University of Bristol - Explore Bristol Research

General rights

Artificial Life and Robotics (2022) 27:264–271
https://doi.org/10.1007/s10015-022-00749-8

ORIGINAL ARTICLE

Stochastic behaviours for retrieval of storage items using simulated robot swarms

Emma Milner[1] · Mahesh Sooriyabandara[2] · Sabine Hauert[1]

Received: 27 August 2021 / Accepted: 8 January 2022 / Published online: 18 April 2022

Abstract
Robot swarms have the potential to be used as an out-of-the-box solution for storage and retrieval that is low cost, scalable to the needs of the task, and would require minimal set up and training for the users. Swarms are adaptable, robust and scalable with a relatively low computational cost which makes them appropriate for this purpose. This project simulated a robot swarm with simple sensors and stochastic movement, collecting boxes from storage to deliver them to the user. We show in simulation that stochastic strategies based on random walk and probabilistic sampling of local boxes could give rise to competitive solutions to retrieve boxes and deliver them unordered, or following a predetermined order, within a storage scenario. The performance of the task is drastically improved using an additional simple bias rule which uses compass measurements and does not reduce the minimalism of the control. It is shown that swarm technology could provide an out-of-the-box system for storage and retrieval using only information local to each robot and with distributed control.

Keywords Swarm · Collective retrieval · Stochastic · Automated storage · Retrieval systems

1 Introduction

A use case study investigating the use of robot swarms for storage and retrieval found that there was an unmet need for an automated system to perform storage tasks instead of manual inventory without extensive set-up costs, maintenance or infrastructure [1, 2]. The use cases considered ranged from museum archives, to charity shops, to food banks, most of which said they were understaffed and lacked the time to organise their storage spaces to be efficient. This lead to errors and waste of stock. These use cases highlighted the need for out-of-the-box solutions [1]. Large commercial companies, such as Amazon or Ocado, already use multi-agent systems for automated storage and retrieval [4]. These solutions improve productivity and increase space for inventory [3]. While these centralised multi-agent solutions work well in controlled settings, they require extensive set up with bespoke infrastructure and high installation costs [5]. Robot swarm control instead is distributed among the robots and does not require instructions from a central computer to perform tasks. Swarm robots use only local sensory information to drive their behaviour. The collective group behaviour then emerges from the interaction between all robots and their environment. Swarms have the potential to be robust to failures and capable of scaling up and down in numbers. Individual robots can be made relatively cheap, and without large upfront investment in infrastructure or time [6]. Users (such as shop workers) could put the swarm in their stock room and leave it to react to their new environment and task. To be useful, the technology should require minimal technological expertise and set up, including no information about the inventory of the warehouse or the layout. In this paper, the authors show in simulation that a swarm of robots with simple, stochastic behaviours may provide solutions for storage tasks. The controllers are random walkers which perform probabilistic sampling of boxes. The

This work was presented in part at the joint symposium with the 15th International Symposium on Distributed Autonomous Robotic Systems 2021 and the 4th International Symposium on Swarm Behavior and Bio-Inspired Robotics 2021(Online, June1-4, 2021).

✉ Emma Milner
emma.milner@bristol.ac.uk

Sabine Hauert
sabine.hauert@bristol.ac.uk

[1] Bristol Robotics Laboratory, University of Bristol, Bristol, UK

[2] Toshiba Bristol Research and Innovation Laboratory, Bristol, UK

random movement and sampling means that no positional information, such as an inventory map, is required.

2 Related work

Centralised multi-robot systems are used for storage and retrieval tasks to improve storage efficiency. In current deployments, they are often used to speed up the order picking process whereby an item it transferred from storage to a human who packages it. Amazon and Ocado are famous users of these automated warehouses [4]. In the original Amazon Robotics system (Kiva Systems), the central job manager recorded the state of each robot and inventory stations and coordinated actions [7]. Such solutions typically require expensive set up of dedicated infrastructure [5]. With centralised control, research questions often focus on robot task allocation, path planning, and navigation [8]. These centrally controlled solutions improve warehouse logistics by optimising routes taken for faster delivery times and can increase inventory storage space by up to 50% [3]. However, they have disadvantages due to their centralised control. Reliance on a central computer to control the whole warehouse means that it can be vulnerable to technical failures with important consequences. For example, online clothing company 'ASOS' reported a 68% drop in pretax profits for the 2019 financial year, compared to 2018, which was attributed in part to an IT glitch in their automated warehouse that caused a backlog of items to be stored [14]. For more information on existing warehouse solutions, see a recent review by Custodio and Machado [9]. Interestingly, another recent review looking at multi-agent solutions for automated order picking makes no mention of decentralised, distributed, or swarm approaches, even though they covered 74 papers in the area [10]. This shows that swarm solutions to warehouse automation have largely remained untapped. A few solutions do make use of decentralisation. Draganjac et al. [11] for example present an algorithm for decentralised path planning and motion coordination control for multi-mobile robot systems in warehouses. They demonstrate benefits in scalability, as each vehicle makes its own path plan and negotiates for priority with other robots. The allocation of tasks however is still centralised. Hao et al. [12] demonstrate in simulation a decentralised retrieval system called GridHub which uses modular, conveyor units to move requested items from the storage grid to pickers. The GridHub storage space can deliver items in a desired sequence with no deadlocking but requires dedicated infrastructure. Following another approach, swarm algorithms have been applied to schedule tasks (e.g. Particle Swarm Optimisation and Ant Colony Optimisation). The execution of these swarm algorithms is however still run on a centralised system [13]. For these centralised systems to work well, they typically store and record information about every part of the system to coordinate the robots [4]. When the system scales up in component parts, so does this computational demand. Commercial solutions are starting to emerge that make use of swarm technology. For example, Agilox's Intelligent Guided Vehicles (IGVs) claim to operate without an external control system, instead using distributed control for autonomous routing and task allocation, and exchanging information with other members of the swarm about the environment [17]. Although this work has not been published, it seems the system calculates individual routes and has a high rate of information exchange between robots. This makes them useful, but it also means that the on board computational cost is high and the machines themselves can be complex and expensive. Additionally, their implementation times are a minimum of 6–12 weeks [16].

Overall, the current literature lacks simple machines that utilise swarm intelligence to perform retrieval tasks in an out-of-the-box way, without the need for any set up. This project suggests a stochastic swarm for retrieval tasks that can maximise the benefits of minimal set up and information exchange as well as flexibility, robustness, and scalability by using simple robots with distributed control. Their resource requirements are at a minimum due to their having no need for information about their task or environment.

3 Methodology

3.1 Simulated experiment

To systematically investigate the performance of a stochastic swarm for storage and retrieval tasks, a rapid 2D physics-based simulation was designed in Python. It can be downloaded at https://bitbucket.org/hauertlab/workspace/snippets/bxqKXq. A robot swarm made up of homogeneous agents is simulated. Stored items, known as boxes, to be collected and delivered are simulated as circles of the same size as the robots (although in the images shown here, Fig. 5, boxes are represented as squares to distinguish them from robots). Each robot is approximated to a 2D circle (as seen from a bird's eye view) of radius 12.5 cm and has sensory range of radius 35.0 cm, measured from the centre of the robot, which is used for collision avoidance. They sense and pick up a box if they make contact with it, which is represented by the robot centre being within a 25.0 cm radius of the box centre. The robots can move in any direction and do so with speed equal to 100 cm/s (acceleration is not considered in this case). The update frequency is once every 0.02 s. Robot parameters and motion specifications are based on a new physical platform produced in our laboratory and yet unpublished.

The storage space is simulated as a bounded 500 cm x 500 cm square. This size was chosen to represent the storage area of a small shop. At the upper bound of the x axis is a rectangular area of 100 cm width and 500 cm length, designated as the delivery area (seen in Fig. 5 as the area to the right of the dotted line). This is where the boxes are deposited for the user.

The aim of the tasks was to minimise the total time taken to deliver all the boxes. The robots are not given any information about their position, the box positions or the location of the delivery area. They also cannot exchange information except to sense their vector relative to nearby objects within sensory range and assign an obstacle type (i.e. robot, box or wall). Two tasks were performed in simulations. In *Unordered retrieval*, the swarm must retrieve all the boxes from storage and bring them to the delivery area. This represents scenarios were boxes are retrieved in parallel because they are not dependent on each other. In the second task, *Ordered retrieval*, items must be retrieved in a given sequence of box IDs. Each of the boxes are given unique ID numbers that can be read by a robot that is within sensory range of the box. In this task box i is not requested until box $i - 1$ has been delivered. A real example of this task could be a luggage store where customers can request their own luggage be brought to them from the storage room, such as the use case outlined in [1]. The size of the swarms tested (N_r) and the number of boxes (N_b) to collect are in the range $10 \leq N_r, N_b \leq 50$. Each combination of parameters N_r and N_b is tested in each of the two tasks. Each unique parameter pair is tested 10 times and then the resulting performance averaged over the 10 trials. Different numbers of boxes to collect are tested because this can be taken as analogous with changes in demand, where demand can be represented by more or fewer boxes requested to be delivered. The boxes and the robots start at randomly chosen positions within the storage space but none of them begin in the delivery area. Each task was tested with purely random walk and then with an additional behaviour named *Biased Heading Behaviour (BHB)*. This is intended as a helpful behaviour that could improve the task performance without complicating the swarm behaviours and hardware. BHB uses an on board compass to influence the robot to move towards the delivery area when it has a box (in the Ordered task, this would only occur when the robot has the correct box for the given sequence). An additional heading vector with a magnitude of 1 and entirely positive x direction (following the convention in Fig. 5) is added to the heading, which influences the robot to move towards the delivery area. Once the box is delivered, BHB is no longer used and it returns to random motion. Note that the compass sensor could also be replaced by a means to locally measure a gradient (e.g. light or radio) guiding the robot to the delivery area.

3.2 Controllers

The equation of motion for a robot used every time step is given in Equation 1 where speed Sp = 100 cm/s and the update frequency is once every 0.02 s (time step, $t_s = 0.02$ s). $\overrightarrow{\mathrm{H_t}}$ is a column vector made up of the x and y components of the heading of the robot at time t ($\overrightarrow{\mathrm{H_t}} = [H_{t,x}, H_{t,y}]$, see Equation 6).

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

The robot heading $\overrightarrow{\mathrm{H_t}}$ is determined by a combination of calculated headings due to random walk ($\overrightarrow{\mathrm{H_{noise}}}$), collision avoidance ($\overrightarrow{\mathrm{H_{r,b}}}$, $\overrightarrow{\mathrm{H_w}}$) and the Biased Heading Behaviour ($\overrightarrow{\mathrm{H_{bias}}}$). Each time step, these headings are calculated as follows. Collision avoidance of robots and boxes is described by Equation 2 and 3. In Equation 2 the $\overrightarrow{\mathrm{H_{r,b}}}$ of a given robot is the sum of the vectors from that robot to all of the robots and boxes within its object detection sensory range (SR = 35 cm). If the robot is not carrying a box then it will not include boxes in this calculation ($|\overrightarrow{\mathrm{H_b}}| = 0$) because it does not intend to avoid any boxes detected, it intends to pick them up. In Equation 2, $[d_{r,x}, d_{r,y}]_{<SR}$ is x and y distances to each robot within sensory range of the robot and $[d_{b,x}, d_{b,y}]_{<SR}$ is to each box. The signs of the vectors are negatives to induce the robot to move away from these obstacles.

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

Collision avoidance of the walls is triggered if the robot is touching the wall (i.e. the distance to the wall is one robot radius length (r = 12.5 cm)) in which case $|\overrightarrow{\mathrm{H_{wall}}}| = 100$ in the direction perpendicular to the wall it has hit. In Equation 3, W, E, N, S represent the West (x = 0), East (x = 500), North (y = 500) and South (y = 0) walls, respectively, following the convention displayed in Fig. 5. These values are binary whereby (e.g.) if $W = 1$ then the robot is touching the West wall and if $S = 0$ then the robot is not touching the South wall.

$$\overrightarrow{\mathrm{H_{wall}}} = 100 * \begin{bmatrix} (W - E) \\ (S - N) \end{bmatrix} \tag{3}$$

BHB is included in the robot behaviour through $\overrightarrow{\mathrm{H_{bias}}}$ (Equation 4) which has an x component of 1 when the robot has a box and 0 otherwise. It always has a y component of 0 and this combination forms a locomotion effect towards the delivery area which is at the upper bound of the x axis, in the warehouse (convention given in Fig. 5).

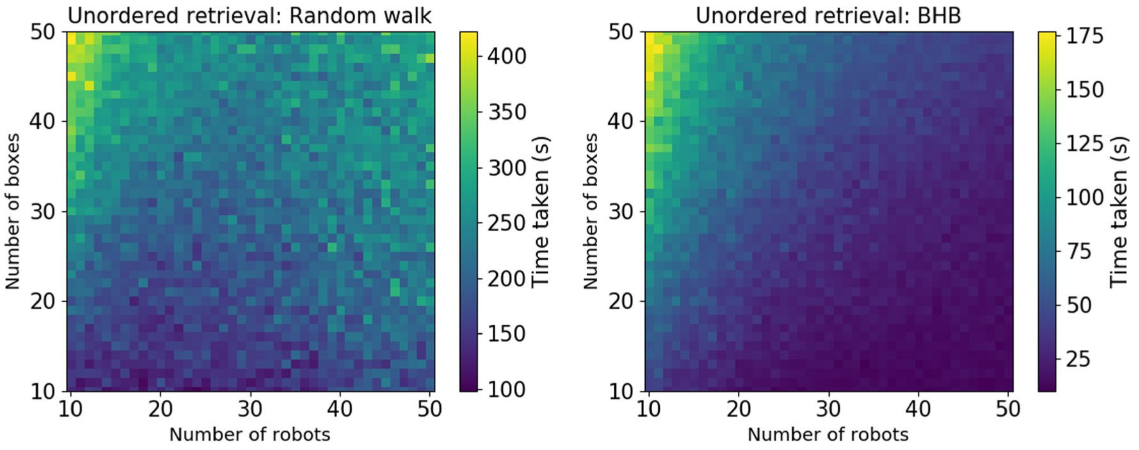

Fig. 1 Heatmap displaying the average time taken to complete the Unordered retrieval task with and without Biased Heading Behaviour (BHB). The results are for each combination of 10 to 50 robots and boxes. The random walk results range from 100-420 s. The BHB results range from 15-175 s

$$\overrightarrow{\mathrm{H}_{\mathrm{bias}}} = \begin{bmatrix}1\\0\end{bmatrix} otherwise \begin{bmatrix}0\\0\end{bmatrix} \qquad (4)$$

Finally, the random walk is implemented using $\overrightarrow{\mathrm{H}_{\mathrm{noise}}}$ (Equation 5) which selects a perturbation, p, of a random number to add to the current heading, in the range $-0.5rad \leq p \leq 0.5rad$. This random perturbation has the biggest influence on the heading of the robot when there are no nearby obstacles and the robot does not have a box.

$$\overrightarrow{\mathrm{H}_{\mathrm{noise}}} = \overrightarrow{\mathrm{H}_{\mathrm{t}}} + \begin{bmatrix}cos(p)\\sin(p)\end{bmatrix}, -0.5rad \leq p \leq 0.5rad \qquad (5)$$

These heading vectors are then summed together using vector addition to form a final robot heading, see Equation 6.

$$\overrightarrow{\mathrm{H}_{\mathrm{t+t_s}}} = \overrightarrow{\mathrm{H}_{\mathrm{r,b}}} + \overrightarrow{\mathrm{H}_{\mathrm{w}}} + \overrightarrow{\mathrm{H}_{\mathrm{noise}}} + \overrightarrow{\mathrm{H}_{\mathrm{bias}}} \qquad (6)$$

When the robot arrives in the delivery area, a simulated signal is sent to it to drop the box as it crosses the boundary line. The box is then instantly removed from the warehouse space. The picking up and dropping of a box is simulated as being instantaneous because the mechanical design of the lifting mechanisms are beyond the scope of this project at this stage, although our designed robot is capable of fast pickup and drop off. Box handling differs for the Ordered and Unordered retrieval tasks. For the Unordered retrieval task, the robots do not drop their box unless they are in the delivery area. In the Ordered retrieval task, it was necessary to change this control mechanism to avoid deadlocking which occurred when all the robots are carrying boxes but none of them have the correct box in the sequence. To address this, a probability based reshuffling of the boxes is included in the algorithm for the Ordered task. In this task, if a robot has found a box but it is not the next box ID in the sequence to be delivered then they pick it up (if they are free) and then each time step there is a probability, $P = 0.03$, that they will drop the box. The robots keep record of the last two box IDs that they have held and put down so that they do not get into a loop of continually putting down and picking up the same box.

4 Results

4.1 Unordered retrieval

A heatmap for the time taken to complete the Unordered task is given in Fig. 1. It displays the average time taken on a colour scale for every number of robots in combination with a number of boxes.

The best performances for the Biased Heading Behaviour (BHB) are seen at larger swarm sizes. This is true across the different numbers of boxes tested. For example, it took 50 robots 13.0 s to collect 10 boxes and 41.6 s to collect 50 boxes. This is a significant improvement compared to using only a random walk algorithm, with no BHB which took a swarm of 50 robots 184.6 s to collect 10 boxes and 307.0 s to collect 50 boxes. The worst performances for BHB are seen at low numbers of robots collecting high numbers of boxes, for example, it took 10 robots 173.4 s to collect 50 boxes. This is an increase of 131.8 s compared to a swarm of 50 robots. Performances were improved for all numbers of boxes by increasing the swarm size. The same pattern of improvement in performance by increasing swarm size is not seen in the random walk algorithm for the same parameters. The times are higher across the range of parameters tested for the random walk algorithm, compared to the BHB. For example, the range of times seen for random walk is roughly

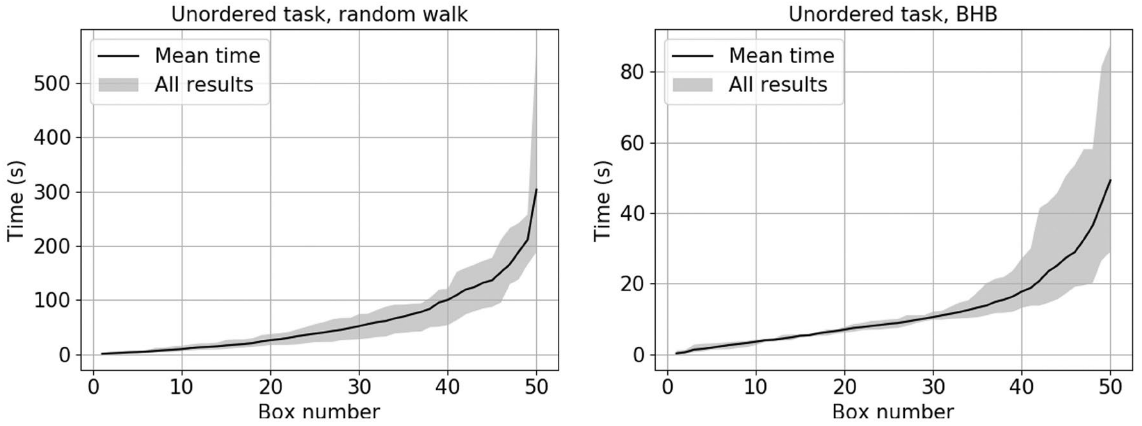

Fig. 2 Performance profile for the Unordered task using 50 robots. The mean time taken is given by the solid line and the transparent cloud around this line displays the full range of results over the course of the 10 trials. Results with random walk have a scale 10x higher than results with Biased Heading Behaviour (BHB)

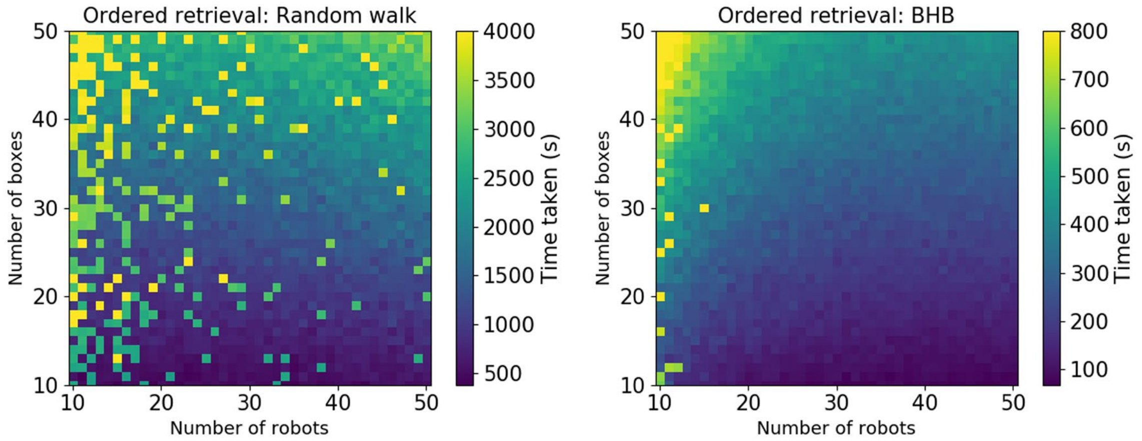

Fig. 3 Heatmap displaying the average time taken to complete the Ordered retrieval task. The results are for each combination of 10 to 50 robots and boxes. Random walk results are from 400-4000 s and any above this limit are set to 4000 s. The Biased Heading Behaviour (BHB) results are from 70-800 s and any above are set to 800 s

100-420 s compared to 15-170 s for BHB. This indicates that BHB improves the ability to collect boxes compared to a swarm using only random walk.

Figure 2 displays the delivery times for each of the 50 boxes collected in the case where the swarm size was 50 robots. These results show that, in the Unordered task, the time taken to collect boxes is not linear with the sequence number of boxes delivered. As the number of boxes left in the storage space decreases, the time taken between boxes collected increases exponentially. Most of the total time taken to complete the task occurs only for the last 10 boxes. For the random walker swarm, the mean time taken to collect the first 40 boxes (100 s) is less than the mean time to collect the last 10 (200 s). With BHB, the total time is lower but the effect seen is similar. The total mean time taken was 50 s, with the last 10 boxes taking longer (33 s) to be delivered than the first 40 (17 s). There is a smaller range of performance values over the 10 trials when using BHB than without, displayed as the cloud around the mean time line. This means that the times seen over the 10 trials were more similar and consistent between boxes delivered with BHB than without. The biggest variation in time results for the BHB was 58 s from the minimum to maximum times (seen at box 50). This is compared to a variation of 382 s for the same parameters, without BHB (with 92 s being the second largest variation, seen at 49 boxes and 400 s being the largest at 50 boxes).

4.2 Ordered retrieval

The results shown in Fig. 3 give the average time taken to complete the Ordered task. The times overall are much

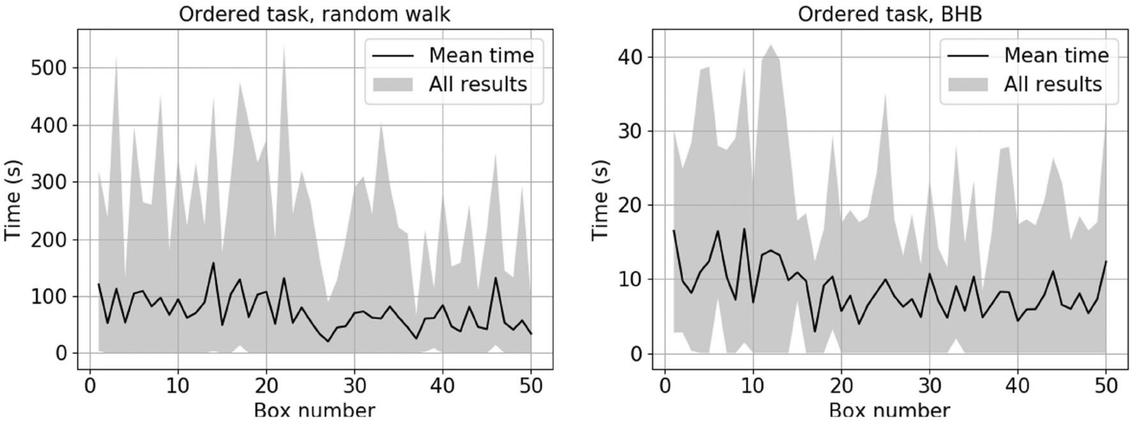

Fig. 4 Performance profile for the Ordered task with and without Biased Heading Behaviour (BHB). The mean time taken is given by the solid line and the transparent cloud displays the full range of results over 10 trials. Times without BHB have a scale 10x higher than results with BHB

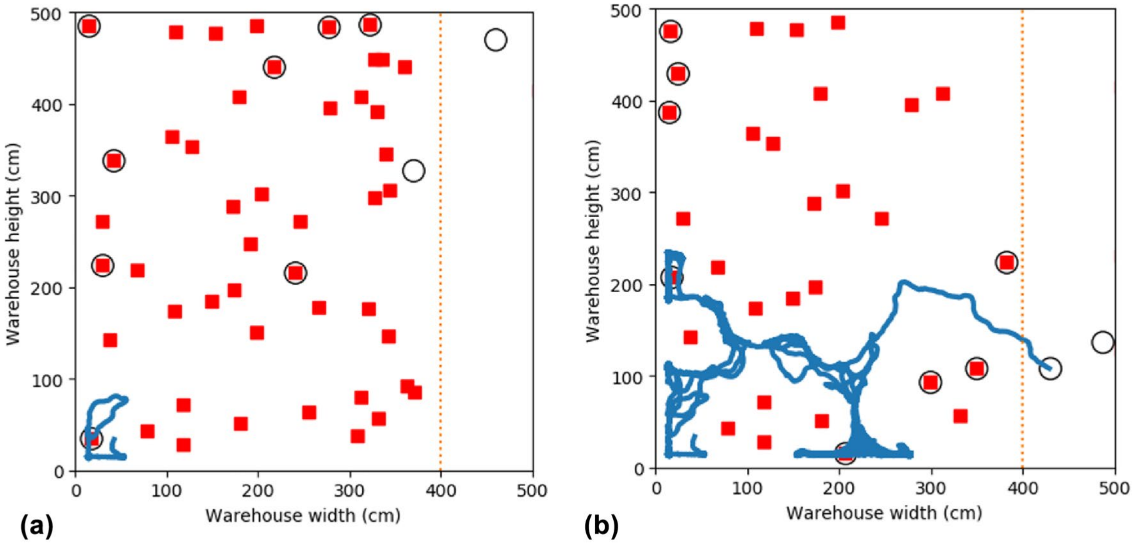

Fig. 5 Bird's eye view of the 2D simulation. A series of screenshots displaying the blocking behaviour seen at low robot numbers in combination with high box numbers. The robot trajectory over time is shown to be inhibited by static boxes in **a.** until they are cleared by the other robots by **b.** The squares represent boxes to be delivered and the circles are the robots. The 500 cm x 500 cm square represents the walls. The dotted line indicates the edge of the delivery area from $x =$ 400 to 500 cm

longer than the Unordered task as would be expected because the boxes cannot be delivered in parallel. The results that are above 4000 s (1.1 hr) for the random walk and above 800 s (13.3 min) for BHB are set equal to the limit (4000 s or 800 s respectively) in Fig. 3 for clarity of communication of the results. Beyond these times the experiment is considered failed, compared to the rest of the results seen, as it took too long to complete the task. It took BHB much less time than the random walk algorithm to complete the task in every case tested. The performance of the BHB algorithm greatly improved above 15 robots, for every box number, with no results above the given limit of 800 s seen beyond this number of robots. The lowest time taken for the BHB was 68.8 s ($N_r = 47$, $N_b = 10$). The lowest time taken for the random walk algorithm was 384.2 s ($N_r = 15$, $N_b = 10$). To collect the highest number of boxes ($N_b = 50$) it took the random walk algorithm a best time of 2614.0 s (43.5 min) using 13 robots, compared to 423.5 s (7.0 mins) using 50 robots using BHB.

The performance profile in Fig. 4 displays the time taken to deliver each box in a sequence of 50 boxes, by a swarm of 50 robots, in 10 trials. The time taken for each box that is delivered is measured from the time that it was requested by the user. Box i is not requested by the user until box $i - 1$ has been delivered. The range of mean delivery times, without BHB, is approximately 20–160 s. This means that the amount of time expected to receive a requested box is quite variable. When BHB is included, the range of mean times seen is much smaller, reduced to approximately 3–18 s. This decrease in variability shows that the BHB makes the swarm more consistent and reliable in their delivery time, making this approach a viable solution for storage applications. The largest time taken between subsequent boxes seen over the 10 trials is 42 s, compared to 525 s seen for random walk only.

5 Discussion

From observations of the simulated tasks it can be seen that higher numbers of robots find more boxes sooner. This contributes to their faster completion times compared to low numbers of robots. When a small number of robots are collecting many boxes (e.g. 10 robots collecting 50 boxes), the robots can be blocked in by surrounding boxes, and will then take a much longer time to clear them than when there are more robots. An example of this behaviour is shown in Fig. 5 where the robots are performing the Unordered task. The trajectory of a single robot with a box is shown, demonstrating how it gets trapped in a corner by boxes until they are cleared by other robots in the swarm.

Results from the performance profiles (Figs. 2 and 4) show that the majority of the time taken is used to collect the final few boxes. The reason for this is due to the random nature of the robot motion which means that a proportion of the box retrievals will take a long time compared to the average. This effect is lessened with BHB because the delivery trajectory is more direct, leading only the collection time to rely on random walk. The performance was best when the controllers included BHB, compared to pure random walk. This did not complicate the control and only requires simple compass data to function. The performance in an Amazon warehouse is estimated[1] to be 600 s (10 mins) to collect 50 boxes when they are all requested at the same time (Unordered retrieval). It is estimated [3] that Amazon would take 45,000 s (12.5 hrs)[2] to collect 50 boxes if the requests are given in a sequence (Ordered task). However the results can only be compared to Amazon warehouses and other automated warehouses with centralised systems qualitatively. This is because the differing distances travelled, speeds and inventory sizes. The aim of this qualitative analysis is to show how long a user would be willing to wait. The swarms proposed here are not designed to compete directly with such large warehouses. Instead, they are designed to fulfil an as yet unmet need for out-of-the-box retrieval systems. The use cases where this technology will be most useful are usually on a smaller scale, such as small retail, a food bank or a space station storage [2]. The time results for both tasks are reasonable and useful for these scenarios. Additionally, these use cases value low maintenance and re-usability as much as efficient times, both of which are advantages of the swarm system. Compared to similar centralised systems, the swarm shown here has greater usability out-of-the-box with minimal set up and no information required about the environment in order to retrieve items. The Biased Heading Behaviour in particular was a non-obvious, useful addition to the stochastic controller which reduced the variability and length of the times seen to collect all the boxes (Unordered task) and for each box collected in a sequence (Ordered task). Other simple clever navigation rules should also be explored to improve performance while keeping the control local to the robot. To this end, distributed situational awareness could further augment the local knowledge of the robot, by making best use of the sensory and computational power on board each robot [1]. Using a back of the envelope calculation it could be assumed that a perfect centralised system in our scenario would take 8 s to collect 50 boxes using 50 robots and 40 s using 10 robots. This is based on an Unordered task scenario where all robots start at the deposit area ($x = 400$ cm) and all boxes are against the opposite wall (the worst case initial positions). The robots would have to travel 400 cm to collect a box and 400 cm back to deliver them (centrally controlled, best paths planned), taking 8 s. If there are more boxes than robots then some would have to repeat the trip of 800 cm which would be an additional 8 s each time. This is a perfect centralised case which would use navigation to move in a direct line in x and does not consider collision avoidance effects. It took the BHB 41.6 s to collect 50 boxes with 50 robots which is a very similar performance to the ideal 9 s considering the robots use random walk when searching for boxes with only direct return journeys and include collision avoidance. In the case of 50 boxes using 10 robots, the BHB performance is 173.4 s which is still within the region of good performance compared to the perfect, centralised case of 40 s. Finally, to examine the performance of the swarm in the real world, further testing will be done on board a new physical robot swarm for logistics being finalised at our laboratory.

6 Conclusions

Automated storage and retrieval technologies use multi-agent systems to collect and deliver stock. This is often done on a large industrial scale but requires bespoke, expensive infrastructure and robust centralised control to work. Instead, there is a critical unmet need of smaller storage tasks that require out-of-the-box solutions with low setup time, and

[1] 300 items per hour according to [15]

[2] 15 mins to deliver 1 item in Amazon warehouses [3]

the ability to adapt and scale to the task at hand without any central control or infrastructure. In this paper we propose that swarm solutions offer these advantages. To demonstrate this, we show in simulation a swarm of robots in a 500 cm x 500 cm storage space. The robots aim to pick up and deliver boxes to a delivery area for the user, at their request. Two tasks were considered, the first was for all of the boxes to be delivered in parallel and the second was for the boxes to be delivered one after the other in a given sequence. The Unordered task was completed in a reasonable time with only the random walker algorithm whereas the Ordered task time was too long to be useful. However, competitive performance was achieved for both tasks using additional Biased Heading Behaviour. The BHB uses an on board compass to bias the robot heading in a given direction when it has a box. The rest of the robot behaviour was entirely stochastic and uncontrolled by the user or a centralised system. This should be investigated further as a cheap and easy solution for an out-of-the-box retrieval system using distributed controllers.

Acknowledgements Author one was supported by the EPSRC Centre for Doctoral Training in Future Autonomous and Robotic Systems (FARSCOPE) and an EPSRC ICASE Award. S.H was supported by the UKRI Trustworthy Autonomous Systems Node in Functionality (EP/V026518/1).

References

1. Jones Simon, Milner Emma, Sooriyabandara Mahesh, Hauert Sabine (2020) Distributed situational awareness in robot swarms. Adv Intell Syst 2:2000110
2. Daniel Carrillo-Zapata, Emma Milner, Julian Hird, Georgios Tzoumas, Vardanega Paul J, Mahesh Sooriyabandara, Giuliani Manuel FT, Alan (2020) Winfield and sabine hauert, mutual shaping in swarm robotics: user studies in fire and rescue, storage organization, and bridge inspection. Front Robot AI 7:53
3. Bogue Robert (2016) Growth in E-commerce boosts innovation in the warehouse robot market. Indus Robot Int J 43:583–587
4. Schranz Melanie, Martina Umlauft, Sende Micha, Elmenreich Wilfried (2020) Swarm robotic behaviours and current applications. Front Robot AI 7:36
5. Jan Roodbergen Kees, Vis Iris FA (2009) A survey of literature on automated storage and retrieval systems. Eur J Operation Res 194:343–362
6. Tan Ying, Zheng Zhong-Yang (2013) Research advance in swarm robotics. Def Technol 9:18–39
7. Peter R (2008) Wurman and Raffaello D’Andrea and Mick Mountz. Coordinating hundreds of cooperative autonomous vehicles in warehouses. AI Mag 29:9
8. Cai Kai (2020) Warehouse automation by logistic robotic networks: a cyber-physical control approach. Front Inform Technol Electron Eng 21:693–704
9. Custodio Larissa, Machado Ricardo (2020) Flexible automated warehouse: a literature review and an innovative framework. Int J Adv Manufactur Technol 106:533–558
10. Yasmeen Jaghbeer, Robin Hanson, Ingemar Johansson Mats (2020) Automated order picking systems and the links between design and performance: a systematic literature review. Int J Prod Res 58:4489–4505
11. Draganjac Ivica, Miklić Damjan, Kovačić Zdenko, Vasiljević Goran, Bogdan Stjepan (2016) Decentralized control of multi-AGV systems in aautonomous warehousing applications. IEEE Trans Autom Sci Eng 13:1433–1447
12. Hao Gang (2020) GridHub: a Grid-based, High-Density Material Handling System, Ph.D. Thesis, University of Louisville, Louisville, KY
13. Zhi Li, Vatankhah Barenji Ali, Jiazhi Jiang, Zhong Ray Y, Gangyan Xu (2020) A mechanism for scheduling multi robot intelligent warehouse system face with dynamic demand. J Intell Manuf 31:469–480
14. Wood Zoe, Kollewe Julia (2019) Asos Issues Second Profit Warning in Seven Months as Shares Fall, The Guardian. Accessed 24 August 2021
15. Bensinger Greg (2019)’MissionRacer’: How Amazon Turned the Tedium of Warehouse Work into a Game. The Washington Post. Accessed 24 August 2021
16. Steve Banker (2018) Robots with Swarm Intelligence. Logistics Viewpoints. https://logisticsviewpoints.com/2018/07/30/robots-with-swarm-intelligence/. Accessed 24 August 2021
17. AGILOX IGV - Intelligent Guided Vehicles (2021). AGILOX. https://www.agilox.net/en/agilox-igv#echte-schwarmintelligenz Accessed 24 August 2021

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.