moves, the more difficult it is for Robot2 to catch up.

For RFID tags/sensors with a longer lifetime, the delay is shorter. This is because the longer an RFID tag/sensor is active, the more guidance Robot2 will receive. It will therefore be easier for Robot2 to catch up to Robot1.

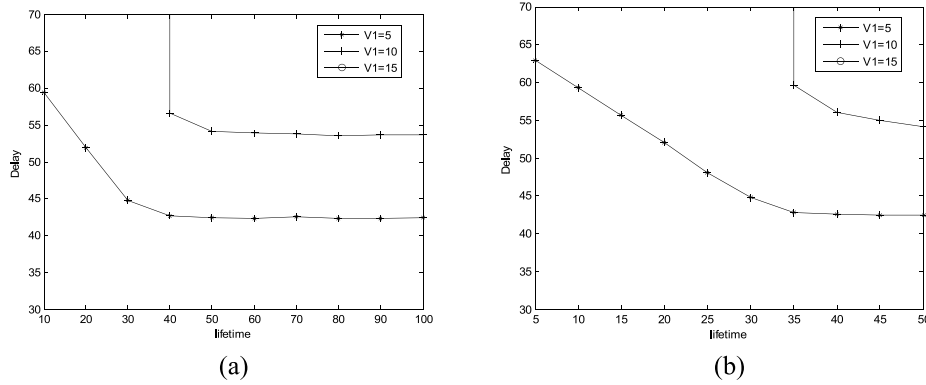


Fig. 37. Delay vs. the lifetime of RFID.

8.6. Delay vs. Lifetime

In this subsection, we continue to study the delay time for Robot2 to catch up to Robot1 while considering the lifetime of RFID tags/sensors. The simulation is conducted with $V_2 = 500$, $t_0 = 35$, and $mean = 10$. Three values of V_1 are chosen for study: 5, 10, and 15. The results are shown in Figure 37.

From the figure, we can learn that the delay decreases with the lifetime of RFID tags. For RFID tags with a longer lifetime, the whole searching progress of Robot2 will receive more guidance. Then the delay of Robot2 will be shorter than it is with a shorter lifetime. This is also consistent with our intuition. With a longer lifetime, RFID tags/sensors would be able to remain for a longer time, which would help Robot2 catch up to Robot1 more easily and quickly.

Meanwhile, for a larger value of V_1 , the corresponding value of the catch probability is lower. The faster Robot1 moves, the more difficult it is for Robot2 to catch up.

From Figure 37, we cannot see the results of the delay with $V_1 = 15$. This is because, with such a large value of V_1 , no matter what lifetime value is chosen, Robot2 cannot catch up to Robot1 and the delay is always infinite.

9. FUTURE RESEARCH DIRECTIONS

9.1. Multiple Robot Communications

Until now, we have mainly focused on the pursuit between two robots. However, the pursuing problem should involve several or even many robots, especially for some difficult tasks, such as firefighting, tracking an enemy tank, and so on, which can only be accomplished through the collaboration of more robots. From a realistic point of view, once RFID tags/sensors are deployed along the track of robots, all other robots may discover them and then pursue the former robot through these tags or sensors.

We will then not only care about whether they can catch up to the former robot, but also about the speed of the robots pursuing the former robot and the number of robots gathering.

First, we provide some description and assumptions of the problem. The former robots, which detect some events, will deploy RFID tags/sensors around the event. Then they will wait at the position of the event for the arrival of other robots. In such a situation, with $V_1 = 0$, all other robots that detect these tags/sensors can catch up to Robot1 with a probability of 1, which will be demonstrated in the following simulations. Robots that detect these RFID tags/sensors will gather at the event's position or the former robot's position, following the instructions of these RFID tags/sensors. The

former robot may also broadcast a message through the wireless network in order to notify more robots within the whole network.

Generally, for each robot, we assume that they are patrolling within the wireless network and monitoring the environment mainly in their areas of responsibility. Once they receive an announcement from the wireless network about some urgent events, they patrol for the destinations following the instructions of the wireless network and RFID tags/sensors deployed by the former robot. We also design a patrolling strategy for robots, which will be introduced in the following section.

For this problem, the most important metric for evaluation is the gathering speed of following robots, which is how many robots gather together within one unit of time.

9.2. Robots May Leave RFID Tags to Override Existing RFID Tags

Robots deploy RFID tags or sensors carrying event-related information to inform their partners. After being deployed, RFID tags /sensors will stay where they were deployed and keep the implanted information until they die. Therefore, the corresponding weakening of information becomes an obstacle for following robots to catch up. In order to enhance the instruction and direction, followers may deploy their own RFID tags or sensors to cover for the old tags if they are dying.

Meanwhile, the network and task are not invariable and the schedule of events also changes with real situations. Therefore, the information carried by an RFID tag may become out of date. We assume that information carried by each RFID tag includes a time stamp telling the age of the event. When followers pass by with newer information, they can deploy their own RFID tags to override the formerly deployed RFID tags. For the sensor case, the information of sensors can be updated by the follower after authentication.

9.3. Hierarchy of Robots

For the real-time problem, different events have different priorities because some events may be more urgent or important. Generally, tougher or more important tasks are always assigned to more reliable robots. Robots are also different in their functions. Thus, some robots may be faster or have more functions and be stronger, while others are limited in their abilities. We assume that robots employed in the network know their hierarchies according to their capabilities. Therefore, when there are some tougher or more urgent tasks, higher hierarchy robots are preferred.

We define the priority of events according to the hierarchy of the robots. When a robot with a higher hierarchy, carrying its own event, passes by an RFID tag, the robot can override the former information by deploying its own RFID tags with implanted information regardless of the former event. Therefore, a robot with a dominant status can dismiss the information deployed by subordinate robots.

9.4. Territoriality Defense

Sensors and RFID tags carrying the distributors' information can not only direct partners but also inform intruders. Commonly, sensor networks are deployed to monitor an area by detecting information from intruders. Such a process is very similar to territoriality defense.

Traditionally, sensors or RFID tags play the part of collectors by gathering information from the environment and reporting events to the network. With a territoriality defense, sensors/RFID tags should be able to actively communicate with the intruder about the territoriality ownership and warn the intruder away from the area. Sensors/RFID tags can send some electronic digital signals, much like the welcome sign of a

city in the real world, to warn intruders, to advise them to avoid further conflicts, and to alert the network or the personnel of the possible intrusions.

Based on different signal characters of sensors and RFID tags, we propose these two kinds of strategies in detail as follows.

9.4.1. RFID Tags. RFID tags are devices with simple and limited capabilities. After deployment they cannot be modified. We assume that RFID tags periodically send out electronic messages and that those who are within communication range receive the messages. There are two performance metrics to consider. The first one is the energy efficiency of the RFID tags. In order to inform intruders, RFID tags need to send messages frequently, which will cost a great deal of energy. Meanwhile, if the messages are sent out over a large period, there might be no messages sent out when an intruder passes through the boundary. Then, the intruder is not rewording intrusion activity. Here is a tradeoff between energy efficiency and intruders being informed.

There are also some other related topics. First, how far to deploy an RFID tag around the boundary is the deployment problem directly impacting the signal coverage. We assume that all intruders passing through the boundary can recognize the signal. Second, in order to save energy we can arrange for the neighboring RFID tags to send signals intermittently. We can divide RFID tags into different groups that work alternately but with the same period. Neighboring RFID tags work alternately.

9.4.2. Sensors. Compared to RFID tags, sensors work in a more flexible way and are more powerful. Sensors can send warning messages whenever an intruder is detected. In such a way, energy consumption is more efficient. We only need to consider how to ensure a high probability that the intruders are noticed, which is greatly related to the sensing coverage of the sensor network. Then the problem is similar to the boundary coverage problem, which we have studied in other papers such as Xiao and Zhang [2011a].

9.5. Sensors for Directing the Followers

When robots deploy sensors instead of RFID tags, the chain of sensors will form a routing path that directs following robots and extends the communication among sensors, especially those sensors on the chain. We assume that sensors deployed by robots are all within the communication range of their neighbors. The former robot's location is known by all sensors on the chain. Then, once a following robot encounters a sensor on the chain, it will get the latest information about the location of the former robot.

This will benefit the pursuing problem in two ways. First, it saves time spent on searching. Robot2 will know the latest location of Robot1 when it detects a sensor on the chain. Therefore, Robot2 can directly go to the location without searching for the sensors on the chain one by one. Second, with information on the location of Robot1, Robot2 can seek a shorter way to catch up to Robot1.

9.6. Prediction Algorithms

Regardless of whether sensors or RFID tags are deployed by robots, the chasing robot can employ an appropriate prediction algorithm to forecast the moving pattern of Robot1. With the help of the prediction algorithm, the pursuing problem of Robot2 will be easier, due to less searching time and higher chasing efficiency.

9.7. Robot Patrol Algorithm

Robots are introduced in our system to handle events or incidents. Here an event may be the detection of an intrusion tank or warning information indicated in a data packet that is related to some abnormal situation. When a robot (Robot1) detects an event,

it will go back to its way coming here. Then it will choose an appropriate distance and turn back to the location of the event. After turning back to the event location, it will deploy RFID tags/sensors on its way. When other robots detect one of these RFID tags/sensors, they will begin to pursue the first robot in order to gather together to handle the events.

When there is no event in the network, robots patrol following a random patrol scheme such as in Xiao and Zhang [2011a] and Zhang and Xiao [2011]. Initially, they are given the topology information from sinks and multihop sensors. They randomly choose a route as their patrol path. When they arrive at a critical point, such as an intersection, they randomly choose a route different than the one they came from Xiao and Zhang [2011a]. The situation where two robots meet together should also be considered. They may meet at a critical point or on a road. They first exchange their information and then decide the path to patrol. They also share tasks if one of them has tasks.

We don't care how robots handle the events (being beyond our), but focus on their patrolling strategy.

9.8. Related to Realistic Problems

Until now we have mainly focused on robots, and the research sounds greatly academic. In fact, this research is very similar to practical problems. A typical example is a wild adventure. People need to leave some mark to record their route, which can also help followers catch up. Meanwhile, animals often do similar things. Ants leave pheromones that can only be recognized by them. Therefore, when they find some food, many partners will gather to carry the food. Meanwhile, some zoologists also track animals according to their excreta to study their living habits.

9.9. Transmission Range

In the previous subsections, we did not consider the ranges of RFID tags/sensors. In fact, robots do not need to arrive at the exact position of an RFID tag/sensor but within a nearby range. Each RFID tag has a signal range, and once robots are in the detection range they can achieve their positions in the formation of the RFID tags/sensors. In the future we will study the impact of this signal transmission range in the pursuing schemes.

10. CONCLUSIONS

Scent-marking is a practical method for applications of mobile and static sensors with RFID tags. Robots are resource-rich mobile devices with more energy, higher power, and better processing capabilities and can make decisions and execute appropriate actions according to the information gathered from sensors. For some tasks, robots need to collaborate. With equipped RFID tags, robots can also leave information for other robots. In this article, we proposed several primate-inspired communication mechanisms including delayed-and-relayed and scent-trail communications among robots. We analytically modeled and simulated scent-trail communication. We proposed and studied the primate-inspired scent-marking method of robots in pursuing and tracking problems. We studied related metrics, such as speed requirements, the related delays, and the catch probabilities.

Future work will include studying the effectiveness of bio-inspired communications in robot collaboration, the optimum number of messages to be relayed as in scent trails, and changing deployment strategies or speeds according to the current situation.

REFERENCES

- ABRAMS, Z., GOEL, A., AND PLOTKIN, S. 2004. Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In *Proceedings of the International Symposium on Information Processing in Sensor Networks (IPSN)*.
- AMUNDSON, I., SALLAI, J., KOUTSOUKOS, X., LEDECZI, A., AND MAROTI, M. 2011. RF angle of arrival-based node localization. *Int. J. Sens. Netw.* 9, 3/4, 209–224.
- ARIENZO, L. AND LONGO, M. 2011. Energy-efficient collaborative tracking in wireless sensor networks. *Int. J. Sens. Netw.* 9, 3/4, 124–38.
- AZAD, A. P. AND CHOCKALINGAM, A. 2011. Enhancing lifetime of wireless sensor networks using multiple data sinks. *Int. J. Sens. Netw.* 9, 3/4, 139–157.
- BOUBICHE, D. AND BILAMI, A. 2011. HEEP (Hybrid Energy Efficiency Protocol) based on chain clustering. *Int. J. Sens. Netw.* 10, 1/2, 2011, 25–35.
- BRASS, P. 2007. Bounds on coverage and target detection capabilities for models of networks of mobile sensors. *ACM Trans. Sens. Netw.* 3, 2.
- DIAS, M. B. AND STENTZ, A. 2000. A free market architecture for distributed control of a multirobot system. In *Proceedings of the 6th International Conference on Intelligent Autonomous Systems (IAS)*. 115–122.
- DIXSON, A. F. AND BRANCOFT, J. 1998. *Primate Sexuality*. Oxford University Press, 14–50.
- FIORIE, A. D., LINK, A., AND STEVENSON, P. R. 2006. Scent marking in two western Amazonian populations of Woolly monkeys (*Lagothrix Lagotricha*). *Amer. J. Primatology* 68, 637–649.
- GUPTA, H., DAS, S., AND GU, Q. 2003. Connected sensor cover: Self organization of sensor networks for efficient query execution. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*.
- JAGGI, N. AND KAR, K. 2011. Multi-sensor activation for temporally correlated event monitoring with renewable energy sources. *Int. J. Sens. Netw.* 10, 1/2, 42–58.
- JEDERMANN, R., BECKER, M., GORG, C., AND LANG, W. 2011. Testing network protocols and signal attenuation in packed food transports. *Int. J. Sens. Netw.* 9, 3/4, 2011, 170–181.
- KAFETZOGLOU, S. AND PAPAVALASSILOU, S. 2011. Energy-efficient framework for data gathering in wireless sensor networks via the combination of sleeping MAC and data aggregation strategies. *Int. J. Sens. Netw.* 10, 1/2, 23–13.
- KRONTIRIS, I. AND DIMITRIOU, T. 2011. Scatter—Secure code authentication for efficient reprogramming in wireless sensor networks. *Int. J. Sens. Netw.* 10, 1/2, 14–24.
- LEWIS, R. J. 2006. Research article scent marking in Sifaka: No one function explains it all. *Amer. J. Primatology* 68, 622–636.
- LI, F. 2011. Adaptive resource allocation in multiuser cooperative networks with proportional rate constraints. *Int. J. Sens. Netw.* 10, 1/2, 104–110.
- LIU, C., WU, K., XIAO, Y., AND SUN, B. 2006. Random coverage with guaranteed connectivity: Joint scheduling for WSNs. *IEEE Trans. Parall. Distrib. Syst.* 17, 6, 562–575.
- LU, J. AND SUDA, T. 2007. Differentiated surveillance for static and random mobile sensor networks. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*. 1102–1107.
- LU, L., WU, J. C., AND CHEN, S. 2011. A cluster-based algorithm for redundant nodes discovery in dense sensor networks. *Int. J. Sens. Netw.* 10, 1/2, 59–72.
- MAJUMDAR, A. AND WARD, R. K. 2011. Increasing energy efficiency in sensor networks: Blue noise sampling and non-convex matrix completion. *Int. J. Sens. Netw.* 9, 3/4, 158–169.
- MIAO, L., QI, H., AND WANG, F. 2005. Self-deployable mobile sensor networks for on-demand surveillance. In *Proceedings of SPIE*, vol. 5778, 496.
- MORREALE, P., QI, F., AND CROFT, P. 2011. A green wireless sensor network for environmental monitoring and risk identification. *Int. J. Sens. Netw.* 10, 1/2, 73–82.
- OLTEANU, A., XIAO, Y., WU, K., AND DU, X. 2010. Weaving a proper net to catch large objects in wireless sensor networks. *IEEE Trans. Wirel. Comm.* 9, 4, 1360–1369.
- PENG, M., XIAO, Y., AND WANG, P. 2009. Error analysis and kernel density approach of scheduling sleeping nodes in cluster-based wireless sensor networks. *IEEE Trans. Veh. Technol.* 58, 9, 5105–5114.
- POORNIMA, A. S. AND AMBERKER, B. B. 2011. PERSEN: Power-efficient logical ring based key management for clustered sensor networks. *Int. J. Sens. Netw.* 10, 1/2, 94–103.
- RAO, R. AND KESIDIS, G. 2004. Purposeful mobility for relaying and surveillance in mobile ad hoc networks. *IEEE Trans. Mobile Comput.* 3, 3, 225–232.

- SHAKKOTTAI, S., SRIKANT, R., AND SHROFF, N. 2003. Unreliable sensor grids: Coverage, connectivity and diameter. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM)*.
- SHEN, C., HONG, Y.-W. P., CHAO, C., AND YANG, S. 2011. Editorial. *Int. J. Sens. Netw.* 9, 3/4, 121–123.
- SIMMONS, R., APFELBAUM, D., BURGARD, W., FOX, D., MOORS, M., AND THRUN, S. 2000. Coordination for multi-robot exploration and mapping. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.
- SMUTS, B. B., CHENEY, D. L., AND SEYFARTH, R. M. 1987. Primate societies. In *The National Academies*, 400–500.
- SOLIMAN, H. AND AL-OTAIBI, M. 2011. Enhancing AODV routing protocol over mobile ad hoc sensor networks. *Int. J. Sens. Netw.* 10, 1/2, 36–41.
- SNOWDON, C. T., BROWN, C. H., AND PETERSON, M. R. 1985. *Primate Communication*. Cambridge University Press, 283–293.
- TANIGUCHI, Y., KITANI, T., AND LEIBNITZ, K. 2011. A uniform airdrop deployment method for large-scale wireless sensor networks. *Int. J. Sens. Netw.* 9, 3/4, 182–191.
- TRAN, T. D., AGBINYA, J. I., AND AL-JUMAILY, A. A. 2011. Per node deployment based detection of controlled link establishment attack in distributed sensor networks. *Int. J. Sens. Netw.* 9, 3/4, 192–208.
- TSENG, C., LEE, S., PRASAD, N. R., AND WÓDCZAK, M. 2011. Editorial. *Int. J. Sens. Netw.* 10, 1/2, 1–2.
- WANG, L. AND XIAO, Y. 2006. A survey of energy-efficient scheduling mechanisms in sensor networks. *ACM/Springer Mob. Netw. Appl.* 11, 5, 723–740.
- WÓDCZAK, M. 2011. Autonomic cooperative networking for wireless green sensor systems. *Int. J. Sens. Netw.* 10, 1/2, 83–93.
- WU, K., GAO, Y., LI, F., AND XIAO, Y. 2005. Lightweight deployment-aware scheduling for wireless sensor networks. *ACM/Springer Mob. Netw. Appl.* 10, 6, 837–852.
- XIAO, Y. AND LIANG, X. 2009. Bio-inspired communications among robots, sensors, and RFID tags. In *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA)*. 3573–3578.
- XIAO, Y. AND ZHANG, Y. 2009. Surveillance and tracking system with collaboration of robots, sensor nodes, and RFID tags. In *Proceedings of the 18th International Conference on Computer Communications and Networks (ICCCN)*.
- XIAO, Y. AND ZHANG, Y. 2011a. Divide- and conquer-based surveillance framework using robots, sensor nodes, and RFID tags. *Wireless Comm. Mob. Comput.* (Special Issue Emerging Techniques for Wireless Vehicular Communications). John Wiley & Sons. To appear.
- XIAO, Y. AND ZHANG, Y. 2011b. A critical line based boundary surveillance strategy in wireless sensor networks. *Telecomm. Syst. J.* (Special Issue on Challenges in Next-Generation and Resource-Constrained Networks). To appear.
- XIAO, Y., CHEN, H., WU, K., SUN, B., ZHANG, Y., SUN, X., AND LIU, C. 2010. Coverage and detection of a randomized scheduling algorithm in wireless sensor networks. *IEEE Trans. Comput.* 59, 4, 507–521.
- YAMAUCHI, B. 1998. Frontier-based exploration using multiple robots. In *Proceedings of the 2nd International Conference on Autonomous Agents*. 47–53.
- ZHANG, Y. AND XIAO, Y. 2009a. A boundary surveillance strategy based on critical line in wireless sensor networks. In *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA)*. 459–464.
- ZHANG, Y. AND XIAO, Y. 2009b. Primate-inspired scent marking for mobile and static sensors and RFID tags. In *Proceedings of the 18th International Conference on Computer Communications and Networks (ICCCN)*.
- ZHANG, Y. AND XIAO, Y. 2011. A patrolling scheme in wireless sensor and robot networks. In *Proceedings of the 3rd International Workshop on Wireless Sensor, Actuator and Robot Networks (WiSARN) in conjunction with IEEE INFOCOM*.
- ZHANG, Y., XIAO, Y., AND BALES, K. L. 2009. Primate social systems, scent-marking, and their applications in mobile and static sensor networks. *Int. J. Sens. Netw.* 5, 4, 210–222.
- ZLOT, R. M., STENTZ, A., DIAS, M. B., AND THAYER, S. 2002. Multi-robot exploration controlled by a market economy. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Received January 2010; accepted August 2010