



UNIVERSITY OF LEEDS

This is a repository copy of *Swarm Foraging under Communication and Vision Uncertainties*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/185204/>

Version: Accepted Version

Article:

Obute, SO, Kilby, P, Dogar, MR orcid.org/0000-0002-6896-5461 et al. (1 more author) (2022) Swarm Foraging under Communication and Vision Uncertainties. IEEE Transactions on Automation Science and Engineering, 19 (3). pp. 1446-1457. ISSN 1545-5955

<https://doi.org/10.1109/TASE.2022.3164044>

© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. Uploaded in accordance with the publisher's self-archiving policy.

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

Swarm Foraging under Communication and Vision Uncertainties

Simon O. Obute, Philip Kilby, Mehmet R. Dogar and Jordan H. Boyle

Abstract—Swarm foraging is a common test case application for multi-robot systems. In this paper RepAtt algorithm is used for improving coordination of a robot swarm by selectively broadcasting repulsion and attraction signals. This is a chemotaxis-inspired search behaviour where robots use the temporal gradients of these signals to navigate towards more advantageous areas. Hardware experiments were used to model and validate realistic, noisy sound communication and vision system. We then show through extensive simulation studies that RepAtt significantly improves swarm foraging time and robot efficiency under realistic communication and vision models.

Note to Practitioners

Abstract—This research developed a swarm foraging algorithm that takes into consideration the vision and communication sensing noise levels faced by robots in real world applications. The algorithm, known as RepAtt, was developed with the aim of emphasizing algorithmic simplicity and limiting the hardware requirements for the robots in the swarm. In this paper, we have focused on the problem of deploying swarm robots to forage litter in an environment such as a park. The communication model of the robots was based on the physics of sound, while their vision system was modelled using experiments with deep neural networks based object detectors. The results show that the RepAtt algorithm is robust to different distributions of targets (or litter) in the search space, exhibits good swarm efficiency with changes in swarm population and is robust to noise in its communication and vision systems. Apart from the RepAtt algorithm, other contributions made by this research include modelling of robot vision system to aid extensive study of the impact of communication and vision noise on swarm coordination. This will be relevant for extensive testing and validation before deployment to swarm robots hardware. The sound communication used in this research limits the kinds of environment the robots can be deployed in. Echoes within an enclosed environment and bandwidth limitation for communication frequency and public disturbance due to sound emitted by the robots can all contribute to this limitation. Thus, this research can be improved by investing in the development of a communication technology with similar physics. Other areas of improvement include adopting better obstacle avoidance algorithms and implementing suitable manipulators for handling litter objects. The algorithm can be extended to make it applicable for solving other problems such as search and rescue operations where foraging targets could be disaster survivors; demining and hazardous waste cleanup, where targets are the mines or waste material; and planetary exploration, where targets could be interesting features of the planets are the targets searched for by the robots.

Simon O. Obute was with the School of Computing, University of Leeds, UK. E-mail: elcymon@gmail.com

Philip Kilby is with CSIRO Data61 and the Australian National University, Australia. E-mail: philip.kilby@data61.csiro.au

Mehmet R. Dogar is with the School of Computing, University of Leeds, UK. E-mail: M.R.Dogar@leeds.ac.uk

Jordan H. Boyle is with the School of Mechanical Engineering, University of Leeds, UK. Corresponding author: J.H.Boyle@leeds.ac.uk

Manuscript received June 5, 2021

Index Terms—Primary Topics: swarm foraging, chemotaxis
Secondary Topic Keywords: bioinspired robotics, multi-robot systems

I. INTRODUCTION

Swarm robotics applies intelligent coordination behaviours observed in natural swarms to solve multi-robot problems [1]. Swarms in nature have the impressive ability of accomplishing complex tasks by following simple rules. For example, ants are able to forage food from locations that are beyond their individual sensory capabilities by following pheromone trails which other ants have laid. An individual agent in the group does not have access to global knowledge of the world and relies only on interaction with its immediate environment (and sometimes memory of previous experience) to make autonomous control decisions. The swarm paradigm presents a means of using decentralized control, local communication and sensing to allow multi-robot systems to automate tasks that are inefficient or impossible for single robots. The actions of individual agents collaborating with other swarm members produces emergent behaviours that solve tasks such as aggregation, clustering, exploration, navigation and foraging among others in robust, scalable and flexible ways [2].

Foraging is a canonical test case for swarm robotics which involves collective search and transport of objects to a specific deposit site known as the nest [3]. It has diverse potential real-world applications for automating farming processes, planetary exploration, hazardous waste clean-up or search and rescue [4]. It also integrates within a single agent robotic tasks such as vision, exploration, manipulation, communication and transport. This paper describes the Repulsion-Attraction (RepAtt) algorithm, which uses simple communication and a chemotaxis-inspired behaviour to improve coordination in a swarm of foraging robots. RepAtt was first proposed in our previous work [5]. This paper extends our previous work by using real-world vision experiments and deep neural networks for object detection as a basis for modeling a probabilistic robot vision system (more details in Section III-B). We use this vision model to study the impact of imperfect vision on the foraging performance of a robot swarm. To the best of our knowledge, this is a novel vision model.

Section II reviews communication and vision uncertainties in swarm foraging; the RepAtt algorithm, communication and vision models are discussed in Section III. In Section IV we present work on optimizing RepAtt parameters, demonstrate that the algorithm still works well with noisy communication signals, show that RepAtt scales well with swarm size and is robust to changes in target distribution. We also show that it is

able to work well despite imperfections in the robots' vision systems. Finally, concluding remarks and future directions are presented in Section V.

II. REVIEW OF COORDINATION FOR SWARM FORAGING

A key means of achieving cooperation among swarm members during foraging is through local sensing of targets and communication with other robots.

Vision-based sensing is the most common means used by swarm foraging robots to detect desired targets. In many simulations, robots detect objects based on their relative distance from the objects. These simulated robots generally have omnidirectional detection capability and they usually have 100% object detection accuracy within their target sensing range [6], [7], [8], [9], [10]. In hardware realisations of swarm robot algorithms, researchers have generally validated the collective behaviour of the robots in absence of realistic vision system. Thus, simplified targets such as QR codes [11], Bar codes [12], coloured shapes [1], [13], [14] and virtual reality [15] were used. However, the field of computer vision has significantly progressed due in part to advances in deep neural networks [16]. Deep neural network based object detection algorithms such as Regions with Convolutional Neural Networks (RCNN) [17], You Only Look Once (YOLO) [18] and Single Shot MultiBox Detector (SSD) [19] can be trained to detect multiple classes of complex objects. There has also been significant work done to reduce their computational requirements so the detectors can work well on constrained computing platforms such as Raspberry Pi and Arduino boards commonly used in swarm robotics. This led to the development of tiny-YOLO [20], MobileNet-SSD [21] and machine learning frameworks such as Tensorflow Lite for microcontrollers [22]. These advances in computer vision make it feasible to test swarm robots collective behaviour under realistic vision challenges they will face when deployed in real-world environments.

Communication, on the other hand, has mostly been realized through shared memory, the environment and direct communication [1]. In shared memory implementations, all robots have access to a shared medium to write and read information, which gives swarm robots a global means of communication. This could be global network access by all robots in the swarm [23], [24] or limited access, where only robots within proximity of the nest have access to the information [25], [26]. Major drawbacks of this approach are issues related to scalability, increased complexity of individual robots and inconsistency with the swarm paradigm of local sensing and communication. Achieving cooperation using the environment as a communication medium involves modification of the search space using "markers" or "beacons" to provide information that guides the search behaviour of foraging robots [1]. This approach is largely inspired by stigmergy based coordination mechanisms, such as pheromones observed in ant colonies. This was achieved through stationary robot beacons in [27], [8], RFID tags that stored pheromone information in [28] and an LCD screen platform that used variation in light intensity to communicate pheromone level [29]. A major challenge for this communication approach is finding

an effective and scalable means of "marking" the environment beyond controlled laboratory conditions. In direct communication, robots adapt their behaviour to improve foraging efficiency based on information exchanged with neighbouring robots. This was implemented in [8] using range sensors, while [6] and [30] exchanged packets of information containing robot state data. Direct communication faces design challenges regarding the type of information robots should exchange, handling interactions with multiple neighbours simultaneously, and robustness and reliability of the communication media. Although hardware implementations of swarm robot algorithms can reflect the impact noise has on swarm foraging, little research has been conducted on the quantification of the effect of noisy recruitment. Some works that considered the impact of noise realised it using imperfection in the location of targets communicated to other swarm members and/or uncertainty in the recruitment information sent to neighbouring robots [11], [13], [31].

The RepAtt algorithm discussed in this paper was first introduced in [5]. RepAtt is inspired by the chemotactic search behaviour observed in micro-organisms such as the *Escherichia coli* bacterium and *Caenorhabditis elegans* nematode. The novelty of RepAtt lies in the use of the foraging robots as sources of signals whose intensity degrades with distance, unlike other implementations that used immobile signal sources [32], [33]. Neighbouring robots then sense the change in intensity of these signals and use them to perform chemotactic search for good areas in which to forage. In addition to studying the impact of noisy communication on swarm foraging, we also modelled a realistic vision system for the robots and extensively investigated its impact on the performance of the swarm.

III. SWARM COORDINATION

A. Communication Model

RepAtt is based on the use of a communication mechanism whose intensity decreases smoothly with increasing distance from the source. The exponentially degrading signal of Equation 1 [34] was used, where A_{ij}^k is the strength of signal type k sensed by robot i , located d_{ij} metres away from signal source j . A_0 is the signal strength at the source, while α and A_e are the attenuation factor and mean ambient sound level - properties dependent on environment condition. Total signal strength sensed by a robot, $I_i^k(t)$, at any location in the world is the sum of same-type signals at that location (Equation 2), where n is the total number of robots and k is the signal type. We consider two signal types that robots can sense and broadcast: repulsion ($k = r$) and attraction ($k = a$) signals. To sense increase or decrease of attraction and repulsion signals, robots compute the difference in signal intensity between two time steps (Equation 3). It is important to note that RepAtt does not consider the nature of signal degradation (logarithmic, linear, exponential, inverse square law) or the size of signal's change. RepAtt uses only the sign of the change (that is,

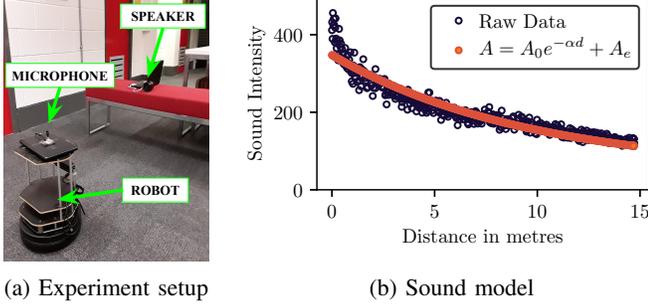


Fig. 1: Experiments for developing sound model.

whether it is positive or negative change).

$$A_{ij}^k = A_0 e^{-\alpha d_{ij}} + A_e \quad (1)$$

$$I_i^k(t) = \sum_{j=1, j \neq i}^n A_{ij}^k \quad (2)$$

$$\Delta I_i^k(t) = I_i^k(t) - I_i^k(t-1) \quad (3)$$

The parameters of Equation 1 were obtained through experiments using Turtlebot2 hardware platforms, speakers and omnidirectional microphones as described in our previous work [34] and shown in Fig. 1a. These parameters are: $A_0 = 299.18$, $\alpha = 0.12$ and $A_e = 48.18$ (see Fig. 1b for how modelled communication fits the raw experiment data). In addition, a sound experiment involving two sound sources was used to validate that multiple (white noise) sound sources always add up to give a higher amplitude. One source broadcast sound continuously for 130 seconds, while the second only broadcast sound for 20 seconds at 10 seconds intervals. The peak of the waveform occurred when the two sources were broadcasting at the same time, while the troughs occurred when only one sound source was active.

To test RepAtt’s robustness to noisy communication, the experiments also quantified noise in the sound signals, which was found to average 6% of signal intensity. This noise was then modelled as a normal distribution with mean of 0 and deviation of 0.06 as shown in Equation 4.

$$B_{ij}^k = A_{ij}^k (1 - \mathcal{N}(0, 0.06^2)) \quad (4)$$

An average filter was introduced to RepAtt to make it more robust to noisy communication. This simple filtering system involved each robot maintaining a limited queue size of attraction and repulsion signals. The robot then uses the average of the signals in its queue as its current signal intensity level and compares this value with a previously computed average to determine the change in signal intensity. An equivalent effect could be easily implemented in hardware through electronic low-pass filtering. The notation for this is $Nx-Qy$, which represent $x\%$ of the modeled noise and y time-step filter queue size. Thus N0-Q1, represents 0% noise and instantaneous signal measurements, while N100-Q40 represent 100% (of the experimentally-obtained value) noise level and queue size of 40 signal measurements. This modifies Equations 2 and 3 to Equations 5 and 6 respectively.



Fig. 2: Sample of images used as training dataset with litter objects annotated using red bounding boxes. The row are images downloaded from ImageNET [35] and the bottom row are sample images from robot’s view of litter objects.

$$I_i^k(t) = \frac{\sum_{b=t-y+1}^t \left(\sum_{j=1, j \neq i}^n B_{ij}^k(b) \right)}{y} \quad (5)$$

$$\Delta I_i^k(t) = I_i^k(t) - I_i^k(t-y) \quad (6)$$

B. Vision Model

The vision model proposed in this paper uses probabilities to model the uncertainties in a robot’s ability to detect targets within its visual field. This model uses real-world experiments of object detection with MobileNet-SSD, a state-of-the-art machine learning object detection network model, for the detection of litter in a moderately realistic environment such as a local park. This vision model will be used to extensively study the impact of imperfect vision on a swarm of foraging robots controlled using RepAtt and Random Walk algorithms.

A training dataset made up of 519 images downloaded from imageNET and 128 images of litter from a robot’s viewpoint were used. The litter in the dataset were then annotated by drawing a bounding box around each litter object in the images. The total litter annotations in the training data was 2609, a sample of which are shown in Fig. 2. This annotated dataset was then used to train the Mobilenet-SSD object detection model for 20,000 iterations.

The test dataset consists of 40 minutes of video recording of litter from a robot’s view. The camera was elevated at 44 cm from ground level and oriented at 35° facing downwards. The camera model used was a GoPro Hero 5 set to record at a full HD resolution of 1920 × 1080 pixels, wide angle field of view (of 118° horizontal and 69.5° vertical view angles), and 50 frames per second recording rate.

After training, the performance of the Mobilenet-SSD network was assessed on the test dataset. All the litter in the videos were tracked across all the frames that they were visible in. Only litter objects whose first appearance within the robot’s view start at the horizon and last visible location end at the bottom or lower sides of the video frame were used for modelling of the vision system. This helped to restrict the test data to litter objects that generated the most data points,

which were useful for a reliable modelling process. The total unique litter objects that meet this modelling criteria were 679.

The detection data while testing the MobileNet-SSD network showed that once a litter was detected by the network in the current frame, it was highly likely that it would be detected in the next frame. However, if the network failed to detect the litter in the present frame, it was unlikely for the network to detect that litter in the next frame. This was because the changes in visual scene between consecutive frames were generally small, considering that the videos were recorded at 50 frames per second and the robot was driven at a slow pace. Thus, the data showed that detections on consecutive frames were not statistically independent. From observing this detection pattern of the networks, a probabilistic vision model was developed to provide a representative approximation of the detection pattern. The model was based on the computation of two transition probabilities:

- 1) Probability that the object detector will detect the litter in the current frame when the object was unseen in the previous frame. This could be because the object in question had just entered the robot's field of view or it was classified as a false negative in the preceding time step. The probability controls the transition of an object from being unseen to it being seen and is represented as P_{u2s} .
- 2) Probability that a detection in the preceding time step is detected in the current time step. This probability controls how the detection of an object can persist in a seen state across multiple consecutive frames or time steps. This probability is represented as P_{s2s} .

These probabilities were computed for 124×124 and 220×220 network input resolutions to assess their relative performance. The steps followed to extract metrics data for developing the model probabilities are:

- 1) Tally all the seen to seen ($s2s$), seen to unseen ($s2u$), unseen to seen ($u2s$) and unseen to unseen ($u2u$) transitions for all the 679 litter objects. Also, tally the number of frames the litters were detected (s) and undetected (u) by the network.
- 2) From the tallied data, compute the transition probabilities for each litter using Equations 7 for P_{u2s} and 8 for P_{s2s} . Use the u and s data to compute the detection probability of each litter as shown in Equation 9.

$$P_{u2s} = \frac{u2s}{u2s + u2u} \quad (7)$$

$$P_{s2s} = \frac{s2s}{s2s + s2u} \quad (8)$$

$$P_s = \frac{s}{s + u} \quad (9)$$

- 3) Compute the overall mean and standard deviation of P_{u2s} , P_{s2s} and P_s from those computed for each of the 679 litter objects. The end result of this process is shown in Table I

Table I shows that MobileNet-SSD's average detection probabilities, P_s , are 0.4078 and 0.6104 for 124×124 and 220×220 input sizes when applied to the test dataset. For a single robot foraging alone, these detection probabilities

TABLE I: Analysis of the MobileNet-SSD metrics computation by analysing all frames of the test dataset containing the 679 filtered litter objects.

	124×124	220×220
seen	68801	103887
seen2seen	59380	91994
seen2unseen	9397	11840
unseen	103252	68166
unseen2seen	9394	11807
unseen2unseen	93203	55733
never seen	11	0
always seen	0	0
P_{s2s}	0.7896 ± 0.1824	0.8558 ± 0.1002
P_{u2s}	0.1293 ± 0.1031	0.2567 ± 0.1586
P_s	0.4078 ± 0.2304	0.6104 ± 0.2156

are low. However, within a swarm context, the combined contributions of multiple foraging agents can minimize the effect of poor robot vision. The effects of this imperfect robot vision system within a swarm context is investigated in detail in Section IV-H. This also reflects the swarm paradigm which relies on inter-robot cooperation among cheap, low-quality robot individuals to achieve a swarm goal.

From the probabilities computed in Table I, it can be seen that there is a significant difference between P_{s2s} and P_{u2s} probabilities. Both input resolutions have a low P_{u2s} , which means that MobileNet-SSD has a low chance of detecting previously unseen litter. However, when the litter has been detected by the network, there is a high likelihood for the network to detect it in consecutive frames that follow the detection, which is represented by the relatively high P_{s2s} probability. The high standard deviations on the computed probabilities indicate that there are differences in the detectability of litter objects. Factors such as lighting, distance, material and angle of approach of a robot to the litter object affect the detection model's ability to successfully detect the presence of the litter.

The P_{s2s} and P_{u2s} transition probabilities of the vision model play an important role in determining the overall target detection performance/probability, P_s , of the foraging robot. These probabilities also affect the detection behaviour of the robots such that as $P_{s2s} \rightarrow 1$, detected objects tend to remain visible to the robot; as $P_{s2s} \rightarrow 0$, previous detections get ignored by the robots; as $P_{u2s} \rightarrow 1$, undetected objects within the robot's frame have a high likelihood of being detected; and as $P_{u2s} \rightarrow 0$, the robot's ability to detect objects when they come within view becomes unlikely.

The variations of P_{s2s} and P_{u2s} transition probabilities and their effect on the overall detection probability P_s can be used to abstract the factors that affect the performance of a robot's vision system. Factors such as lighting condition, weather and distance in addition to object detection model, vision hardware and inference rate among others can have a significant impact on the performance of a robot's vision. The effects of these changes in transition probabilities on P_s were investigated

1.0	0.3304	0.8822	0.9897	0.9962	0.9976	0.9983	0.9990	0.9992	0.9997	0.9997	0.9999	1.0000	1.0000	1.0000
0.999	0.2630	0.8241	0.9814	0.9909	0.9952	0.9965	0.9975	0.9977	0.9984	0.9987	0.9987	0.9990	0.9989	0.9989
0.99	0.0807	0.4920	0.9012	0.9484	0.9651	0.9748	0.9792	0.9833	0.9854	0.9877	0.9888	0.9901	0.9901	0.9899
0.9	0.0097	0.0938	0.5013	0.6589	0.7486	0.8012	0.8350	0.8579	0.8765	0.8891	0.8999	0.9089	0.9079	0.9089
0.8	0.0044	0.0472	0.3336	0.4992	0.6027	0.6647	0.7165	0.7479	0.7774	0.7988	0.8186	0.8312	0.8350	0.8321
0.7	0.0033	0.0310	0.2501	0.4036	0.4969	0.5701	0.6288	0.6665	0.6995	0.7252	0.7513	0.7667	0.7691	0.7697
0.6	0.0025	0.0245	0.1995	0.3308	0.4272	0.5057	0.5522	0.6009	0.6391	0.6650	0.6931	0.7126	0.7156	0.7151
0.5	0.0026	0.0207	0.1678	0.2859	0.3715	0.4449	0.5027	0.5440	0.5823	0.6135	0.6447	0.6663	0.6652	0.6675
0.4	0.0017	0.0161	0.1421	0.2475	0.3322	0.4009	0.4556	0.5005	0.5372	0.5729	0.6008	0.6224	0.6254	0.6258
0.3	0.0015	0.0140	0.1250	0.2243	0.3020	0.3622	0.4160	0.4600	0.4999	0.5321	0.5636	0.5862	0.5877	0.5873
0.2	0.0012	0.0126	0.1108	0.1985	0.2740	0.3348	0.3848	0.4283	0.4667	0.5005	0.5302	0.5530	0.5558	0.5560
0.1	0.0011	0.0112	0.0975	0.1839	0.2521	0.3074	0.3574	0.4001	0.4363	0.4712	0.5004	0.5238	0.5258	0.5267
0.01	0.0011	0.0103	0.0906	0.1677	0.2331	0.2873	0.3361	0.3768	0.4153	0.4472	0.4758	0.5001	0.5022	0.5025
0.001	0.0010	0.0100	0.0921	0.1676	0.2308	0.2866	0.3340	0.3764	0.4132	0.4445	0.4746	0.4978	0.5003	0.5003
	0.001	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.99	0.999	1.0

Fig. 3: Monte Carlo simulation for 100,000 time steps showing the mean P_s for various combinations of $P_{u_{2s}}$ and $P_{s_{2s}}$.

using a Monte Carlo simulation. The transition probabilities were varied from 0.001 - 1.0 and applied for detecting 100 litter objects for a period of 100,000 time steps. Sections IV-G and IV-H investigate the effects of imperfect vision on robot communication and swarm foraging respectively.

The mean P_s resulting from the combinations of the modelling probabilities are shown in Fig. 3. The data shows that increasing $P_{s_{2s}}$ and $P_{u_{2s}}$ increases the P_s probability. When $P_{s_{2s}} = 1$, P_s started from 0.3304 ($P_{u_{2s}} = 0.001$) and sharply approached 1.0 (when $P_{u_{2s}} > 0.001$). This is a special case where once the object is detected by the agent, it will always be seen by the agent as long as it is within detection range. Thus, in this scenario, $P_s \rightarrow 1.0$ as time approaches infinity as long as $P_{u_{2s}} > 0$ (when $P_{u_{2s}} = 0$, the agent will never detect the object and P_s will always be 0). When $P_{s_{2s}} = 0.001$, P_s gradually increased from 0.001 to 0.5003 as $P_{u_{2s}}$ varied from 0.001 to 1.0. The value of $P_s \rightarrow 0.5$ in this scenario because as $P_{s_{2s}} \rightarrow 0$ the ability of the agent to detect the object for at least two consecutive frames reduces. The special case where $P_{s_{2s}} = 0$ and $P_{u_{2s}} = 1.0$ will lead to $P_s = 0.5$ because the agent is only able to detect the target once every two time steps. When $P_{u_{2s}} = 0.001$, P_s had minor increments until $P_{s_{2s}}$ became greater than 0.99, where a steep rise in P_s was observed. This huge increase in P_s is attributed to the near perfect ability of the agent to retain its previously detected targets with these parameters.

Two factors that can affect the value of the $P_{s_{2s}}$ and $P_{u_{2s}}$ transition probabilities are the quality of the object detector and number of frames the detector processes per second (fps). A high quality detector will be consistent in its detection ($P_{s_{2s}} \rightarrow 1.0$) and will quickly detect new litter that come within the robot's view ($P_{u_{2s}} \rightarrow 1.0$). A low quality detector will have the opposite effect on $P_{s_{2s}}$ and $P_{u_{2s}}$.

One the other hand, a high fps (such as the 50 fps used in the vision modelling experiments in Table I) promotes consistent detection pattern across two consecutive frames, resulting in high $P_{s_{2s}}$ and minimises chances of detecting objects missed

in preceding frame (that is low $P_{u_{2s}}$). A low fps will have the opposite effect on $P_{s_{2s}}$ and $P_{u_{2s}}$.

C. Repulsion-Attraction Algorithm (RepAtt)

The task for RepAtt is to improve coordination of swarm robots with limited carrying capacity searching for targets in an unknown environment and returning them to a central nest. Algorithm 1 is a pseudocode description of RepAtt. The coordination behaviour executed by a robot at each time step depends on whether the robot is in the searching, acquiring, homing or obstacle avoidance states, which are described in the subsequent paragraphs.

Obstacle Avoidance State (3 - 4) is used by robots to avoid static (nest and walls) and dynamic (other robots) obstacles when it bumps into them. It turns 45° to the left for obstacles on its right (or to the right for obstacles on its left) and random angle greater than 90° for obstacles in its front. It then makes a random linear motion between 0 and 1m before transitioning to either the searching, acquiring or homing states.

Homing State (5 - 6) is activated when the robot's capacity, cap , is full. In this state, the robot heads to the nest (it is assumed that the nest broadcasts a homing signal) and deposits the collected targets. The robot ignores attraction and repulsion signals from nearby robots until it has successfully offloaded all foraged targets at the nest.

The **Acquiring State** (22 - 23) is activated when a robot detects target(s) within its visual range ($found > 0$). The robot navigates to the nearest target to pick it up. During this process, it broadcasts the attraction signal if it detects more targets than its current carrying capacity, $found > cap$ (11 - 12). Thus searching robots within communication range can sense the attraction and appropriately adapt their search behaviour.

Searching State (24 - 27) is when a robot does not sense any target item to forage within its visual range ($found = 0$). The robot broadcasts a repulsion signal (9 - 10) to its neighbours while using random walk to search for targets.

Algorithm 1 Swarm Foraging Algorithm

```

1: Initialize Parameters: tumble probability  $P_b$ , robot capacity  $cap$ ,
   attraction multiplier  $a_m$ , attraction divisor  $a_d$ , repulsion multiplier
    $r_m$ , repulsion divisor  $r_d$ , tumble mean  $\mu$ , tumble deviation
    $\sigma$ 
2: while true do
3:   if obstacle encountered then
4:     Enter Obstacle Avoidance State
5:   else if  $cap == 0$  then
6:     Go home and drop collected targets
7:   else
8:      $P_t = P_b$ ,  $G_r = 1$ ,  $G_a = 1$ 
9:     if  $found == 0$  then
10:      Broadcast Repulsion  $A_i^r$ 
11:     else if  $found > cap$  then
12:      Broadcast Attraction  $A_i^a$ 
13:     if  $\Delta I_i^r > 0$  then
14:        $G_r = r_m$ 
15:     else if  $\Delta I_i^r < 0$  then
16:        $G_r = 1/r_d$ 
17:     if  $\Delta I_i^a > 0$  then
18:        $G_a = 1/a_d$ 
19:     else if  $\Delta I_i^a < 0$  then
20:        $G_a = a_m$ 
21:      $P_t = P_b \times G_r \times G_a$ 
22:     if  $found > 0$  then
23:       Go and pick up closest target
24:     else if  $\text{rand}(0,1) < P_t$  then
25:       make random turn of  $\mathcal{N}(\mu, \sigma^2)$ 
26:     else
27:       make straight motion

```

Its goal in this state is to minimize the repulsion (I^r) and maximize the attraction (I^a) signals it senses. This is achieved by detecting the change in intensity of these signals between two time steps (Equation 3 or 6). A robot increases its turning probability when moving in the wrong direction, i.e. when $\Delta I^r > 0$ or $\Delta I^a < 0$. Doing this increases a robot's likelihood of reorienting itself in the desired direction. On the other hand, when the robot senses a positive gradient for attraction ($\Delta I^a > 0$) or a negative repulsion gradient ($\Delta I^r < 0$), it reduces its turning probability, which in turn helps the robot to maintain its current direction for a longer period of time and consequently approach a region that increases its likelihood of finding a target. Lines 13 - 21 represent this turn probability adaptation, where $a_m \geq 1$, $a_d \geq 1$, $r_m \geq 1$ and $r_d \geq 1$ are predefined constants.

In Algorithm 1, the Random Walk algorithm (RW) used as a baseline in Section IV can be achieved by setting $a_m = 1$, $a_r = 1$, $r_m = 1$ and $r_d = 1$. This disables tumble probability adaptation by robots based on attraction and repulsion gradients, making them explore with constant probability of turning.

D. Adaptive Large Neighbourhood Search (ALNS)

The ALNS heuristic presented in [36] is a centralized, offline route computation algorithm that has been shown to be very effective in many transportation problems. We modelled the target foraging task of the swarm using ALNS to represent a centralized coordination approach to multi-robot foraging.

In the ALNS approach, the robots' foraging route is computed offline, using the nest as drop-off location for all robots

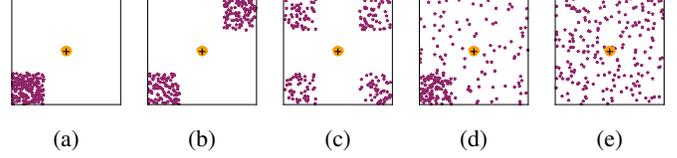


Fig. 4: (a) One50m, (b) Two50m, (c) Four50m, (d) Half50m, (e) Uniform50m. Plot of initial world states, for 50 m \times 50 m worlds. Targets are purple, black '+' is nest and yellow blob represent the robots. For 100 m \times 100 m worlds, target and robot locations were kept constant, while world width and length dimensions were doubled.

with full capacity. The exact setup described in [36] was implemented, where the simulated annealing route optimization was performed for 25,000 iterations, with a maximum of 50 or 100 visits removed in each iteration. The searching state of RepAtt is replaced with the offline simulated annealing optimization of the large neighbourhood search. Robots used the optimized ALNS routes as waypoints when foraging. This approach therefore gives a lower bound on the total foraging time. However, it is not scalable or robust to changes in target locations or swarm size.

The Random Walk and ALNS approaches are used to allow comparison of the RepAtt coordination mechanism's performance against two extremes: absence of coordination (Random Walk); and a near-optimal solution based on complex centralized coordination with perfect knowledge of the environment (ALNS).

IV. EXPERIMENTS AND RESULTS

A. Simulation Setup

The Gazebo Simulation platform was used to simulate robots under 5 target distributions, 2 world sizes, variable parameter settings and swarm sizes, under noiseless and noisy communication settings. A simulation time step of 25ms was used and each simulation was repeated 30 times. The number of targets used was 200 and the swarm task was to locate and pick up 90% of these targets in each world setup (sample setups are shown in Figure 4). Each robot in the swarm moved with velocity of 0.6 m/s and spent 5 seconds stationary to process each target it finds to simulate the target pick up process. Other algorithm parameters are: $P_b = 0.0025$ applied at every time step, robot targets capacity 5, target detection distance of 3 metres, $\mu = 180^\circ$ and $\sigma = 90^\circ$. The μ and σ values were chosen to mimic the approximate 180° turns observed in chemotactic behaviour of biological organisms such as *C. elegans*.

B. Chemotaxis Gains Optimization

The attraction and repulsion gains, a_m , a_d , r_m and r_d play significant roles in the performance of RepAtt because they affect the responsiveness of robots to changes in communicated signals. To investigate their effects and find the best combinations for a swarm of foraging robots, a_m and r_m were selected from 1, 2, 4, 6, 8, 10 while a_d and r_d were selected from 1, 10, 50, 100, 1000. This resulted in 900 different

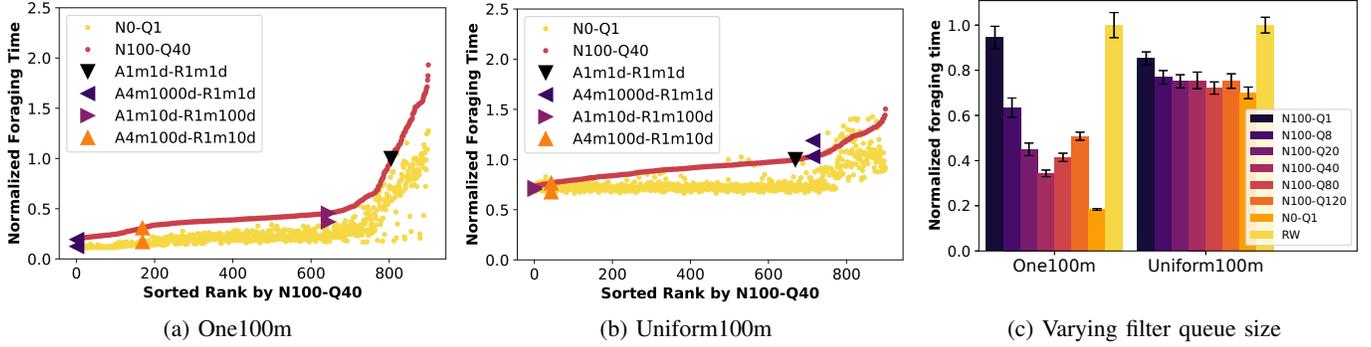


Fig. 5: Results represent mean of 30 independent repetitions for a swarm of 36 robots, y-axis represent time normalized based on performance of Random Walk. (a) and (b): $Aa_mma_{ad}-Rr_mmr_d$ match the respective gains in the legend. N100-Q40 foraging times were used to sort the x-axis and corresponding data for N0-Q1 have been included in the plots. (c): Variation of average filter queue size, RepAtt gains of $a_m = 4$, $a_d = 100$, $r_m = 1$ and $r_d = 10$ were used because they gave best foraging performance. Error bars represent 95% confidence interval.

combinations of these gains. Each gain combination was used by robots performing RepAtt in the 10 world setups, with each simulation experiment repeated 30 times under noiseless (N0-Q1) and noisy (N100-Q40) communication. Thus, 540,000 simulations were performed ($900 \times 30 \times 10 \times 2$) to search for best performing gain combinations. A specific combination is represented as $Aa_mma_{ad}-Rr_mmr_d$.

In each simulation, the task was for a swarm of 36 robots with carry capacity of 5 targets to pick up 180 targets in the world. The performance of each of the 900 parameter combinations was then sorted and assigned scores such that the combination with the shortest mean time had a score of 1 and longest mean time got a score of 900. Total score was computed by summing the scores across the 10 different world setups, with the best parameter combination attaining the lowest overall score (ultimately we used only the N100-Q40 results to select the best parameters, because this is the more realistic configuration). Sample results from the ranking are shown in Fig. 5a and 5b, where foraging times are normalized based on time taken by Random Walk (A1m1d-R1m1d). N100-Q40 data points were used to sort the rankings, and the corresponding performance for N0-Q1 has also been included in the plots. The results indicate that in clustered environments (for example One100m, Fig. 5a) increasing parameters that aid attraction toward targets (i.e. a_m and a_d) and minimizing repulsion parameters (i.e. r_m and r_d) produced better results. In addition, an a_m value of 4 performed better than 10 because of noise in the attraction signal - when a_m is too large, robots would make too many turns and explore only a limited area due to noise-induced inaccuracies in their gradient sensing.

In less clustered environments (for example Uniform100m), only r_d played a major role in swarm performance, where the best parameter combination was A1m10d-R1m100d. The results indicate that parameters that helped robots to make more tumbles when moving in the wrong direction (i.e. a_m and r_m) negatively impacted RepAtt, while parameters that aided swimming (a_d and r_d) positively affected RepAtt's performance.

Overall, the best parameter combination was

A4m100d-R1m10d, which is clearly an integration of the best parameters for clustered and uniform target distributions. In addition, the difference between best and worst performing combinations in One100m (0.30 vs 1.95) compared to Uniform100m (0.74 vs 1.48) indicates that communication has more significant impact in highly clustered environments in comparison to uniform environments.

C. Communication Noise Filtering

Moving from an idealised noiseless communication signal (N0-Q1) to the realistic noisy model (N100-Q1) in our simulated foraging task initially had an extremely detrimental effect, making RepAtt's performance only comparable to Random Walk (RW) as shown in Fig. 5c, where the optimized RepAtt gains of $a_m = 4$, $a_d = 100$, $r_m = 1$ and $r_d = 10$ were used. However, including the average filter with queue size of 8, 20, 40, 80, 120 improved RepAtt's performance. In addition, excessively large queue sizes (for example 80 or 120) decreased RepAtt's performance because robots lost too much information to make the gradient useful for its current location. A queue size of 40 gave the best performance across the 10 world setups in comparison to other queue sizes when working with noisy communication.

D. Foraging Performance Results

The simulation results for the 5 target distributions in $50 \text{ m} \times 50 \text{ m}$ and $100 \text{ m} \times 100 \text{ m}$ world sizes are shown in Figs. 6a and 6b respectively for a swarm size of 36 robots for Random Walk (RW), N100-Q40, N0-Q1 and ALNS algorithms. The optimized RepAtt gains of $a_m = 4$, $a_d = 100$, $r_m = 1$ and $r_d = 10$ were used.

In comparison with Random Walk, RepAtt improved swarm coordination and decreased the foraging time in all target distributions for both world sizes. In the $50 \text{ m} \times 50 \text{ m}$ world size, this improvement was 77% in the One50m world, which is more than half of the improvement offered by ALNS (90%). Similarly, the remaining four distributions recorded significant improvements in foraging time, with the weakest effect (33%

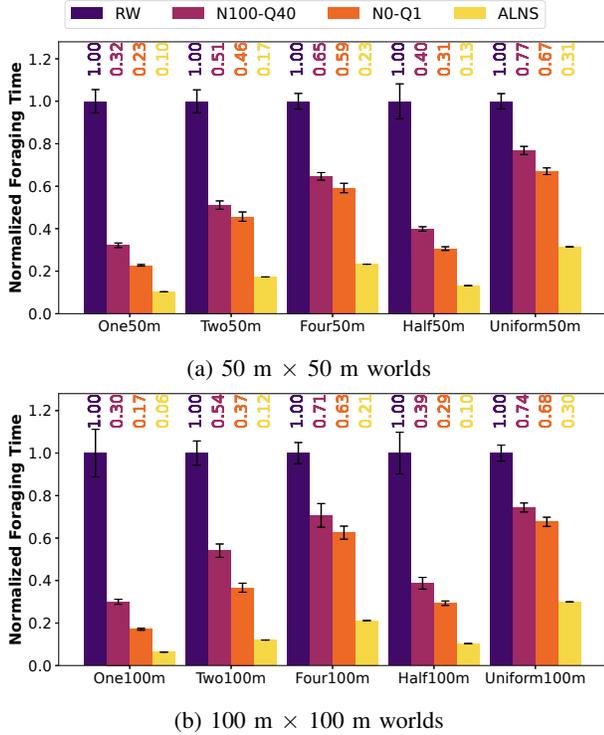


Fig. 6: Time taken in seconds to pick up 90% of targets for different world scenarios, normalised using the time taken by Random Walk. Each bar represents the mean of 30 simulation repetitions (also given numerically above each bar). The error bars represent 95% confidence interval. The optimized RepAtt gains of $a_m = 4$, $a_d = 10$, $r_m = 1$ and $r_d = 10$ were used for N0-Q1 and N100-Q40.

improvement) in the Uniform50m world. For the 100 m \times 100 m world size, where the search space was quadrupled, RepAtt also achieved excellent coordination to exploit target regions. Its improvements over Random Walk were 83%, 63%, 37%, 71% and 32% for the One100m, Two100m, Four100m, Half100m and Uniform100m distributions respectively. This is compared to ALNS’s values of 94%, 88%, 79%, 90% and 70% for the respective distributions.

It is logical that coordination would have a greater beneficial effect for highly clustered distributions. This is the reason for particularly large performance gaps between Random Walk and ALNS in the One, Two and Half cluster distributions and relatively smaller margins for the less clustered Four and Uniform worlds. It is also for these distributions that RepAtt gained the most improvements over Random Walk.

Comparing N100-Q40 and N0-Q1, noise reduced the effectiveness of RepAtt by 8% (in Uniform100m) to 43% (in One100m). Nonetheless, N100-Q40 performed well under the different target distributions with performance ranging between 30% to 77% of the time taken by the Random Walk algorithm compared to ALNS’s 6% to 31%.

These results indicate that this simple RepAtt algorithm is an effective mechanism for achieving swarm coordination when performing foraging tasks. They also show that the presence of noise, distribution of targets and size of the world

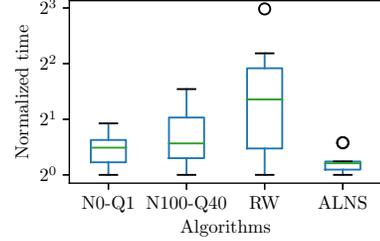


Fig. 7: Change in swarm foraging times across the ten target distributions. Each simulation was repeated 30 times, with y-axis showing mean time to pick up 90% of 200 targets (normalised based on distribution with shortest mean time for each algorithm, which is Uniform50m in all cases).

can have positive and negative impacts on the algorithm’s performance. The effectiveness of the algorithm is more pronounced when targets are clustered in smaller regions.

E. Robustness of RepAtt

One key advantage of autonomy of individual robots that make up the swarm is their robustness to changes in world setups and swarm size. This section focuses on the effects of changes in targets distribution on the swarm’s foraging ability (in terms of time taken to complete the foraging task), while Section IV-F covers the effects of swarm size on foraging efficiency. Fig. 7 shows the box plots of how swarm foraging time varied across the 10 world setups with RepAtt parameters of A4m100d-R1m10d, where foraging times were normalised by the shortest time taken for that communication model across all environments (which occurred in the Uniform50m).

Random Walk displayed the highest variability of 7.91, indicating that it is the least robust (or adaptable) to variation in target distributions and world sizes. Thus, the performance of Random Walk is highly dependent on the kind of problem, making it a more specialised solution that is not generally applicable to a wide variety of conditions. ALNS showed the least variability of 1.49, thus making it more generally applicable. However, ALNS requires *a priori* knowledge of the search space, which impacts its wider applicability. RepAtt displayed good performance across the different target distributions and did not result in any outlier when tested across the ten world setups. The variation in foraging time across the ten world setups for N0-Q1 and N100-Q40 were 1.90 and 2.91 respectively. N100-Q40 represents a 53% reduction in robustness in comparison to N0-Q1. However, when comparing N100-Q40 with Random Walk, the RepAtt algorithm improves the robustness of the swarm’s by 63.21%, thus, making the swarm more adaptable to changes in targets distribution.

F. Scalability of RepAtt

The scalability of RepAtt was evaluated by comparing the efficiency improvements as the swarm size varied from 1 to 100. Efficiency in this foraging task was computed as shown in Equation 10 where n is swarm size, tp is number of targets picked up, t_{tp} is time to pick up tp targets and E_r

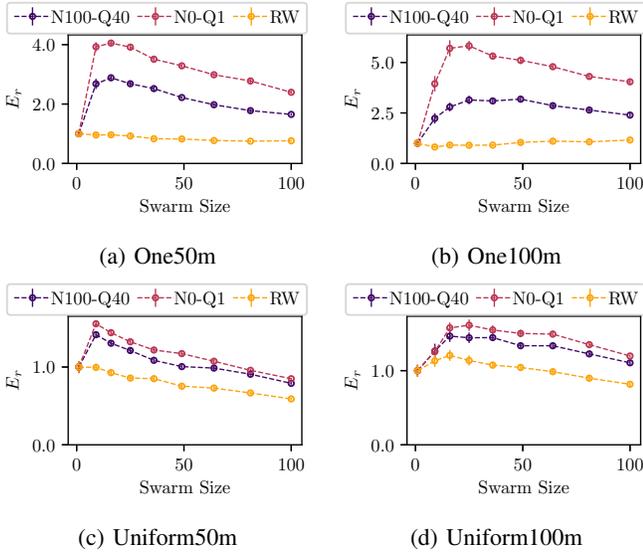


Fig. 8: Relative efficiency was computed based on $tp = 180$, where total targets were 200. Each simulation was repeated 30 times and error bars represent 95% confidence interval. The optimized RepAtt gains of A4m100d–R1m10d were used.

is relative efficiency (Equation 11). Thus, $n = 1$ represents a relative efficiency of 1, while $E_r > 1$ and $E_r < 1$ represent improvement and degradation in efficiency respectively.

$$E_n = \frac{tp}{n} \times \frac{1}{t_{tp}} \quad (10)$$

$$E_r = \frac{E_n}{E_1} \quad (11)$$

Fig. 8a and 8b show that RepAtt exhibited good scalability performance by improving relative efficiency by a factor of 5.82 when there was no communication noise (N0-Q1). With realistic noise, N100-Q40 always maintained an efficiency improvement of more than a factor of 2. However, Random Walk was at best able to maintain swarm efficiency as one would expect due to the lack of coordination. In Fig. 8c and 8d, the lack of coordination in Random Walk caused swarm efficiency to continuously degrade as swarm size increased, while RepAtt was able to maintain good efficiency improvement, especially for the Uniform100m world. In general, swarm efficiency is expected to drop as swarm size increases beyond some acceptable level. This is due to the effects of robot-to-robot interference, size of the search area and limited resources available for robots to forage.

G. Effects of P_{u2s} , P_{s2s} and vision update rate on RepAtt

The robots' vision update rate impacts the ability of a robot to forage and communicate effectively. The extent of this impact is different for varying combinations of the transition probabilities. This was studied in a simplified simulation setup, where a robot acting as the signal source should be broadcasting an attraction signal, but will repel instead if it fails to detect the target(s) within its detection proximity (the vision was controlled using P_{u2s} and P_{s2s} probabilities). The

listening robot senses the intensity and type (attraction or repulsion) of this signal and uses it to perform chemotaxis-based choice to move forward (toward the signal source) or reverse (away from the signal source). In addition, the rate at which the signal source applied the P_{u2s} and P_{s2s} probabilities was controlled by a detection frequency of 40 Hz, 4 Hz or 1 Hz to simulate varying inference times of the object detection vision system. The N0-Q1 version of RepAtt with $a_m = 4$, $a_d = 100$, $r_m = 1$ and $r_d = 10$ was used for this set of simulations each lasting 5,000 seconds. The result of these simulations are shown in Fig. 9a.

The data shows that the proportion of forward (correct) movements consistently increased with increasing P_s probability for vision update rates of 1 Hz and 4 Hz. For the 40 Hz update rate, the proportion of forward movements were the same as those observed for 1 Hz and 4 Hz rates when $P_s < 0.2$ and $P_s > 0.9$. However, for $0.2 < P_s < 0.9$, there was a noticeable distribution of the proportion of forward movements such that many of the P_{u2s} and P_{s2s} combinations for a specific P_s consistently gave lower proportion of forward movements.

In Figs. 9b and 9c, a heat map was used to colour-code the P_{s2s} and P_{u2s} probability combinations for the 40 Hz vision update rate. The data shows that for a specific P_s value, a high P_{s2s} probability combined with a low P_{u2s} value performed better than combinations where P_{s2s} was low and P_{u2s} was high.

The reason for this comes from the nature of the underlying N0-Q1 RepAtt algorithm used by the robot to locate the attracting robot. RepAtt relies on computing change in signal intensity between two time steps, which relies on consistency in the signal type and change in intensity. As $P_{s2s} \rightarrow 0$ and $P_{u2s} \rightarrow 1$ the consistency in the attraction signal type the robot senses reduces substantially. This is most obvious when $P_{s2s} = 0$ and $P_{u2s} = 1$. Although this results in $P_s = 0.5$, it would not be useful for a robot using instantaneous change in signal intensity to perform chemotaxis. This is because any correct gradient measurement is immediately offset by a wrong one in the next time step.

These issues do not arise for the lower frequencies of 1 Hz and 4 Hz because the robot was able to reliably compute signal gradient across multiple time steps. For example, at a vision update frequency of 1 Hz, the communicated signal is of a consistent type for 40 time steps. Thus, the robot performing RepAtt would compute accurate gradient values for this duration.

H. Foraging with Imperfect Vision

The combinations of the P_{s2s} and P_{u2s} model probabilities were also used to examine the foraging performance for a swarm of 36 robots foraging 200 targets in One50m, One100m, Uniform50m and Uniform100m environments (see Fig. 10). The vision update rate of the robots in each simulation was either 1 Hz, 4 Hz or 40 Hz in order to study the effects of object detection update rates on the swarm. In all simulations, the chemotaxis parameters of $a_m = 4$, $a_d = 100$, $r_m = 1$ and $r_d = 10$ were used. Simulations

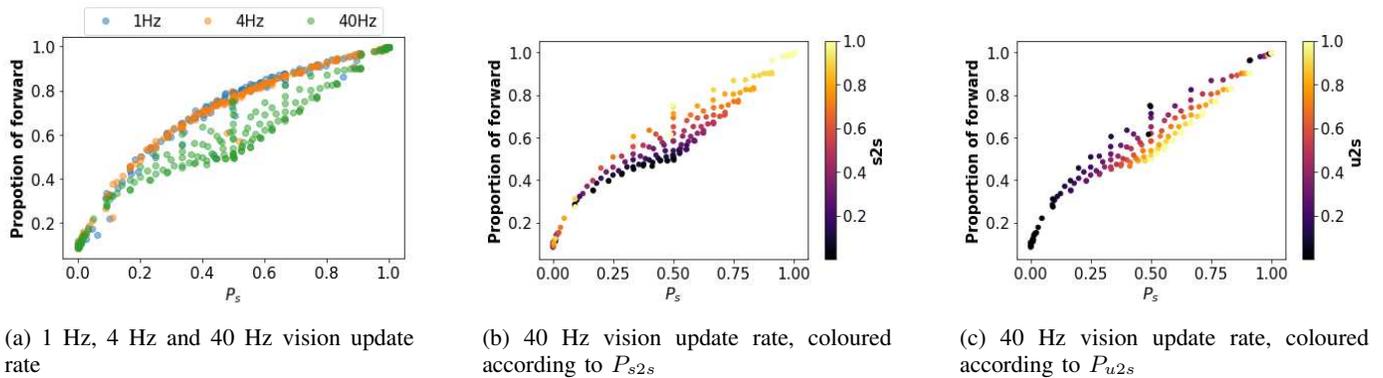


Fig. 9: The average proportion of forward movements by a one-dimensional robot as P_s increased. The P_{u2s} and P_{s2s} probabilities were applied at 1 Hz, 4 Hz or 40 Hz. In (a), darker colours represent multiple data points in same location. (b) and (c) use heatmaps of transition probabilities to distinguish between multiple data points.

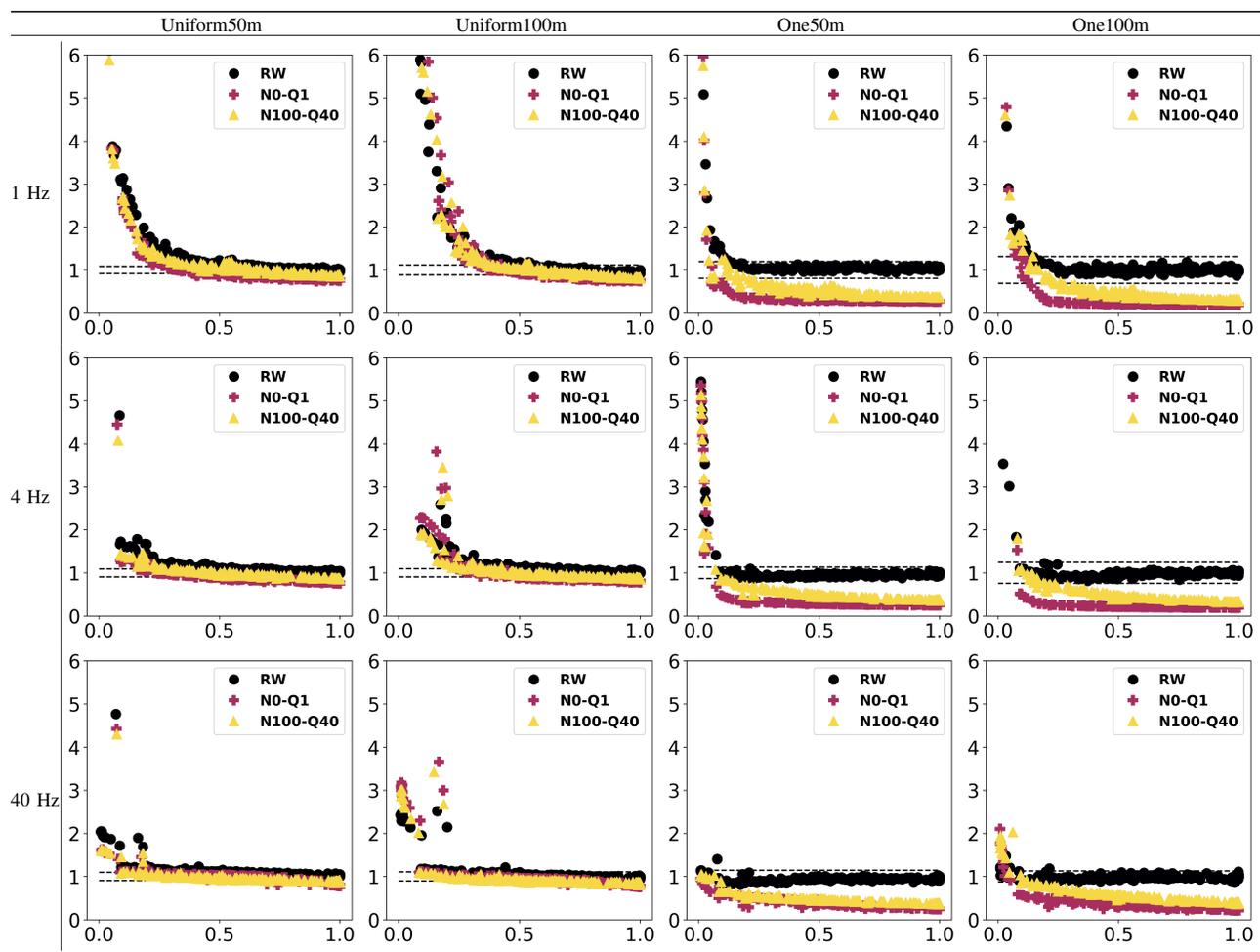


Fig. 10: Mean normalised time to pick up 90% of 200 targets by a swarm of 36 robots (y-axis) and P_s on the x-axis. Foraging times were normalised using the time taken by Random Walk with $P_{u2s} = P_{s2s} = 1.0$ for the corresponding world and vision update rate setups.

for each combination of model probabilities was repeated 30 times and the time it took for the swarm to pick up 180 targets was averaged across the repeated simulation runs.

For Random Walk, the impact of imperfect vision was more obvious in the uniform worlds and vision update rate of 1

Hz. However, at higher vision update rates (and in the single cluster worlds), imperfect vision had no noticeable effect until $P_s < 0.2$, which was the point at which search time increases due to poor target detection, resulting in a longer total foraging time.

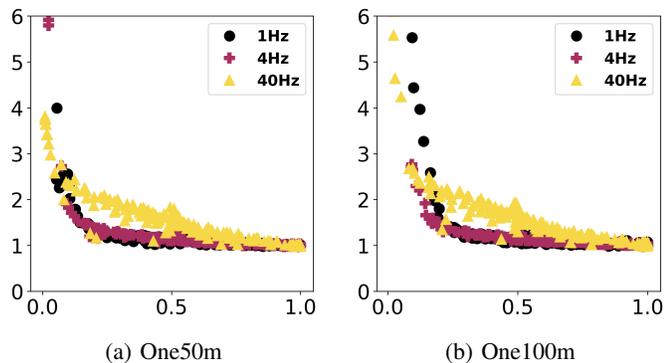


Fig. 11: The impact of vision update rate and model probabilities on N0-Q1. Foraging time is the y-axis, while x-axis represents P_s .

For the RepAtt algorithm, imperfect vision affects both the robots' ability to sense the presence of targets and choice of signal to communicate. A very low P_s means robots will broadcast repulsion signals when they ought to be attracting, which in turn negatively impacts the recruitment of other swarm members to clusters of target objects. This is why the impact of imperfect vision on swarm foraging is noticeable for slight reductions in P_s values, with stronger effects observed in the one cluster environments. RepAtt outperformed Random Walk for $P_s \geq 0.1$ in all the test environments and vision update rates, which indicates that the swarm was still able to exhibit some level of cooperation to help improve their foraging efficiency.

A closer look at the single cluster environments shows that increasing the vision update rate negatively impacted N0-Q1 foraging performance as shown in Fig. 11. However, this was not noticeable for Random Walk and N100-Q40 algorithms. The reason for this interesting behaviour was because the ability of searching robots to respond to attraction signals was significantly impacted negatively by the alternating attraction-repulsion behaviours of the robots that located targets cluster (as discussed in Section IV-G). Furthermore, this behaviour is most noticeable in environments that swarms could benefit most from communication.

V. CONCLUSION

This paper has presented the development of the RepAtt swarm foraging algorithm followed by the analysis of the impact of imperfections in communication and target detection on the collective behaviour of the robot swarm. Through realistic modelling of communication noise based on hardware experiments, the results presented in this paper revealed that, with a simple average noise filter, RepAtt was robust to the communication noise. This helped the swarm to forage in a robust, scalable and efficient way. Additionally, MobileNet-SSD - a state-of-the-art machine learning based object detection algorithm - was used as an experimental tool for developing a probabilistic vision model for swarm robots that conforms with recorded observations of real-world object detection data. The results indicated that RepAtt still exhibited superior swarm foraging performance in comparison to Random Walk at

detection probabilities as low as 0.2. This is good because it shows that RepAtt is able to adequately support swarm coordination in highly uncertain vision environments.

The simplicity of the RepAtt algorithm and its ability to be robust under realistic communication and vision noise makes it attractive for real-world implementations. The vision model used in this paper and, to a lesser extent, the communication model will serve as a useful tool for other researchers in the field with which to test and examine the effect of realistic noise models on the performance of their algorithms. This will provide beneficial and informative insight on the practicality of their proposed algorithmic solutions.

The results obtained from the chemotaxis-based algorithm strongly suggests there is great potential for further work in this area. Areas for future investigation include a survey of communication technologies for realising the RepAtt algorithm, implementation of the algorithm on swarm robots hardware platforms and studying the algorithm's performance within a three-dimensional search space.

ACKNOWLEDGMENT

The authors acknowledge the support of National Information Technology Development Agency, Nigeria and EPSRC self-repairing cities project. Simulations were undertaken on the High Performance Computing facilities at the University of Leeds, UK.

REFERENCES

- [1] L. Bayindir, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, pp. 292–321, 2016.
- [2] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.
- [3] O. Zedadra, N. Jouandeau, H. Seridi, and G. Fortino, "Multi-Agent Foraging: state-of-the-art and research challenges," *Complex Adaptive Systems Modeling*, vol. 5, no. 1, p. 3, 2017.
- [4] A. F. T. Winfield, "Towards an engineering science of robot foraging," *Distributed Autonomous Robotic Systems* 8, no. 3, pp. 185–192, 2009.
- [5] S. O. Obute, P. Kilby, M. R. Dogar, and J. H. Boyle, "Repatt: Achieving swarm coordination through chemotaxis," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 2020, pp. 1307–1312.
- [6] L. Pitonakova, R. Crowder, and S. Bullock, "Information flow principles for plasticity in foraging robot swarms," *Swarm Intelligence*, vol. 10, no. 1, pp. 33–63, mar 2016.
- [7] O. Zedadra, H. Seridi, N. Jouandeau, and G. Fortino, "A distributed foraging algorithm based on artificial potential field," in *12th International Symposium on Programming and Systems, ISPS 2015*. IEEE, apr 2015, pp. 201–206.
- [8] N. Hoff, R. Wood, and R. Nagpal, "Distributed colony-level algorithm switching for robot swarm foraging," in *Springer Tracts in Advanced Robotics*, vol. 83 STAR. Springer Berlin Heidelberg, 2012, pp. 417–430.
- [9] J. H. Lee and C. W. Ahn, "Improving energy efficiency in cooperative foraging swarm robots using behavioral model," in *Proceedings - 2011 6th International Conference on Bio-Inspired Computing: Theories and Applications, BIC-TA 2011*. IEEE, sep 2011, pp. 39–44.
- [10] T. Schmickl and K. Crailsheim, "Trophallaxis within a robotic swarm: Bio-inspired communication among robots in a swarm," *Autonomous Robots*, vol. 25, no. 1-2, pp. 171–188, aug 2008.
- [11] J. P. Hecker and M. E. Moses, "Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms," *Swarm Intelligence*, vol. 9, no. 1, pp. 43–70, feb 2015.
- [12] K. Russell, M. Schader, K. Andrea, and S. Luke, "Swarm robot foraging with wireless sensor motes," in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 1, Istanbul, 2015, pp. 287–295. [Online]. Available: www.ifaamas.org

- [13] E. Castello, T. Yamamoto, F. D. Libera, W. Liu, A. F. T. Winfield, Y. Nakamura, and H. Ishiguro, "Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach," *Swarm Intelligence*, vol. 10, no. 1, pp. 1–31, mar 2016.
- [14] R. Mayet, J. Roberz, T. Schmickl, and K. Crailsheim, "Antbots: A feasible visual emulation of pheromone trails for swarm robots," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6234 LNCS. Springer, Berlin, Heidelberg, 2010, pp. 84–94.
- [15] M. S. Talamali, T. Bose, M. Haire, X. Xu, J. A. Marshall, and A. Reina, "Sophisticated collective foraging with minimalist agents: a swarm robotics test," *Swarm Intelligence*, vol. 14, no. 1, pp. 25–56, mar 2020.
- [16] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," may 2019. [Online]. Available: <http://arxiv.org/abs/1905.05055>
- [17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, sep 2014, pp. 580–587.
- [18] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," apr 2018. [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS. Springer Verlag, 2016, pp. 21–37.
- [20] J. Ma, L. Chen, and Z. Gao, "Hardware implementation and optimization of tiny-YOLO network," in *Communications in Computer and Information Science*, vol. 815. Springer Verlag, 2018, pp. 224–234.
- [21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," apr 2017. [Online]. Available: <http://arxiv.org/abs/1704.04861>
- [22] P. Warden and D. Situnayake, *Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers*. O'Reilly Media, Inc., 2019.
- [23] R. Arkin, T. Balch, and E. Nitz, "Communication of behavioral state in multi-agent retrieval tasks," in *Proceedings IEEE International Conference on Robotics and Automation*. IEEE Comput. Soc. Press, 1993, pp. 588–594.
- [24] Yifan Cai and S. X. Yang, "A PSO-based approach to cooperative foraging tasks of multi-robots in completely unknown environments," in *2014 World Automation Congress (WAC)*. IEEE, aug 2014, pp. 813–822.
- [25] J. P. Hecker, J. C. Carmichael, and M. E. Moses, "Exploiting clusters for complete resource collection in biologically-inspired robot swarms," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, sep 2015, pp. 434–440.
- [26] Q. Lu, J. P. Hecker, and M. E. Moses, "Multiple-place swarm foraging with dynamic depots," *Autonomous Robots*, vol. 42, no. 4, pp. 909–926, 2018.
- [27] F. Ducatelle, G. A. Di Caro, C. Pinciroli, L. M. Gambardella, F. Ducatelle, G. Di Caro, L. Gambardella, and C. Pinciroli, "Self-organized cooperation between robotic swarms," *Swarm Intell*, vol. 5, pp. 73–96, 2011.
- [28] A. L. Alfeo, E. C. Ferrer, Y. L. Carrillo, A. Grignard, L. A. Pastor, D. T. Sleeper, M. G. Cimino, B. Lepri, G. Vaglini, K. Larson, M. Dorigo, and A. S. Pentland, "Urban swarms: A new approach for autonomous waste management," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May. IEEE, may 2019, pp. 4233–4240.
- [29] S. Na, M. Raoufi, A. E. Turgut, T. Krajník, and F. Arvin, "Extended Artificial Pheromone System for Swarm Robotic Applications," in *The 2019 Conference on Artificial Life*. Cambridge, MA: MIT Press, jul 2019, pp. 608–615.
- [30] I. F. Pérez, A. Boumaza, and F. Chappillet, "Learning Collaborative Foraging in a Swarm of Robots using Embodied Evolution," in *ECAL 2017 - 14th European Conference on Artificial Life*. Lyon: Inria, sep 2017. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01534242/>
- [31] A. Campo, Á. Gutiérrez, S. Nouyan, C. Pinciroli, V. Longchamp, S. Garnier, and M. Dorigo, "Artificial pheromone for path selection by a foraging swarm of robots," *Biological Cybernetics*, vol. 103, no. 5, pp. 339–352, nov 2010.
- [32] S. G. Nurzaman, Y. Matsumoto, Y. Nakamura, S. Koizumi, and H. Ishiguro, "Biologically inspired adaptive mobile robot search with and without gradient sensing," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, St. Louis, USA, 2009, pp. 142–147.
- [33] G. R. Andrade and J. H. Boyle, "A minimal biologically-inspired algorithm for robots foraging energy in uncertain environments," *Robotics and Autonomous Systems*, vol. 128, p. 103499, jun 2020.
- [34] S. O. Obute, M. R. Dogar, and J. H. Boyle, "Chemotaxis based virtual fence for swarm robots in unbounded environments," in *Biomimetic and Biohybrid Systems*. Springer International Publishing, 2019.
- [35] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2010, pp. 248–255.
- [36] S. Ropke and D. Pisinger, "An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows," *Transportation Science*, vol. 40, no. 4, pp. 455–472, nov 2006.



Simon O. Obute received his B.Eng. in Electrical Engineering in 2010, M.Sc. in Computational Intelligence in 2014. In 2020, he completed his Ph.D. in Computing at the University of Leeds, UK. His research focused on developing robust, flexible and scalable algorithms for swarm robots faced with real-world challenges.



Dr. Philip Kilby is a Principal Research Scientist at CSIRO's Data61. His research interests are in discrete optimisation problems, particularly vehicle routing problems and other problems in transportation. His expertise is in applying AI and OR methods - particularly Constraint Programming and meta-heuristics - to provide solutions for real-world problems, including fleet logistics, optimisation of public transport systems, optimisation of traffic systems, nurse rostering, and patrol boat scheduling to name just a few.

Mehmet R. Dogar is an Associate Professor at the School of Computing, University of Leeds, UK. His research focuses on robotic planning. He is a co-chair of the IEEE-RAS Technical Committee on Mobile Manipulation and an Associate Editor for the IEEE Robotics and Automation - Letters (RA-L). Before joining Leeds, he was a postdoctoral researcher at Massachusetts Institute of Technology. He received his PhD in 2013 from the Robotics Institute at Carnegie Mellon University.

Dr. Jordan H. Boyle did his BSc and MSc degrees in Electrical Engineering at the University of Cape Town, South Africa, before moving to Leeds to do a PhD in the School of Computing. His PhD and subsequent EPSRC fellowship were in the field of biological modelling and bio-inspired robotics. He is now Associate Professor in the School of Mechanical Engineering at Leeds and works primarily in the field of mobile robotics for infrastructure inspection, but maintains some collaboration with biologists working on a range of topics.