Analysis of Exploration in Swarm Robotic Systems

Minyoung Jeong, John Harwell, and Maria Gini

Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, USA {jeong242,harwe006,gini}@umn.edu

Abstract. The exploration time for a swarm of robots doing object gathering depends on the specific task, environment, number of objects, and number of robots. Hence, it is important to understand how different properties affect the exploration time. To address this practical challenge, we present a theoretical analysis of the expected exploration time for a swarm of robots that are searching for objects in an arena and present experimental results obtained with different swarm sizes. The experimental results are consistent with the theoretical analysis.

1 Introduction

Swarm robotics is the study of decentralized multi-robot systems consisting of robots with limited individual capabilities, such as local sensing, low processing power, and limited or no inter-agent communications. In swarm robotics, large groups of simple robots coordinate implicitly to collectively perform tasks such as pattern formation, foraging, or collective mapping [17].

In a foraging task, robots search for goal locations where target objects (e.g., food, humans, etc) reside, and bring them back to a centralized collection location called the nest. Applications of foraging include hazardous waste clean up [4], search and rescue [6, 5], demining and terrain sample collections [15].

Since object locations are initially unknown, exploring the environment is important for completing a foraging task. Due to limited individual capabilities, random walk (or variants) is frequently used for exploration, because of its limited processing requirements and scalability [20, 14, 8]. We expect random walks of a swarm to speedup the exploration process compared to a single robot.

Random walk is a discrete time stochastic process where a path is established by successive random steps. When the random steps are time-independent, the process can be described by a discrete-time Markov chain [13]. If the next step is correlated to the orientation of the previous step, the walk is categorized as correlated random walk (CRW). In contrast to uncorrelated random walk, in a CRW the probability of the next steps depends on the direction at which the robot has moved previously [10, 21, 20]. While most theoretical results in the literature are for strictly random walks, parallel CRWs in robot swarms share many of the same properties and are more feasible with real robots (e.g., most wheeled robots are non-holonomic and not able to move perpendicularly to their orientation, as is required for strictly random walks).

An important property of random walks that has been studied extensively is the *"first passage time"*, which is the first time when a random walk reaches a specific location, described formally as a random variable. Many papers have studied how to compute the mean first passage time of a more general stochastic process [18, 15]. In particular, Hunter [12] focused on numerical algorithms for computing the mean first passage time of a Markov chain. Other works focus on computing a first passage time distribution of Markov chains [1, 9].

Increasing the swarm size tends to decrease the *exploration time* $t_e(n, N)$, i.e., the time it takes for n robots to reach the goal location $\mathbf{G}$ when there are N robots, where $n \leq N$, with linear decreases proportional to increasing N. However, beyond a certain swarm size in a fixed size arena, adding more robots would result in a negligible decrease in the $t_e(n, N)$, as suggested by the experimental results from [11, 20]. Increasing N even further may increase $t_e(n, N)$ due to high levels of inter-robot interference.

For different foraging problems, the desired $t_e(n, N)$ may differ. Consider a search and rescue mission as an example, in which $t_e(n, N)$ may be determined by how long a missing person can stay alive [2]. To save as many as possible, it is important to determine the appropriate N to minimize $t_e(n, N)$ without having to run a large number of experiments. To solve such problem, we seek to answer the following research question: **For swarms of N robots doing correlated random walks (CRWs), can we compute a priori an estimate of the time by which n robots will have reached the goal locations?**

This paper is focused on swarm exploration in foraging tasks. In Section 4, we model robots that perform a CRW by Markov chains. Using the model, we compute the probability that n robots have reached $\mathbf{G}$ when there are N robots in total. To solve this problem, we first consider a simpler problem in Section 5.1, where the first passage time of a single robot is computed. In Section 5.2, under the assumption that random walks of robots are independent and identically distributed (i.i.d.), we extend our results to n robots. Finally, in Section 7, we validate the effectiveness of our model by comparing the experimental results we obtained in simulation with the theoretical predictions of the model.

2 Related Work

Pang *et al.* [19] model the random walk of a swarm of robots using models from biology research. The mean squared displacement (MSD) is proposed as a performance measure to quantify the effectiveness of random walk methods, where MSD is proportional to the performance of the random walk method. The authors derived that the MSD of an uncorrelated random walk in two dimensions is equal to $m \times D(L) + m \times (E(L))^2$, where m is the number of steps taken, L is the step length, $E(L)$ the expected value of step lengths L, and $D(L)$ the variance of step lengths.

In more general settings, Alon *et al.* [3] studied speed ups of cover times for different classes of graphs when there are k random walks. The cover time of a graph is the maximum expected time it takes to explore the whole graph

starting from an arbitrary initial node. The speed up of the cover time is simply the ratio between the cover time of a single random walk and the cover time of k random walks. The authors relate other quantities, such as the hitting time and the mixing time, with the cover time. While the work gives results with formal justifications, some results, in particular speed up of 2-dimensional grid graph, are limited to $k < O(\log^{1-\epsilon} n)$ for any $\epsilon > 0$, where n is the number of nodes in the graph. In the context of swarm exploration, it is important to consider the case when there are many robots, i.e., when $k \gg O(\log^{1-\epsilon} n)$.

Dong *et al.* [7] derived an explicit formula for the first hitting time distributions of the standard Brownian motion with piece-wise linear boundaries, which can be used to further deduce formulas for rectangular, triangle, and quadrilateral boundaries. For general nonlinear boundaries, approximate solutions were obtained.

3 Problem Statement

Suppose there are N robots, and their states are random variables, $X_t^{(1)}, \ldots, X_t^{(N)}$, which are discrete time Markov processes with the transition matrix, $P_{i,j}$, where i, j are states. The probability that the robot is in state j after t steps is denoted as $P_{i,j}(t) = (P^t)_{i,j}$.

In the context of swarm exploration, each state encodes the position of the robot in the arena, $\mathcal{A} = [0, H] \times [0, W]$, where H is the heigth and W is the width of the arena. We allow states to include other information besides position. Let $\mathbf{G}$ denote the goal location, which is a proper subset of $\mathcal{A}$. $\mathbf{G}$ contains objects that the robots need to collect. For simplicity, we assume that $\mathbf{G}$ is rectangular, i.e. $\mathcal{G} = [a', b'] \times [c', d']$ where a', b' are X-coordinates, and c', d' are Y-coordinates of the arena.

In addition, assume that the random variables $X_t^1, \ldots, X_t^N$ are independent and identically distributed (i.i.d.) for each $t \geq 0$. Their initial states are all the same, i.e. $X_0^1 = \cdots = X_0^N$.

With the given assumptions, we want to find the probability mass function, $P(n, t)$, which is the probability that n robots out of the total N robots have reached the goal $\mathbf{G}$ by time t.

4 Proposed State Representation

To represent the states, we describe first how to represent the arena and then we describe the transition probability.

Arena Representation: A naive way to model a continuous arena as a discrete time Markov chain is to place a node at each square grid element (see Fig. 1) [16]. Edges are placed between neighboring nodes, including the ones diagonal to each other. With this representation, if the side length of a square grid is $L_\square$, then there are $R_\square := \lfloor \frac{H}{L_\square} \rfloor$ rows and $C_\square := \lfloor \frac{W}{L_\square} \rfloor$ columns of $R_\square \times C_\square$ nodes. The

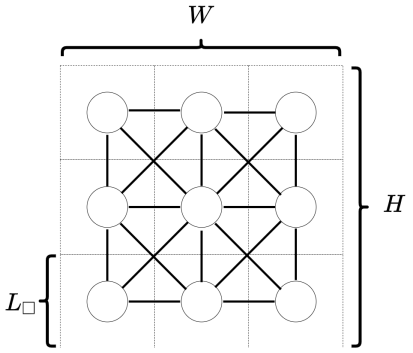


Fig. 1. $R_\square = C_\square = 3$, where $R_\square, C_\square$ are the numbers of rows and columns, respectively, for a square grid representation.

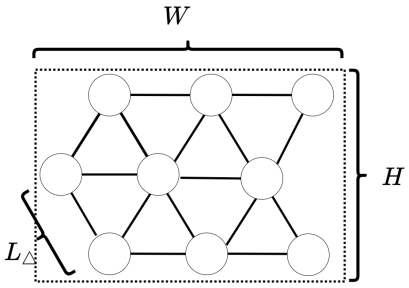


Fig. 2. $R_\triangle = C_\triangle = 3$, where $R_\triangle, C_\triangle$ are the numbers of rows and columns, respectively, for a triangular grid representation.

disadvantage of the square grid representation is that the distance between a node and the neighboring node in the diagonal direction is different from the distance between the given node and the neighboring node in the horizontal or vertical direction.

To address this issue, a triangular grid, as shown in Fig. 2, can be constructed from a square grid by moving every n-th row to the right and every m-th row to the left by some amount, where n is odd and m is even. With this model, one benefit is that the distance between each node is the same, which corresponds to a more accurate representation of a robot moving at constant velocity. Given the side length of a triangular grid, $L_\triangle$, there are $R_\triangle := \lfloor \frac{2H}{\sqrt{3}L_\triangle} \rfloor$ rows and $C_\triangle := \lfloor \frac{W}{L_\triangle} \rfloor$ columns of $R_\triangle \times C_\triangle$ nodes. Here, the number of rows is computed by dividing the height of the arena by the height of a triangle.

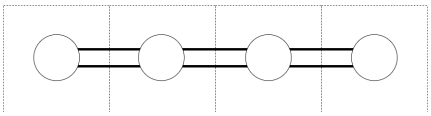


Fig. 3. States representation in a one-dimensional grid encoding only robot positions, with no direction of motion encoded.

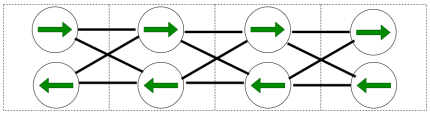


Fig. 4. States in a one-dimensional grid encoding position and direction of motion. A right arrow encodes a robot moving rightward, a left arrow encodes a robot moving leftward.

In addition to modeling robot position, we also need to incorporate the direction of robot motion for CRW exploration strategies in order to build our hitting

time model (this is not needed for strictly random walk exploration strategies). One way to obtain a direction at each time step is to consider a process where the next position depends on the previous two positions (see Fig. 3). However, this is undesirable, since we loose the Markov property, which is essential for deriving the hitting time distribution, which we show in the next section.

Instead, we let each node to express not only the position, but also the direction of a robot's motion [16] (see Fig. 4). With this representation, we have $D \times R \times C$ nodes, where D is the number of directions that can be encoded in the given representation, while R and C are the number of rows and columns in the arena, respectively. For the square grid, we use $D = 8$, and for the triangular grid, $D = 6$. By convention, we express the direction as an integer, $\xi \in \{0, 1, \ldots, D-1\}$, which represents a motion in the direction $\frac{2\pi}{D}\xi$ from the global reference line.

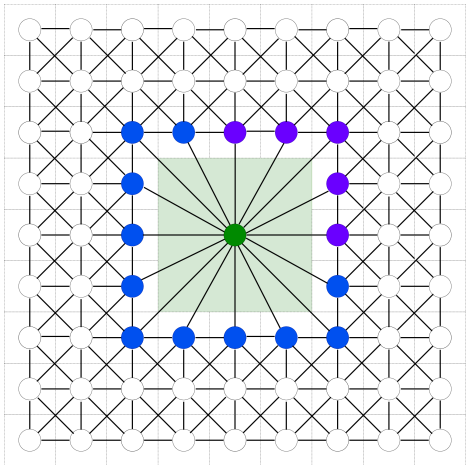


Fig. 5. Reduction of multiple goal nodes to a single node in a square grid representation. Green represents a goal state and goal region, blue a goal neighboring state.

To include a goal node in the representation, we first remove any nodes which are inside **G**. Then, we create a single node which connects all the neighboring nodes (see Fig. 5). We denote nodes that are adjacent to the goal node as *goal-neighboring nodes*. Instead of considering multiple nodes as a goal, reducing the goal to a single node simplifies the overall model and the computation of the hitting time. There could be multiple disconnected goal regions, but here we make the simplifying assumption that there is only one goal region.

Transition Probability: Let p_θ denote the probability that a robot turns left or right by θ radians, where $\theta \in [0, \pi]$. For an interior node i, we let the probability of moving from node i to node j to be $P_{i,j} = p_{|\theta_j - \theta_i|}$ where $\theta_i = \frac{2\pi}{D}\xi_i$ and $\theta_j = \frac{2\pi}{D}\xi_j$ (see Fig. 6).

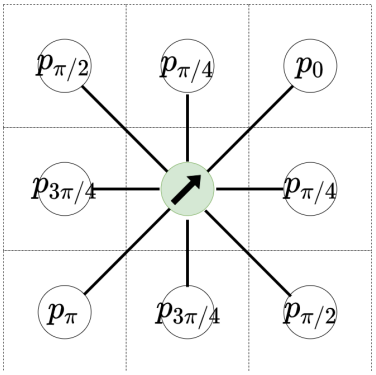
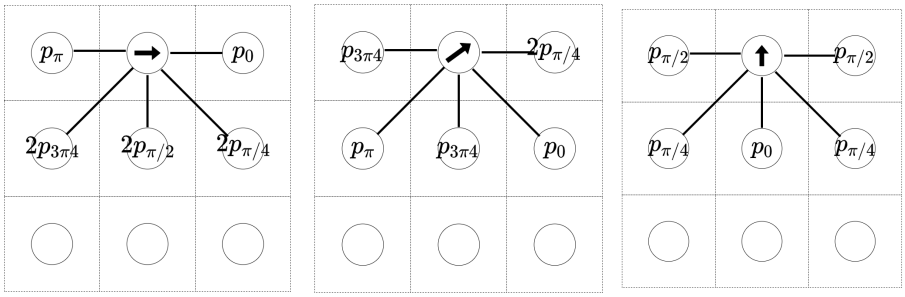


Fig. 6. Interior node transition probability. The arrow represents the direction of the robot, p_θ on the nodes denotes the probability of going there from the green node.

Due to physical restrictions, the robot's motion is persistent, which means that a robot tends to move smoothly without making sharp turns [21]. To capture this persistence, we add the following constraint: $p_\theta > p_\phi$ whenever $\theta < \phi$. In other words, a robot is more biased toward moving in the forward direction.



(a) Parallel (b) Angled (c) Perpendicular

Fig. 7. Transition probabilities for different cases when a robot is on a boundary node depending on the direction of motion.

For a boundary node i, there are three different cases. When the robot is moving along the boundary, the probability is assigned similarly to an interior node, except that the probability of turning is doubled so that the total probability adds up to 1 (see Fig. 7(a)). When the robot is heading toward $\pi/4$ or $\pi/2$, we add a constraint where the reflecting node has a higher probability to be the next state (see Fig. 7(b) and Fig. 7(c)).

Finally, goal neighboring nodes have a relatively higher probability of reaching the goal node. This is to capture the robot's local sensing capability, where the robot is more likely to move toward a goal when it is nearby. To make this

precise, let p be the probability of moving towards the goal node when the robot is on the goal neighboring node. Then p is assigned so that $p > 1-p$ is satisfied. The precise value of p depends on the local sensing capability.

5 Derivation of Hitting Time Distributions

We are now ready to use the arena representation to compute the distributions of hitting time. We start with the distribution for a single robot and then compute the distribution for multiple robots.

5.1 Hitting Time Distribution: Single Robot

Suppose that the model has M states. We wish to compute $f_{i,j}(t)$, which is the hitting time density distribution for moving from state i to j in t steps.

From [1], the following equality holds

$$P_{i,j}(t) = f_{i,j}(t) * P_{j,j}(t) = \sum_{t'=0}^{t} f_{i,j}(t')P_{j,j}(t-t') \tag{1}$$

where $*$ denotes a convolution operator.

By applying the Z-Transform and the convolution theorem for Z-Transform [23], it follows that

$$\hat{f}_{i,j}(z) = \frac{\hat{P}_{i,j}(z)}{\hat{P}_{j,j}(z)} \tag{2}$$

where

$$\begin{aligned} \hat{f}_{i,j}(z) &= \mathcal{Z}\left\{f_{i,j}(t)\right\}(z) = \sum_{t=0}^{\infty} f_{i,j}(t)z^{-t} \\ \hat{P}_{i,j}(z) &= \mathcal{Z}\left\{P_{i,j}(t)\right\}(z) = \sum_{t=0}^{\infty} P_{i,j}(t)z^{-t} \end{aligned} \tag{3}$$

Similar equations hold for $\hat{P}_{i,j}(z)$ and $\hat{P}_{j,j}(z)$. Finally, we obtain $f_{i,j}(t)$ by applying the inverse Z-Transform to Eq. (2):

$$f_{i,j}(t) = \mathcal{Z}^{-1}\left\{\frac{\hat{P}_{i,j}(z)}{\hat{P}_{j,j}(z)}\right\}(t) \tag{4}$$

While it is possible to compute $\hat{P}_{i,j}(z)$ directly by using a symbolic algebra system, doing so may take a significant amount of time. To reduce the computational time cost, we consider the case where $P_{i,j}$ is diagonalizable. The transition matrices derived from Section 4 (both for the square and triangular representations) are indeed diagonalizable.

From the assumption that $P_{i,j}$ is diagonalizable, we derive:

$$P_{i,j}(t) = (P^t)_{i,j} = (\widetilde{P}D^t\widetilde{P}^{-1})_{i,j} = \sum_{k=1}^{M} \widetilde{P}_{i,k}(\widetilde{P}^{-1})_{k.j}d_k^t \tag{5}$$

where D is a diagonal matrix and $d_k = D_{k,k}$. By the linearity of the Z-Transform,

$$\begin{aligned}\hat{P}_{i,j}(z) = \mathcal{Z}\left\{P_{i,j}(t)\right\}(z) &= \mathcal{Z}\left\{\sum_{k=1}^{M} \widetilde{P}_{i,k}(\widetilde{P}^{-1})_{k.j}d_k^t\right\}(z) \\ &= \sum_{k=1}^{M} \widetilde{P}_{i,k}(\widetilde{P}^{-1})_{k.j}\mathcal{Z}\{d_k^t\}(z) \\ &= \sum_{k=1}^{M} \widetilde{P}_{i,k}(\widetilde{P}^{-1})_{k.j}\frac{z}{z-d_k}\end{aligned} \tag{6}$$

Therefore, Eq. (4) can be written as

$$f_{i,j}(t) = \mathcal{Z}^{-1}\left\{\frac{\sum_{k=1}^{M} \widetilde{P}_{i,k}(\widetilde{P}^{-1})_{k.j}\frac{z}{z-d_k}}{\sum_{k=1}^{M} \widetilde{P}_{j,k}(\widetilde{P}^{-1})_{k.j}\frac{z}{z-d_k}}\right\}(t). \tag{7}$$

5.2 Hitting Time Distribution: Multiple Robots

From the assumption that the states of robots are i.i.d., at each time, it is clear that all robots have the same cumulative first passage time distribution, $\phi_{i,j}(t) = \sum_{\tau=0}^{t} f_{i,j}(\tau)$.

We can interpret $\phi_{i,j}(t)$ as the success probability of a robot reaching state j when starting from state i from time t. Hence, the probability that n robots have reached the goal, $\mathbf{G}$, by time t follows the binomial distribution with N 'trials' with the bias factor equal to $\phi_{i,j}(t)$:

$$P(n,t) = \binom{N}{n}(\phi_{i,j}(t))^n(1-\phi_{i,j}(t))^{N-n}. \tag{8}$$

When t is small, $\phi_{i,j}(t)$ is close to zero. As a result, $P(n,t)$ will be small. This corresponds to having robots that just started exploring, and so the probability of reaching the goal is small for any number of robots. In contrast, when t is large, $\phi_{i,j}(t)$ is close to one, and so $P(n,t)$ is close to one when $n = N$. This happens when robots explored for a long time and so all robots have reached the goal.

6 Experimental Setup

To assess the validity of the proposed method, we ran simulations using ARGoS [22] to obtain experimental data. We experimented with $N = 50, 100$, and

for each N we averaged 2,000 simulations to derive the empirical hitting time distributions.

To derive the theoretical distributions, we used the triangular grid model with $L_{\triangle} = 50, 100$. Instead of computing the exact values for $f_{i,j}(t)$, we used the standard Monte Carlo sampling method [24] to estimate $f_{i,j}(t)$. We could not compute the exact values for $f_{i,j}(t)$ since it required inverting large matrices.

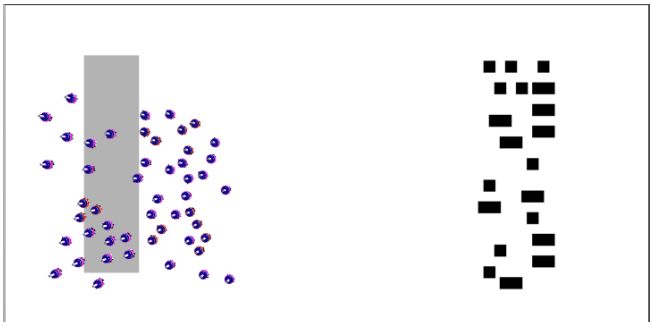


Fig. 8. ARGoS foraging simulation with 50 robots. The grey area is the collection location (nest), where robots return the objects, which are represented as black squares.

7 Results

The shapes of the experimental graphs in Fig. 9 suggest that the binomial distribution is suitable for approximating the actual hitting time distribution. There is a strong agreement between experimental and theoretical results for $P(15, t)$ and $P(25, t)$ with $R_{\triangle} = 50, C_{\triangle} = 100$. From these observations, it can be inferred that the i.i.d. assumption is appropriate. Specifically, the interactions between robots are negligible for computing the hitting time distribution.

By comparing the first and second row in Fig. 9, it is clear that the representation with more nodes gives better predictions. The improvement is most obvious for $P(25, t)$. This observation is more clear when looking at Fig. 10, where the errors between the theoretical and experimental results with 100 robots were computed using the $L2$ norm. When there are more nodes ($R_{\triangle} = 50, C_{\triangle} = 100$), the errors grow slower than when there are fewer nodes ($R_{\triangle} = 25, C_{\triangle} = 50$). Also, the maximum error for $R_{\triangle} = 25, C_{\triangle} = 50$ is 30, while the maximum error for $R_{\triangle} = 50, C_{\triangle} = 100$ is 120. From these results, we can infer that the number of nodes has a big impact on the accuracy of the predictions.

8 Conclusions and Future Work

In this work, we modelled swarm exploration using a Markov chain. To express the persistence of robot motions in correlated random walks, we added states

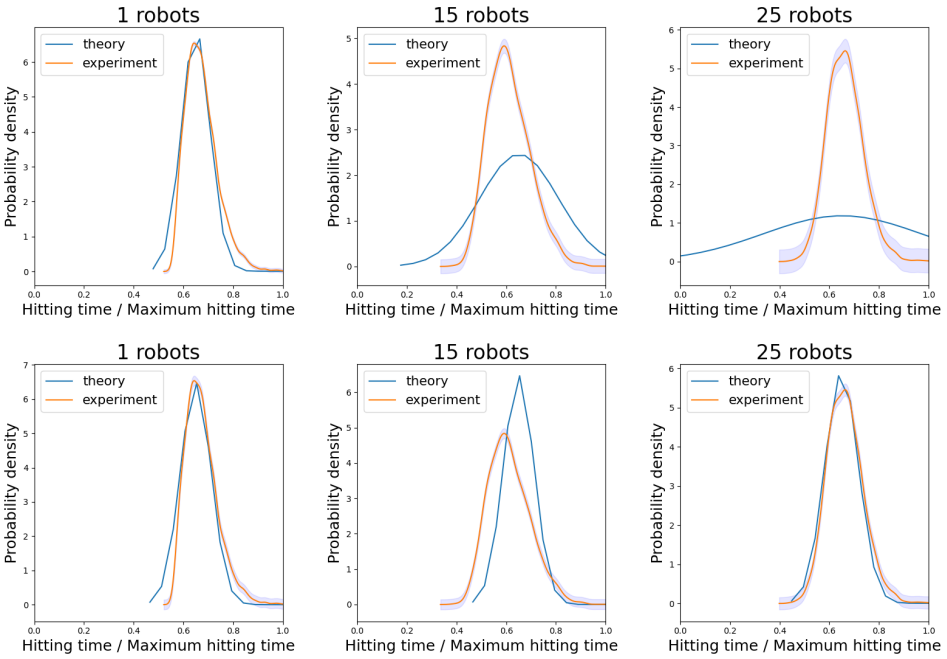


Fig. 9. Experiments with $N = 50$ robots using the triangular space representation. Graphs on the first row used $R_{\triangle} = 25, C_{\triangle} = 50$ for the theoretical model, on the second row $R_{\triangle} = 50, C_{\triangle} = 100$. Each column shows $P(n, t)$ for $n \in \{1, 15, 25\}$. The x-axis represents the hitting time normalized by the maximum hitting time, the y-axis represents the probability density. Experimental graphs have 95% confidence intervals.

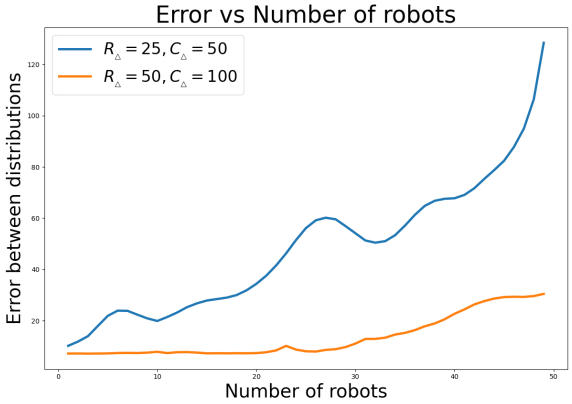


Fig. 10. Errors between theoretical and experimental results for two different model sizes up to 50 robots. The errors are computed using the $L2$ norm.

which encode the direction of the robot motion. Using our model, we obtained a single robot hitting time distribution, and we used it to further derive a hitting time distribution for multiple robots. Finally, we experimentally verified using the ARGoS simulator that our model is predictive.

To extend our work we will consider practical applications. As an example, one can use our model and hitting time distributions as a performance measure for an exploration strategy. Moreover, computing the exact value of $f_{i,j}(t)$ has the potential to increase the accuracy of the theoretical results more than using the standard Monte Carlo sampling method to approximate them. To do this, it is important to derive a tractable numerical method for computing $f_{i,j}(t)$.

Currently, our work is limited to an environment with no obstacles and a single goal area. Also the robots are assumed to work without any failure (e.g., no wheels or hardware failures, no battery depletion). In more real settings, those are important factors to consider, and hence one way to expand our work is to build a more realistic model of the robots and the arena.

Finally, the i.i.d. assumption may not be valid when the interactions between robots are significant, which is the case when the ratios of the arena to robot size and of the arena size to the number of robots are small. Likewise, if robots are close to each others at the beginning and the goal location is nearby, the prediction will not be accurate as there may not be enough time for the robots to spread out so that the interaction becomes negligible. One direction of study is to identify when the assumptions fail both theoretically and experimentally.

Acknowledgments: We gratefully acknowledge the MnDRIVE initiative, the Minnesota Robotics Institute, the Minnesota Supercomputing Institute, and the Undergraduate Research Opportunities Program at the University of Minnesota for their support.

References

1. Abate, J., Choudhury, G.L., Whitt, W.: An introduction to numerical transform inversion and its application to probability models. In Grassmann, W.K., ed.: Computational Probability. Volume 24 of International Series in Operations Research & Management Science. Springer, Boston, MA (2000)
2. Adams, A.L., Schmidt, T.A., Newgard, C.D., Federiuk, C.S., Christie, M., Scorvo, S., DeFreest, M.: Search is a time-critical event: When search and rescue missions may become futile. Wilderness & Environmental Medicine **18**(2) (2007) 95–101
3. Alon, N., Avin, C., Koucky, M., Kozma, G., Lotker, Z., Tuttle, M.R.: Many random walks are faster than one. In: SPAA '08, New York, NY, USA, ACM (2008) 119–128
4. Amjadi, A.S., Raoufi, M., Turgut, A., Broughton, G., Krajník, T., Arvin, F.: Cooperative pollution source localization and cleanup with a bio-inspired swarm robot aggregation. arXiv:1907.09585v1 [cs.RO] (2019)
5. Arnold, R., Yamaguchi, H., Tanaka, T.: Search and rescue with autonomous flying robots through behavior-based cooperative intelligence. Journal of International Humanitarian Action **3** (December 2018)
6. Couceiro, M.: An overview of swarm robotics for search and rescue applications. In Tan, Y., ed.: Design, Control, and Modeling of Swarm Robotics. IGI Global (2016) 345–382

7. Dong, Q., Cui, L.: First hitting time distributions for brownian motion and regions with piecewise linear boundaries. Methodology and Computing in Applied Probability **21** (2019) 1–23
8. Fujisawa, R., Dobata, S.: Lévy walk enhances efficiency of group foraging in pheromone-communicating swarm robots. In: IEEE/SICE International Symposium on System Integration, SII 2013. (December 2013) 808–813
9. Harrison, P.G., Knottenbelt, W.J.: Passage time distributions in large Markov chains. In: Proc. Int'l Conf. on Measurement and Modeling of Computer Systems (SIGMETRICS), Association for Computing Machinery (2002) 77–85
10. Harwell, J., Gini, M.: Broadening applicability of swarm-robotic foraging through constraint relaxation. In: 2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR). (2018) 116–122
11. Harwell, J., Gini, M.: Swarm engineering through quantitative measurement of swarm robotic principles in a 10,000 robot swarm. In: Proc. 28th Int'l Joint Conference on Artificial Intelligence, (IJCAI-19), IJCAI (7 2019) 336–342
12. Hunter, J.J.: The computation of the mean first passage times for markov chains. Linear Algebra and its Applications **549** (July 2018) 100–122
13. Levin, D.A., Peres, Y., Wilmer, E.L.: Markov Chains and Mixing Times. Volume 107. American Mathematical Soc. (2017)
14. Martinez, F., Jacinto, E., Acero, D.: Brownian motion as exploration strategy for autonomous swarm robots. In: 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO). (2012) 2375–2380
15. Megalingam, R.K., Nagalla, D., Kiran, P.R., Geesala, R.T., Nigam, K.: Swarm based autonomous landmine detecting robots. In: 2017 International Conference on Inventive Computing and Informatics (ICICI). (2017) 608–612
16. Navarrete Michelini, P., Coyle, E.: Mobility models based on correlated random walks. In: Proc. Int'l Conf. on Mobile Technology, Applications, and Systems (Mobility). (2008) 1–8
17. Navarro, I., Matía, F.: An introduction to swarm robotics. ISRN Robotics **2013** (September 2012)
18. Nyberg, M., Ambjörnsson, T., Lizana, L.: A simple method to calculate first-passage time densities with arbitrary initial conditions. New Journal of Physics **18**(6) (June 2016)
19. Pang, B., Qi, J., Zhang, C., Song, Y., Yang, R.: Analysis of random walk models in swarm robots for area exploration. In: 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO). (December 2019) 2484–2489
20. Pang, B., Song, Y., Zhang, C., Wang, H., Yang, R.: A swarm robotic exploration strategy based on an improved random walk method. Journal of Robotics **2019** (March 2019) 1–9
21. Patlak, C.: Random Walk with Persistence and External Bias: A Mathematical Contribution to the Study of Orientation of Organisms. University of Chicago, Committee on Mathematical Biology (1953)
22. Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., Birattari, M., Gambardella, L.M., Dorigo, M.: ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. Swarm Intelligence **6**(4) (2012) 271–295
23. Smith, J.O.: Introduction to Digital Filters with Audio Applications. W3K Publishing (2007) http://www.w3k.org/books/.
24. Taverniers, S., Tartakovsky, D.M.: Estimation of distributions via multilevel Monte Carlo with stratified sampling. Journal of Computational Physics **419** (October 2020) 109572