White Rose Research Online
eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Leeds.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/163130/

Version: Accepted Version

Proceedings Paper:

Obute, SO, Kilby, P, Dogar, MR et al. (2020) RepAtt: Achieving Swarm Coordination through Chemotaxis. In: 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE 2020). 16th International Conference on Automation Science and Engineering (CASE 2020), 20-21 Aug 2020, Online. Institute of Electrical and Electronics Engineers. ISBN: 978-1-7281-6905-7. ISSN: 2377-3766. EISSN: 2161-8089.

https://doi.org/10.1109/CASE48305.2020.9216993

© IEEE 2020. This is an author produced version of a paper published in 16th International Conference on Automation Science and Engineering (CASE 2020) 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.

UNIVERSITY OF LEEDS
University of Sheffield
UNIVERSITY of York

RepAtt: Achieving Swarm Coordination through Chemotaxis

Simon O. Obute[1], Philip Kilby[2], Mehmet R. Dogar[1] and Jordan H. Boyle[3]

***Abstract*— Swarm foraging is a common test case application for multi-robot systems. In this paper we present a novel algorithm for improving coordination of a robot swarm by selectively broadcasting repulsion and attraction signals. Robots use a chemotaxis-inspired search behaviour based on the temporal gradients of these signals in order to navigate towards more advantageous areas. Hardware experiments were used to model and validate realistic, noisy sound communication. We then show through extensive simulation studies that our chemotaxis-based coordination algorithm significantly improves swarm foraging time and robot efficiency.**

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 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. In this paper we introduce the Repulsion-Attraction (RepAtt) algorithm, which uses simple communication and a chemotaxis-inspired behaviour to improve coordination in a swarm of foraging robots. Our approach emphasizes extreme algorithmic simplicity and demonstrates the power of minimalist bio-inspired search algorithms. By foregoing complex communication systems, the algorithm lends itself to simple, low cost hardware implementations.

Section II reviews coordination in swarm foraging, while our RepAtt algorithm is introduced in Section III. In Section IV we present work on optimizing RepAtt parameters, demonstrate that the algorithm still works well with realistic, imperfect communication signals and show that RepAtt scales well with swarm size. 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 communication. This has mostly been realized through a shared memory, the environment or 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. In [5], robots that locate attractants (objects to forage) use the shared medium to notify all swarm members of the target's location. Ref. [6] used the global knowledge of percentage of targets found and environment covered by the swarm to adapt the foraging strategy. 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. A closely related approach is the use of a central nest as means of exchanging information among the swarm, where only robots within a limited range of the nest are able to communicate [7], [8].

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. This approach is largely inspired by stigmergy based coordination mechanisms, such as pheromones observed in ant colonies. The work in [9] used robots to form stationary beacon networks that broadcast hop counts of their proximity to the nest and target locations, thereby forming a gradient to be used by foraging robots to locate and transport targets to the nest. In [10], pre-deployed flying robots were used to form beacon networks that communicated target locations to ground robots. Swarm robots in [11] used pheromone information on RFID tags placed at crossroads to optimise

---

*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.

[1] Simon O. Obute and Mehmet R. Dogar are with the School of Computing, University of Leeds, UK. E-mail: `{scsoo,M.R.Dogar}@leeds.ac.uk`

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

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

their paths while performing waste management within a model city. Ref. [12] used an LCD screen platform, where light intensity was used to display pheromone level for robots to sense with a downward-facing camera. A major challenge for this communication approach is finding an effective and scalable means of "marking" the environment beyond controlled laboratory environments.

In direct communication, robots adapt their behaviour to improve foraging efficiency based on information exchanged with neighbouring robots. For example, in [9], robots used range sensors to control their distance from their neighbours while maintaining a foraging front around the nest. In [13], robots used light to communicate locations of found targets. In [14] and [15], robots communicated their foraging success to other swarm members to improve the swarm's performance. In [16], robots applied repulsive "force" to nearby agents when searching for targets and reduced this force when transporting targets. 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. Our approach shows that an analogue signal that degrades with distance and relies on the natural physics of the environment to handle multiple signals simultaneously can be used as a simple but effective means of direct communication among a swarm of foraging robots.

The biological foundation for our approach is the chemotaxis behaviour observed in micro-organisms such as the *Escherichia coli* bacterium [17] and *Caenorhabditis elegans* nematode [18], whose motions are characterized by near-linear runs ('swimming' mode) with occasional turns ('tumbling' mode) that randomise the organism's next run direction. The probability that an individual *E. coli* or *C. elegans* will perform a tumble at any given moment depends on the change in concentration of chemical attractant (or repellent) it senses during exploration of the environment. If conditions are improving (increase in attractant or decrease in repellent chemical) the organism suppresses tumbles in favour of swimming, whereas worsening conditions lead to increased probability of tumbling. The simple but elegant approach of responding to the change over time means that a non-directional analogue sensor is sufficient for the organisms to aggregate around regions of high attractant concentrations or disperse to low repellent areas. In our proposed swarm algorithm, the robots themselves selectively propagate repellent and attractant signals that degrade exponentially with increasing distance to provide a flexible coordination mechanism to improve foraging efficiency of swarm robots. Furthermore, the robots sense and broadcast these signals selectively depending on the state they are in, thereby creating a dynamic sensory landscape. These properties make our approach different from the biological foundations of our algorithm and its previous robotic implementations in [17] for localising sound source and in [19], where swarm robots foraged energy from light spots in their environment.

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 [20] 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, whether it is positive or negative change).

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

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

$$\Delta I_i^k(t) = I_i^k(t) - I_i^k(t-1) \tag{3}$$

The parameters of Equation 1 were obtained through experiments using Turtlebot2 hardware platforms, speakers and omnidirectional microphones as described in our previous work [20]. These parameters are: $A_0 = 299.18$, $\alpha = 0.12$ and $A_e = 48.18$. 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.

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

An average filter was introduced to RepAtt to make it 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 $x\%$ noise level 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.

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 rand(0,1) < $P_t$ then
25:            make random turn of $\mathcal{N}\left(\mu, \sigma^2\right)$
26:        else
27:            make straight motion
```

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

$$\Delta I_i^k(t) = I_i^k(t) - I_i^k(t-y) \tag{6}$$

B. Repulsion-Attraction Algorithm (RepAtt)

The task for RepAtt is to improve coordination of swarm robots with limited 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. 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 (more tumbles) 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 (longer swims), 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.

C. Adaptive Large Neighbourhood Search (ALNS)

The ALNS heuristic presented in [21] 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 with full capacity. The exact setup described in [21] 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.

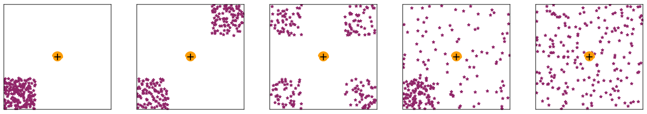

(a) (b) (c) (d) (e)

Fig. 1: (a) One50m, (b) Two50m, (c) Four50m, (d) Half50m, (e) Uniform50m. Plot of initial world states, for 50 m × 50 m worlds. Targets are purple, black '+' is nest and yellow blob represent the robots. For 100 m × 100 m worlds, target and robot locations were kept constant, while world width and length dimensions were doubled.

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 1). Each robot in the swarm moved with velocity of 0.605 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, $\mu = 180^0$ and $\sigma = 90^0$. 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 swarm 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 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_dd$-$Rr_mmr_dd.

In each simulation, the task was for a swarm of 36 robots with 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 score of 1 and longest mean time got 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. 2a and 2b, 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. 2a) 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 inaccurate gradient measure.

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.

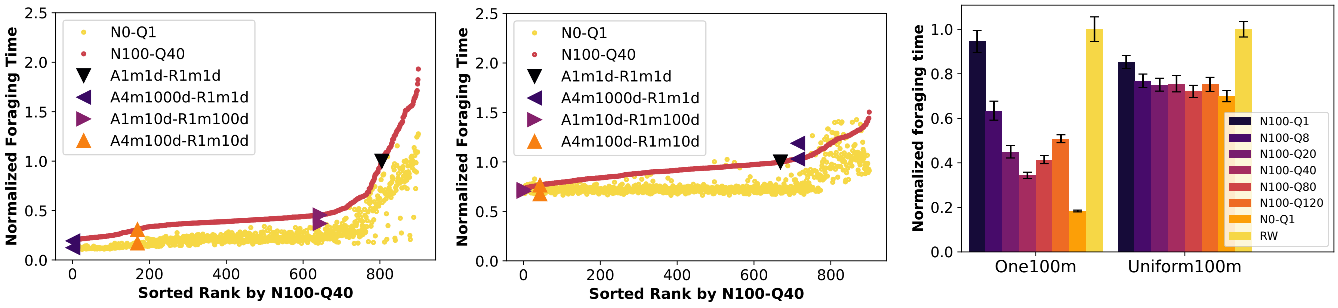
Overall, the best parameter combination was A4m100d$-$R1m10d, which is clearly an integration of the best parameters for clustered and uniform target distributions. In addition, the difference between best and worst performing combinations in One100m (0.30 vs 1.95) compared to Uniform100m (0.74 vs 1.48) 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. 2c, 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. Queue size of 40 gave best performance across the 10 world setups in comparison to other queue sizes when working with noisy communication.

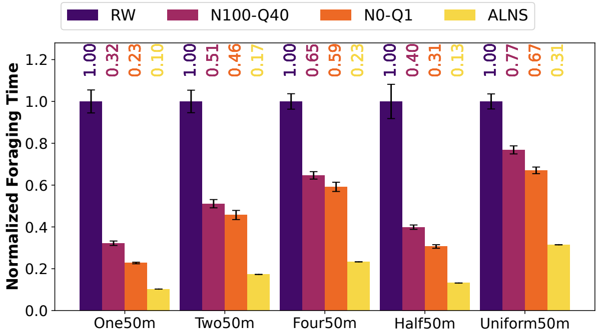
D. *Foraging Performance Results*

The simulation results for the 5 target distributions in 50 m$\times$50 m and 100 m × 100 m world sizes are shown

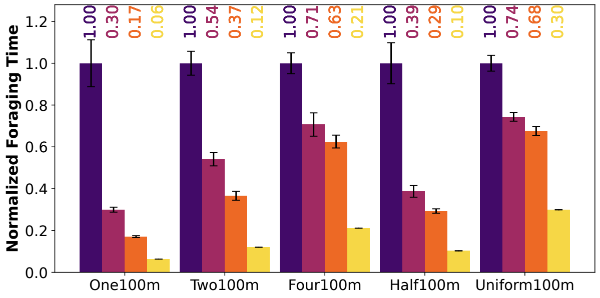


(a) One100m (b) Uniform100m (c) Varying filter queue size

Fig. 2: 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_dd−Rr_mmr_dd 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.



(a) 50 m × 50 m worlds



(b) 100 m × 100 m worlds

Fig. 3: 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.

in Figs. 3a and 3b 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 m × 50 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% improvement) in the Uniform50m world. For the 100 m × 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.

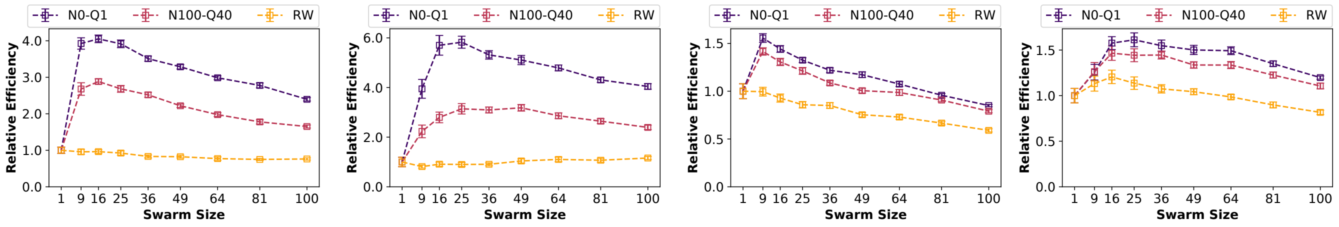
It is logical that coordination would have a greater beneficial effect for highly clustered distributions. This is the reason for large performance gaps between Random Walk and ALNS in the One, Two and Half cluster distributions and relatively smaller margins for the less clustered Four and Uniform worlds.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

One advantage of swarm robotics is that it presents improved efficiency in solving problems. We test the scalability of RepAtt by comparing the efficiency improvements as the number of robots varied from 1 to 100. Efficiency in this foraging task is computed as shown Equation 7 where n is swarm size, tp is number of targets picked up, t_{tp} is time to pick up tp targets and E_r is relative efficiency (Equation 8). Thus, $n = 1$ represents a relative efficiency of 1, while $E_r > 1$ and $E_r < 1$ represent improvement and degradation



(a) One50m (b) One100m (c) Uniform50m (d) Uniform100m

Fig. 4: 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.

in efficiency respectively.

$$E_n = \frac{tp}{n} \times \frac{1}{t_{tp}} \tag{7}$$

$$E_r = \frac{E_n}{E_1} \tag{8}$$

Fig. 4a and 4b 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 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. 4c and 4d, 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.

V. CONCLUSION

We have presented RepAtt, a simple yet effective swarm coordination algorithm. It significantly improves the efficiency of the underlying test case application of swarm foraging, even when using a realistic and far from ideal communication model that is grounded in physical experiments with un-optimised hardware. RepAtt is based on the concept of selective broadcasting of simple analogue repulsion and attraction signals among swarm agents, which they use to adapt their turning probability while searching for targets. The end result was a significant improvement in the swarm's coordination, which we measured by analysing the swarm's foraging time in 10 different world scenarios.

In future work, we will work on implementing better noise filtering for the swarm to improve communication as well as research alternative communication media that could be used in place of sound. We will also test our full algorithm using multiple physical robot platforms to further investigate and validate this coordination mechanism for the swarm within foraging and other applications.

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] R. Arkin, T. Balch, and E. Nitz, "Communication of behavorial state in multi-agent retrieval tasks," in *ICRA*, 1993, pp. 588–594.

[6] Yifan Cai and S. X. Yang, "A PSO-based approach to cooperative foraging tasks of multi-robots in completely unknown environments," in *World Automation Congress*, 2014, pp. 813–822.

[7] J. P. Hecker, J. C. Carmichael, and M. E. Moses, "Exploiting clusters for complete resource collection in biologically-inspired robot swarms," in *IROS*. IEEE, 2015, pp. 434–440.

[8] Q. Lu, J. P. Hecker, and M. E. Moses, "Multiple-place swarm foraging with dynamic depots," *Autonomous Robots*, pp. 909–926, 2018.

[9] N. Hoff, R. Wood, and R. Nagpal, "Distributed colony-level algorithm switching for robot swarm foraging," in *Springer Tracts in Advanced Robotics*, vol. 83, 2012, pp. 417–430.

[10] F. Ducatelle, G. A. Di Caro, C. Pinciroli, and L. M. Gambardella, "Self-organized cooperation between robotic swarms," *Swarm Intell*, vol. 5, pp. 73–96, 2011.

[11] A. L. Alfeo, E. C. Ferrer, Y. L. Carrillo, A. Grignard, *et al.*, "Urban swarms: A new approach for autonomous waste management," in *ICRA*. IEEE, 2019, pp. 4233–4240.

[12] 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*, 2019, pp. 608–615.

[13] P. E. Rybski, A. Larson, H. Veeraraghavan, M. Anderson, and M. Gini, "Performance Evaluation of a Multi-Robot Search & Retrieval System: Experiences with MinDART," *J Intell Robot Syst*, pp. 363–387, 2008.

[14] 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.

[15] I. F. Pérez, A. Boumaza, and F. Charpillet, "Learning Collaborative Foraging in a Swarm of Robots using Embodied Evolution," in *ECAL*. Lyon: Inria, 2017.

[16] R. C. Arkin, "Cooperation without Communication: Multi-Agent Schema-Based Robot Navigation," *Journal of Robotic Systems*, vol. 9, no. 3, pp. 351–364, 1992.

[17] S. G. Nurzaman, Y. Matsumoto, Y. Nakamura, S. Koizumi, and H. Ishiguro, "Biologically inspired adaptive mobile robot search with and without gradient sensing," in *IROS*, 2009, pp. 142–147.

[18] S. Ward, "Chemotaxis by the nematode Caenorhabditis elegans: identification of attractants and analysis of the response by use of mutants." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 70, no. 3, pp. 817–21, 1973.

[19] 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, 2020.

[20] 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.

[21] 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, 2006.