Swarm Diffusion-Taxis: Transport of spatial information for cooperative gradient-based navigation

Conference Paper · January 2022

3 authors:

Emma Milner
University of Bristol

Mahesh Sooriyabandara
Toshiba Research Europe Limited

Sabine Hauert
Massachusetts Institute of Technology

Swarm Diffusion-Taxis: Transport of spatial information for cooperative gradient-based navigation

Emma Milner[1†], Mahesh Sooriyabandara[2] and Sabine Hauert[1]

[1]University of Bristol, United Kingdom
(E-mail: emma.milner@bristol.ac.uk)
[2]Toshiba Europe Limited, United Kingdom

Abstract: Swarm Diffusion-Taxis is a new algorithm for navigation of unknown environments to areas of interest. The algorithm disperses robots into an unmapped space using random walk and robots communicate locally how long ago they were in the area. Because the robots spatially diffuse, this timer estimates radial distance to the area. This creates a gradient of spatial information which can be used by robots to navigate. It is shown in simulation of ground-based robots that this creates a successful taxis effect. An intralogistics use case is simulated which requires the delivery of items to a user and compares the time taken with a fixed and dynamic area of interest. The time performance is similar to a global gradient algorithm (using a solar compass) and a connected communication algorithm (hop-based navigation). The benefits of minimal set-up and requirements, mean that robots could be cheap, simple to maintain and deployed out-of-the-box.

Keywords: Swarm intelligence, Swarm robotics, Self-organisation, Intralogistics

1. INTRODUCTION

A new swarm algorithm known as Swarm Diffusion-Taxis (DT) is presented to facilitate the navigation of unmapped areas. The aim of the algorithm is to use the swarm to find an area of interest (AOI) in a space where the robots know nothing about their environment and are told nothing by an observer system. The control is distributed to include the inherent benefits of swarm intelligence which include robustness, scalability and adaptability [1]. The key principle relies on diffusive motion (random walk) and their recording of time elapsed from the visitation of an area of interest. This time taken is an estimate only (no odometry data is recorded) of radial distance from the area of interest. The spatial diffusion of these decaying timers creates a gradient that can then be navigated simply by moving towards those with lowest timers. There is no path planning or global communication needed and the algorithm only requires local, single messages to be exchanged between robots. Other techniques that provide gradients for navigation either rely on environmental gradient information (e.g. light sources, sound or electromagnetic fields that can be read by a robot sensor), or hop-based navigation that rely on constant connection throughout a communication network. The same effects could be achieved with the Swarm DT algorithm without the requirement of connectivity or globally available information. Indeed, in Swarm DT, robots only rely on their individual diffusive motion to spread spatial information. This is exemplified in Fig. 1 which has examples of global gradient, connected communication and diffusion based (i.e. Swarm DT) communication at work. In the diffusion case, the robots who have been to the AOI in the previous 10 s are displayed with their timer numbers and the ones who have never been to the area have none displayed. The range of timer values is in between 0-500 to indicate how recently they have been in the area. When they exit the area, the timer increases from 0 by 1 every 0.02 s and stops broadcasting after 10 s (once the timer reaches 500). The robot with the symbol $>$ is indicating that it is choosing to move towards another robot and we can see that this is the correct direction to move to get closer to the area. The robot makes this decision based on its communication with its two local neighbours, the only other robots within sensory range. One has not seen the area, the other has seen the area recently. The navigating robot can compare the two and knows to move towards the lower timer to increase the likelihood of moving closer to its goal. However, if the same robot were to be using a connected communication method, such as hop-based navigation, then the robots who are furthest from the area have not yet been able to receive any spatial information because they are too far from the area to be part of the connected robots. This is also shown in the diagram in Fig. 1. The connected robots have formed a connected china in order of proximity (i.e. hop-count) to the area which is guiding the robot with label $>$ to move towards the area but the robots with the '?' symbol have no spatial information. If there are no robots in the area at all then there is no connection at all and no spatial information being shared at all. Fig. 1 also shows how the task could be completed using global communication methods, such as following a light intensity gradient to a light source within the area. All robots in the space can access the gradient information which can guide them to the area. The requirements of this method however are that all the robots can access globally available information which requires the space to have pre-existing, global communication infrastructure.

A use case for the Swarm DT algorithm could be the delivery of items in an unmapped storage space. Here the AOI is the delivery area of a warehouse and the robots are required to disperse throughout the unmapped space, with no inventory list, to find items for the user. An example of such a task is included in Jones et al. (2020) which provides further information about a swarm storage and retrieval system [2]. Ideally, a swarm could create a system for this purpose

† Emma Milner is the presenter of this paper.

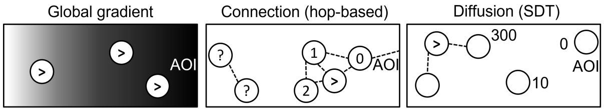

Fig. 1. Diagrams show the global gradient, connection (hop-based) and diffusion (Swarm DT) communication types: Global uses environmental gradients; Connection communication uses a chain of a sequence of hop-counts (displayed as numbers increasing from 0); Diffusion (Swarm DT) uses timers which state how recently the robot has been in the AOI, to guide the robot seeking the AOI (displaying >). The dotted lines indicate a communication occurring.

which is usable out-of-the-box, requiring no previous infrastructure, mapping, inventory list or set-up of any kind. The benefits that the Swarm DT algorithm bring to this task including simple, local communication and no infrastructure requirements mean that it could facilitate an out-of-the-box retrieval system for such a use case.

2. LITERATURE REVIEW

Swarm Diffusion-Taxis (DT) creates a gradient of spatial information based on diffusive transport of information from areas of interest (AOIs) that are accessed locally to each robot and can be used for navigation of unmapped areas. Global communication has been used previously to convey a spatial gradient to robots using a source located in the AOI which emits a signal (e.g. odor [3], light [4], temperature trails [5]) that can be accessed by every robot within range. They follow their own measurements of this source signal up a gradient of signal intensity until they reach the area. This global communication method is often based on bio-inspiration such as by the solar compass used by ants [6]. Sugawara et al. (2004) projects graphics on the floor which replicate diffusing chemical signals for the robots to follow towards a home area [7]. These techniques are dependent on their environment which make them less robust if that one guiding system fails and also requires set up which makes the system usable only when the necessary infrastructure is in place, limiting application scenarios. Hop-based navigation is a method of navigation in unknown spaces but does not require a method of global information sharing to be set up. In this, robots with distributed control overcome their limited transmission ranges (i.e. no global communication) by exchanging local messages which can be forwarded around the swarm, extending information beyond a single robot's communication range. These multiple passes of messages throughout the network are known as "hops" [10]. Robots in the swarm act as nodes in a communication network. robots who discover an area of interest, known as seed robots, advertise themselves as a '0' [8]. Every robot within range of the seed robot then increments their own hop-count from the source by 1. This continues throughout the network with robots always adding 1 to the smallest hop-count they sense in their neighbourhood, creating a shortest path that can be navigated to the area of interest. This has been successfully used on real robots including an aerial swarm [9] and a swarm of 1000 kilobots [8].These methods are found to be robust [9] but do not extend information to robots outside of communication range of the connected robots creating the shortest path. This reduces the ability of robots to explore further afield places and still be able to access spatial cues which can guide them back to an area of interest. In the intralogistics use case robots must first search the warehouse by diffusing to every corner to find items before they navigate to the deliver area. However if they were far away from the connected robots or there is no seed robot at all (because none are currently in the AOI) then no spatial information will be being shared.

Other communications methods are similar to Swarm DT because they have inform navigation through diffusive methods of information storage. For example, Ducatelle et al. (2014) also forwards information around a swarm, propagating out from a target area to guide robots along a diffusive gradient. Their cooperative navigation algorithm used one robot known as the Target robot who, once it found the area of interest, would remain stationary and send periodic broadcasts to its neighbours. Each robot keeps a table of other robots' IDs and their estimated distance to the target robot as well as the age of this information. Robots would communicate their logs with local neighbours and update their own information based on the age of other robot's data and their own on-board odometry calculations [11]. The Swarm DT algorithm uses similar methods to Ducatelle et al. but has less on-board computation or data storage requirements and no need for recording each robot's own odometry. There is an unmet need for a retrieval system which is usable out-of-the-box. In a series of interviews done by Carrillo-Zapata et al. (2020) it was found that there were many small business use cases where sorting of items in storage was considered a laborious task which would benefit from automation. However these places could not afford the time, money or space required for the sorting systems currently on the market. The advantages of scalability, robustness and adaptability brought by the Swarm DT algorithm make it a useful alternative for these use cases.

3. METHODOLOGY

The following section details the Swarm Diffusion-Taxis (DT) algorithm and its implementation in a robot simulation. The codes for this simulation can be accessed here `https://bitbucket.org/hauertlab/workspace/snippets/A9ygaR` Algorithms for global gradient communication (solar compass method) and a connected communication algorithm (hop-based navigation) are also tested in the same simulated conditions and compared to the Swarm DT results.

3.1. Swarm Diffusion-Taxis algorithm

The Swarm DT algorithm uses random walk to disperse the robots around an unmapped area. If there are no nearby objects and the robot is not seeking the area of interest (AOI) then it will move with random walk by adding a random perturbation, p, in the range $-0.5rad \leq p \leq 0.5rad$ to its current heading direction every time step. When the robot is within the AOI, it will broadcast a timer value T_i of 0 to robots within its sensory range. When it leaves the area (via random walk), this timer will increase by 1 every time step (0.02 s) for 10 s. This is part of the algorithm described by Fig. 2 which also states how if a robot is actively seeking the AOI (for example, if it has an item to deliver to a user), then it will record the timers of each of the neighbours, T_n, within its sensory range. From these numbers, the robot will select the neighbour with the lowest timer and move its current heading so it moves towards this neighbour (making H_{Ti} non-zero, see Eq. (4)). Each robot will only receive timers from neighbours if they are actively looking for the AOI to avoid communicating when not necessary.

3.2. Simulated use case set-up

The simulation used here is physics-based in 2D, written in Python. It is based on an intralogistics use case in which there are items within the unmapped space which must be picked up by ground robots and delivered to a delivery area (the AOI), where the user is waiting to receive them. The storage space, known as the warehouse, is a bounded 5 m by 5 m square. Screenshots describing the simulation environment can be found in Fig. 3. Here it can be seen that there are two versions of this task tested, one with a fixed AOI and one with multiple, dynamic AOIs. In the fixed case, the area is between width = 400 cm and 500 cm, stretching the full length of the height axis, following the convention given in Fig. 3. The aim of the fixed area task is to deliver all the boxes to the area in the shortest time, in no particular order. In the dynamic area case, the area is a circle of radius 100 cm whose centre can appear at any point on the warehouse walls. There is only one box in the warehouse that can be delivered to this particular area (this is the black box seen in Fig. 3). Once this correct box has been delivered, the AOI location will change (as seen in the Figure from the transition between Time is 5 to 9 to 39 s) and the specific box ID to be delivered will also change. The aim of the dynamic AOI

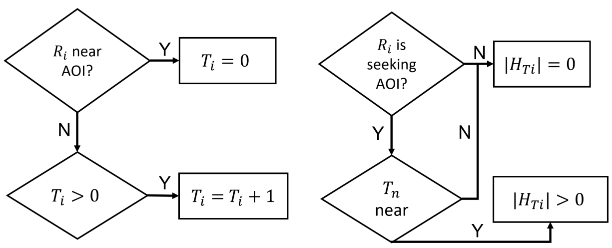

Fig. 2. Swarm Diffusion-Taxis algorithm. Robot R_i has timer value T_i and can read other robot timers T_n within sensory range. If the robot is seeking the area of interest (AOI) then it will be attracted to nearby robots with low timers. This creates the heading H_{Ti} which is added to the robot heading (see Eq. (4)). If the robot is in the AOI it will have a timer of 0 and increase by 1 every time step it is outside the AOI.

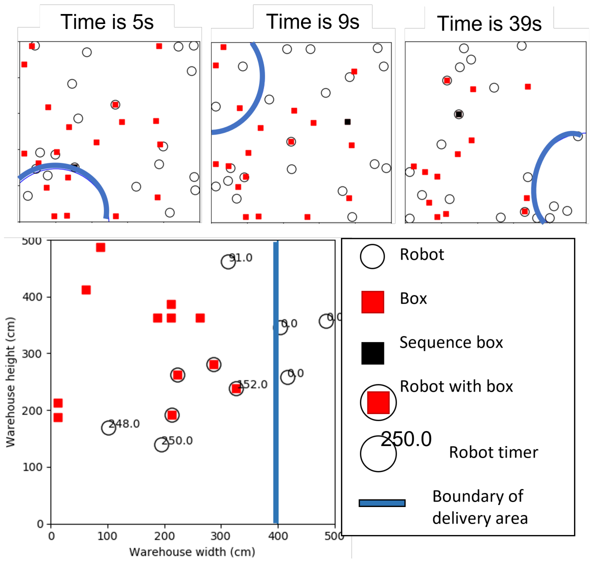

Fig. 3. Screenshots of the simulation. In the fixed AOI task the boundary of the AOI is a straight line and the area remains static. For the dynamic task, the black box must be delivered to a circular area which will change position once it is delivered and a new box is now the next to be delivered. The Swarm DT algorithm is illustrated here, with the timers shown.

task is to deliver all the boxes to their particular area in the shortest time total but the boxes cannot be delivered in parallel and must be delivered in a given sequence. This mimics a queue of individual requests from the user for specific items from anywhere in the unmapped warehouse.

Background warehouse delivery algorithm The robots pick up and put down boxes in the simulation, performing a background task used to test the Swarm DT algorithm on a real application. In the dynamic AOI case, if the robot does not have the correct box for the area that is currently showing then there is a probability factor of 0.03 every time step that the robot will drop their box where they are. This keeps the robots reshuffling the boxes in this task to avoid a deadlock. This reshuffling does not happen in the fixed AOI case in which boxes can only be dropped once the robot carrying it is within the AOI. Dropped boxes in the correct area are instantaneously removed from the warehouse by the user.

Robot motion model The robot radius is 12.5 cm which is the same size as the box radius. The robot moves with speed, S_p equal to 100 cm/s. The update frequency of each robot is once every 0.02 s which equals the time step (t_s = 0.02 s). The range at which a robot can detect a box is 25 cm which is when the boxes and robots are physically touching. The sensory range for object recognition, used for collision avoidance, is 35 cm. The robots are able to communicate their timers over a range of 150 cm. All the sensory and communication ranges are measured from the centre of the robot to the centre of the object. The equation of motion for each robot is given in Eq. (1). In this, the robot heading $\vec{H_{t+t_s}}$ at time $t + t_s$ is the sum of column vector headings due to each of: random walk ($\vec{H_{noise}} = \vec{H_t} + [\cos(p), \sin(p)]$, where p is a perturbation in the range $-0.5rad \leq p \leq 0.5rad$ and $\vec{H_t}$ is the heading in the previous time step); collision avoidance Eqs. (2), (3); and timer readings Eq. (4).

$$\begin{bmatrix} X \\ Y \end{bmatrix}_{t+t_s} = \begin{bmatrix} X_t + Sp * t_s * cos(H_{(t+t_s),x}) \\ Y_t + Sp * t_s * sin(H_{(t+t_s),y}) \end{bmatrix} \tag{1}$$

Collision avoidance is applied to the robot heading when there is an obstacle within the robot's sensory range (SR = 35 cm). A heading pointing the robot away from the obstacle is generated, described in Eq. (2) for other robots and boxes and Eq. (3) for the walls of the warehouse. Eq. (2) uses the distance (d_r and d_b) to the obstacle in x and y measured from the centre of the robot to the centre of the obstacle. It will only include the distances of obstacles that are within the sensory range ($|d| < SR$) of the robot. The signs of the vectors to the obstacles are inverted to point the robot away from the obstacle. If there are no obstacles within the robot's sensory range then $\vec{H_{r,b}} = [0, 0]$.

$$\vec{H_{r,b}} = \vec{H_r} + \vec{H_b} = -\sum \begin{bmatrix} d_{r,x} \\ d_{r,y} \end{bmatrix}_{<SR} - \sum \begin{bmatrix} d_{b,x} \\ d_{b,y} \end{bmatrix}_{<SR} \tag{2}$$

$\vec{H_w}$ is non-zero when the robot is within a radius length of the wall (i.e. it is touching the wall). In this case, the heading generated has magnitude 100 in the direction perpendicular to the wall. The factor of 100 is used so that if the robot is touching a wall it will not go through it to e.g. avoid other robots because $\vec{H_w}$ will always be greater than $\vec{H_r}$. In Eq. (3), letters W, E, S and N stand for West (x = 0), East (x = 500), South (y = 0) and North (y = 500), for the four walls. This follows the coordinate conventions given in Fig. 3. When the robot is touching e.g. the West wall, then W = 1. When it is not touching e.g. the South wall, then S = 0, and so on.

$$\vec{H_w} = 100 * \begin{bmatrix} (W - E) \\ (S - N) \end{bmatrix} \tag{3}$$

Finally, Eq. (4) describes the attractive behaviour that occurs when a robot with a box that is seeking the delivery area comes into sensory range of a robot which is broadcasting a timer value. In this case $\vec{H_T}$ will be non-zero and will be the vector towards the robot with the lowest timer within that navigating robot's sensory range. This vector is described by d_T which is a column vector that describes the distance between the two robots. The vector magnitude is reduced by including the 0.01 factor in Eq. (4) so that the timer heading does not become dominant over collision avoidance headings ($\vec{H_{r,b,w}}$).

$$\vec{H_T} = 0.01 * \begin{bmatrix} d_{T,x} \\ d_{T,y} \end{bmatrix}_{Tmin} \tag{4}$$

4. RESULTS

The following describes the performance results for the Swarm Diffusion-Taxis (DT) algorithm when tested in simulation. A fixed area of interest (AOI) and a dynamic AOI are both tested for a range of numbers of robots (N_r) and boxes (N_b). Each time taken results is an average of 10 trials with these parameters. The simulation was also used to test two other algorithms for comparison, under the same conditions. The first was a global communication method which was based on a solar compass. A light source was simulated as being in the AOI which could be sensed by the robots from any position within the warehouse. The robots follow a global gradient of light intensity to lead them directly to the area. Occlusions and the physics of light were not modelled in the simulation. The second uses hop-based navigation to form a chain of hop-counts leading to the area. Any robot within the area becomes a seed robot but no robot becomes stationary. Random walk is also tested whereby the robots follow the same random heading perturbation rules as the other algorithms but with no communication between agents or environmental sensing.

4.1. Single, fixed area of interest

For the single, fixed AOI the times taken by the global, hop-based and Swarm DT algorithms are given as heatmaps in Fig. 4, each of which is on a scale of 10 to 260 s, representing the time taken to deliver all the boxes. The random walk results are given on a scale of 95 to 430 s. This shows that the Swarm DT is having a positive effect on the performance and produces results that are closer to the global gradient and hop-based algorithms compared to random walk. The performance for the Swarm DT algorithm is worse than the global gradient algorithm (and the hop-based in many cases) but are comparable to the hop-based method. The Swarm DT performance improved for each given N_b, by increasing N_r. This is also true for the other algorithms. Using a swarm of $N_r = 50$, the Swarm DT algorithm took 27.8 s to collect 10 boxes and 53.6 s to collect 50 boxes. These are much better than the time recorded for pure random walk which was 184.6 s to collect 10 boxes and 307.0 s for 50 boxes, using 50 robots. The Swarm DT times are are closer to the times recorded for hop-based navigation and the global algorithm which are as follows: hop-based navigation took 15.8 s to collect 10 boxes and 49.8 s to collect 50 boxes, using 50 robots; the global algorithm took 13.0 s to collect 10 boxes and 41.6 s to collect 50 boxes, using 50 robots. The similarity in times and patterns, particularly at larger swarm sizes, seen between the Swarm DT and the other navigation algorithms suggest that a taxis effect is being successfully created. The swarm size was increased to 150 to test the limits of these results. The results for 10-150 robots collecting 50 boxes are given in Fig. 5. The average results are given as a line and all results across the 10 trials are shadows surrounding the average. Swarm DT average times significantly increase beyond 105 robots. The hop-based and global gradient algorithms also have increased averages beyond 115 robots. The increase in hop-based navigation times is much lower than Swarm DT but the increase in global gradient times are very similar to the Swarm DT algorithm. The range of times seen across the 10 trials vastly increases for Swarm DT algorithm beyond 125 robots and the same effect is seen for the global gradient algorithm but significantly less so for the hop-based navigation. These increases in variability of results and in average time are due to crowding in the delivery area which physically blocks new robots from reaching the area. Increasing robot numbers from 100 to 150 increases the average time taken to collect 50 boxes from 74.4 s to 404.9 s using Swarm DT, 65.3 s to 132.9 s using hop-based navigation and 65.4 s to 336.9 s using the global gradient algorithm. The crowding causes the Swarm DT robots to not be able to re-enter the warehouse from the delivery area to spread their diffusing spatial information which keeps the times high at high robot numbers. The global gradient sees high times at high robot numbers also due to crowding but instead it is because there is reduced random walk in the algorithm because the return to the delivery area is so one-directional. This prevents the robots from navigating around each other when trying to get into the crowded delivery area because the navigating robots will not move out of the way with as much range of movement as in hop-based navigation or Swarm DT. Hop-based times are less affected by crowding because the spatial information is spread through connection which provides the directed return information without the need for robots to physically move from one end of the warehouse to the other, so is not affected by crowding. It is actually helped by crowding because this information propagates more easily through closely packed robots. It also includes more random walk in the directed return path than the global gradient case because the robots move towards the area through robot-robot attraction rather than a direct compass measurement which allows them to move around each other more easily to navigate blockages. A figure is not included of these behaviours because it is too difficult to see from a static image. However a link to the code used for these experiments is included at the start of this section which can be run by readers and the effects of crowding can be seen.

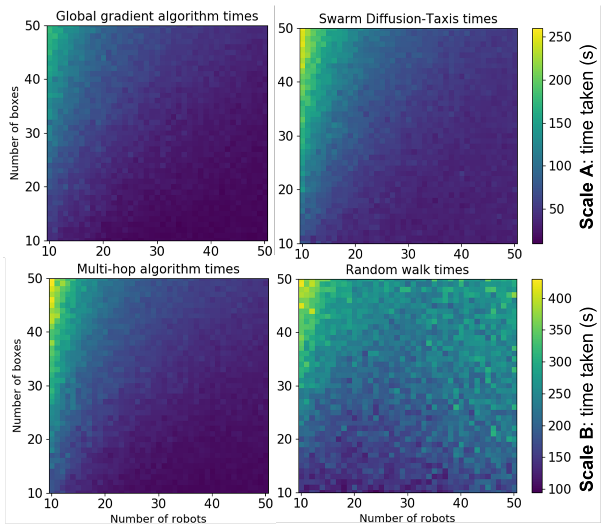

Fig. 4. Heatmaps showing the average time taken on Scale A (10-260 s) for the Swarm Diffusion-Taxis, global gradient and hop-based algorithms and Scale B (95-430 s) for random walkers. 10-50 boxes and robots are tested.

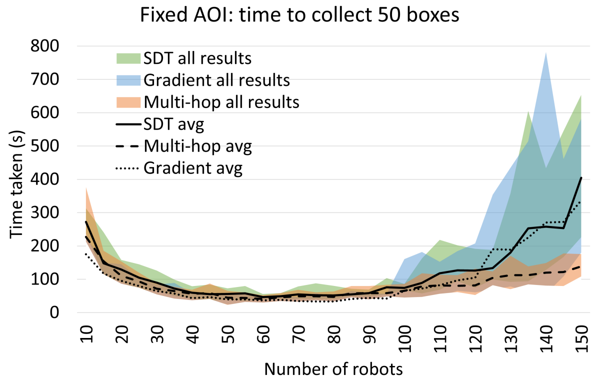

Fig. 5. The results for average time and all results over 10 trials display the time to collect 50 boxes in the fixed AOI task for 10-150 robots. There is crowding has at higher numbers of robots which negatively affects performance.

4.2. Multiple, dynamic areas of interest

The second task tested used multiple areas of interest which, when found, changed position during the task. Boxes could only be delivered in a given sequence of box IDs and each to its own specific AOI. The dynamic AOI task was tested using N_b = 10 for $10 \leq N_r \leq 50$. Each swarm size was tested 10 times and the full range of times seen in these trials is given as a shadow of values behind the average time in Fig. 6. The Swarm DT times are longer than hop-based navigation and global algorithms for this task. However the average times seen are close to the other algorithms which is a good performance for the Swarm DT algorithm. The maximum difference in average time between the Swarm DT and the global gradient method is 156.1 s (for N_r = 10) with the minimum average time difference being just 16.78 s (for N_r = 35). The differences between the Swarm DT and hop-based navigation are maximum at 80.1 s (for N_r = 25) and minimum at 21.6 s (for N_r = 10). As N_r increases from 10 to 20, the average times for Swarm DT and the global gradient method become more similar (as Swarm DT times decrease more significantly than global gradient times) whereas the Swarm DT and hop-based navigation times grow further apart (hop-based times continue to decrease while Swarm DT plateau). The times taken for the Swarm DT algorithm improve by increasing robot numbers from 10 to 30 robots however increasing swarm size beyond 30 robots does not see an improvement in time. This suggests that in the case of 10 boxes to collect, 30 is the best and most efficient swarm size because adding more robots to the swarm does not improve performance. The most efficient number of robots (where times steady to an approximately constant value despite increasing swarm size) for both hop-based and the global gradient algorithms is 20 robots which is fewer than for the Swarm DT algorithm. The approximate average times that the algorithms settle to for these last 30-50 robot numbers is 120 s for Swarm DT, 70 s for hop-based and 40 s for the global algorithm. The range of times over all 10 trials is much more varied for the Swarm DT algorithm, compared to

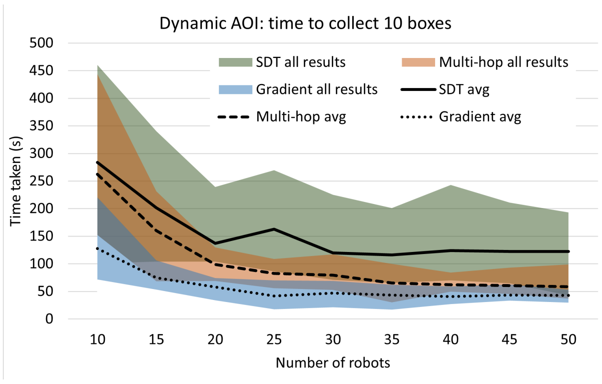

Fig. 6. Line graph showing the results across all 10 trials as a shadow surrounding the average time taken line for each of the algorithms tested. The results display the time taken to collect 10 boxes in the dynamic AOI task for 10-50 robots.

the hop-based and the global gradient algorithms.

5. DISCUSSION

The similarity in times and patterns between Swarm Diffusion-Taxis and the other navigation algorithms suggest that a taxis effect is being successfully formed. The similarity in times is less so for the dynamic AOI case but is still within a reasonable range compared to the other given algorithms which suggests that a taxis effect to the area is still being created even in this more complex task. These performances suggest that a diffusion gradient of spatial information is created and communicated by the swarm to navigate the unmapped environment to find the area of interest. It was found that increasing the swarm size improves the performance but is limited by crowding effects as the number of robots passes a certain density within the given warehouse space. For this reason, there is likely an ideal robot density for each task and a given number of boxes which could be found to optimise efficiency in this use case. For the dynamic AOI task, a larger range of times are seen for the Swarm DT algorithm than the others which suggests that it is less consistent in performance. The relatively low average time (compared to the maximum time seen in the full range of 10 trails) suggests that it is prone to a small percentage of much higher times which could make it less reliable in application.

6. CONCLUSION

The Swarm Diffusion-Taxis algorithm has been shown in simulation to cause the spread of spatial information throughout the swarm which leads robots towards an area of interest. This does not rely on global communication or for the robots to be connected to many other robots to create a successful taxis effect towards the area. This means that the algorithm can be used on cheap robots with simple communication hardware and does not require complex infrastructure or set-up. Due to these minimum requirements, the algorithm is likely to be useful for use cases which require a sorting system that is usable out-of-the-box.

REFERENCES

[1] M. Brambilla, E. Ferrante, M. Birattari et al. "Swarm robotics: a review from the swarm engineering perspective", *Swarm Intelligence*, Vol. 7, pp. 1–41, 2013.

[2] S. Jones, S. Hauert, E. Milner, M. Sooriyabandara. "Distributed Situational Awareness in Robot Swarms", *Advanced Intelligent Systems*, 2020.

[3] A. Marjovi, J. Nunes, P. Sousa, R. Faria and L. Marques, "An olfactory-based robot swarm navigation method," *IEEE International Conference on Robotics and Automation*, pp. 4958-4963, 2010.

[4] A. Winfield and J. Nembrini, "Emergent Swarm Morphology Control of Wireless Networked Mobile Robots", *Morphogenetic Engineering: Toward Programmable Complex Systems*, pp. 239–271 2012.

[5] A. R. Russell, "Heat trails as short-lived navigational markers for mobile robots," *Proceedings of International Conference on Robotics and Automation*, Vol. 4, 1997.

[6] R. Fujisawa, S. Dobata, K. Sugawara, F. Matsuno. "Designing pheromone communication in swarm robotics: Group foraging behaviour mediated by chemical substance", *Swarm Intelligence*, Vol. 8, pp. 227–246, 2014.

[7] K. Sugawara, T. Kazama and T. Watanabe,"Foraging behavior of interacting robots with virtual pheromone," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vol. 3, pp. 3074–3079, 2004.

[8] M. Gauci, M. E. Ortiz, M. Rubenstein, and R. Nagpal, "Error cascades in collective behavior: a case study of the gradient algorithm on 1000 physical robots", *In Proceedings of the 16th Conference on Autonomous robots and Multirobot Systems*, pp. 1404–1412, 2017.

[9] S. Hauert, L. Winkler, J. C. Zufferey, et al. "Ant-based swarming with positionless micro air vehicles for communication relay", *Swarm Intelligence*, Vol. 2, pp. 167-–188, 2008.

[10] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 85–97, 1998.

[11] F. Ducatelle, G. A. Di Caro and A. Förster, et al. "Cooperative navigation in robotic swarms", *Swarm Intelligence*, Vol. 8, No. 1, pp 1-–33, 2014.

[12] D. Carrillo-Zapata, E. Milner and Julian Hird et al., "Mutual Shaping in Swarm Robotics: User Studies in Fire and Rescue, Storage Organization, and Bridge Inspection", *Frontiers in Robotics and AI*, Vol. 7, No. 53, 2020.