Optimal Swarm Strategy for Dynamic Target Search and Tracking

Hian Lee Kwa
Singapore University of Technology and Design & Thales Solutions Asia
Singapore
hianlee_kwa@mymail.sutd.edu.sg

Jabez Leong Kit
Singapore University of Technology and Design
Singapore
jabez_leong@mymail.sutd.edu.sg

Roland Bouffanais
Singapore University of Technology and Design
Singapore
bouffanais@sutd.edu.sg

ABSTRACT

Dynamic target search and tracking represents one of the most challenging problems for multi-agent systems. Effective strategies are critically needed to address numerous real-world robotic applications. Hitherto, the most common approach still relies on centrally controlled agents that become ineffective when tasked with both finding and tracking fast-moving targets in large and unstructured environments. While dynamic Particle Swarm Optimization (PSO) networks have been previously considered, the central effect played by the level of connectivity among swarming agents has been overlooked. In this paper, we present a fully decentralized swarming strategy offering a tunable exploration-exploitation multi-agent dynamics. This approach is achieved by combining adaptive inter-agent repulsion and an adjustable network PSO-based strategy. By tuning the topological distance between agents—i.e. the level of connectivity—we identify an optimal balance between exploration and exploitation leading to an effective performance of the swarm even in the presence of very fast moving targets. Beyond the quantitative results obtained through simulations, we present experimental test and validation of this approach with a fully decentralized swarm of eight ground miniature robots.

KEYWORDS

PSO-based Robotics; Multi-Robot Search; Dynamic Target Search; Swarm Intelligence

ACM Reference Format:
Hian Lee Kwa, Jabez Leong Kit, and Roland Bouffanais. 2020. Optimal Swarm Strategy for Dynamic Target Search and Tracking. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

1 INTRODUCTION

The search and tracking of a dynamic target is a challenging problem with applications such as search and rescue [13], as well as environmental monitoring [31]. Currently, the leading strategies for dynamic target search and tracking are based on centrally controlled agents using a predetermined search pattern to perform a systematic sweep of a given search area [23, 24]. However, these search methods are rendered ineffective when searching for and tracking a target moving faster than the capability of the individual searchers.

One proposed solution to deal with such highly dynamic target search and tracking problems is the usage of swarming multi-robot systems (MRS). These multi-agent systems have clear benefits over the use of single-robot systems or other non-decentralized or non-swarming MRS, including faster search times, the ability to withstand the sudden loss of agents—robustness, the ability to perform a task with differing number of agents—scalability, and the ability to perform a task in a dynamic environment—flexibility or swarm intelligence [5, 18].

Recently, the particle swarm optimization (PSO) algorithm, an algorithm suitable for stationary multidimensional and nonlinear functions [17], has been brought into the robotics realm to achieve effective target searches with swarming MRS [8, 10, 15, 29]. Modifications enabling this translation include, for instance, the addition of obstacle or inter-agent collision avoidance behaviors [9], as well as the implementation of a limited communications range [8, 25, 30]. The fact that PSO is a metaheuristic makes it a highly compelling approach for decentralized MRS operations in dynamic search-spaces.

However, two main challenges of implementing a PSO-based target search and tracking strategy in dynamic environments have not been sufficiently addressed. These challenges are: (1) solving the problems caused by outdated memory, and (2) excessive exploitation of the search environment at the expense of exploration [16]. In both cases, the swarm converges on the target but is unable to continue tracking the target's movements due to insufficient exploratory actions performed by the agents (i.e. particles) once engaged with the target. This highlights the requirement that the swarm needs to be able to constantly explore the search space, even after the target is found.

To promote sufficient exploration of the search space, swarm reinitialization procedures have been considered. This involves randomizing the particle positions and fully restarting the entire PSO process when the 'global best' (`gBest`) value has not been updated in several iterations [12, 20]. However, such procedures can only be considered when using PSO as a computational optimization technique. They are neither practical nor feasible in the physical world and as such, Jatmiko *et al.* have proposed dispersing the robotic swarm if the `gBest` value has not changed after a specified number of time-steps [15]. Another strategy to promote exploration of the search environment is to employ agents with a constant repulsive parameter, preventing them from clustering [3]. However, these approaches require an all-to-all network to facilitate inter-agent repulsion, thereby making them essentially centralized.

Another key factor to take into account when devising PSO-based search and tracking strategies is the connectivity of the agents involved when performing such tasks. From the computational standpoint of PSO, the most commonly studied network structures are the static `lBest` (ring) and `gBest` (all-to-all) network topologies, with the `gBest` topology being the predominant approach [19]. Typically, these networks are fixed during the initialization of the particles and do not change while the PSO simulation unfolds. It

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

has recently been recognized that agents in PSO networks with lower amounts of connectivity (i.e. low degree) perform more exploratory actions achieving higher global search performance. In contrast, agents in networks with greater connectivity (i.e. high degree) perform more exploitative actions and fare better at local search tasks [4]. It is worth pointing out that the network topology plays a central role in the effectiveness of distributed information exchanges underpinning the collective response of a swarm [26].

Although, the use of static networks has not been seen as a problem when considering PSO's applications *in silico*, using such methods *in robotico* with an MRS evolving in an unstructured environment poses significant challenges. This includes the limited communication ranges and limited information processing capabilities of a robot that constrains the number of connections it can effectively establish and sustain. As such, dynamic PSO networks based on metric-distance neighbors have been proposed and studied [25, 30]. It has been shown that larger communication ranges result in faster convergence times for the swarm [25]. This holds true until the communications radius of each member is able to cover the entire swarm, at which point, an all-to-all network is formed and there is no further reduction in convergence time [30].

Recently, it has also been shown that regardless of the technical limitations of the individual robotic units, there exists an optimal level of connectivity between swarming agents involved in a decentralized decision-making process. Interestingly, this optimal connectivity is directly related to speed of the driving signal [21]. Furthermore, it has also been shown that an excessive amount of interaction limits a swarm's ability to collectively respond to perturbations [22]. This implies that while tracking a dynamic target, instead of relying on all agents within a set communication range, tuning the level of connectivity of a swarm may result in a better collective tracking performance. While efforts have been made to implement dynamic network topologies in robotic swarms, this effect of varying the level of connectivity of a swarm operating in a dynamic environment has largely been ignored.

In this paper, a fully decentralized swarming strategy with tunable exploration-exploitation dynamics (EED) is introduced and quantitatively evaluated. The tunable EED allows the agents to prioritize exploitation or exploitation of the search space depending on information it collects from the environment, as well as information from its neighbors. This novel multi-agent dynamic is achieved by means of two fundamental elements: (1) an adaptive inter-agent repulsion behavior, and (2) an adjustable network PSO-based dynamics. By varying the topological distance between particles—i.e. the degree k of the k-nearest neighbor network—the balance between exploitative and exploratory behaviors changes. Through this process, we are able to uncover an optimal collective performance of the swarm corresponding to a specific connectivity k. This swarm strategy for dynamic search and tracking is experimentally tested and validated with a decentralized swarm of eight wheeled miniature robots operating in the absence of any supporting infrastructure.

2 METHODS

2.1 Search and Tracking Strategy

Our PSO-based swarming MRS with adjustable exploration and exploitation dynamics (EED) search and tracking strategy is based on: (i) a decentralized and dynamic network PSO-based algorithm searching for a global minimum value corresponding to the target, and (ii) an adaptive repulsion behavior promoting collective exploratory actions. The control of the inter-agent information exchanges is achieved by selecting the value of the degree k of the inter-connecting k-nearest neighbor network.

2.1.1 Decentralized Particle Swarm Optimization. Similar to the original PSO algorithm, the N agents (i.e. particles) that make up the swarm are initially spread out across the search-space. At any discrete time-step t, each agent i is fully characterized by three state variables: its two-dimensional velocity $\mathbf{v}_i[t]$, its position $\mathbf{x}_i[t]$, and its objective function value $f(\mathbf{x}_i[t], t)$. If agent i is on target at instant t, then $f(\mathbf{x}_i[t], t) = -1$. Otherwise, the its objective function value is zero. The explicit dependence on time of the objective function value $f(\mathbf{x}, t)$ reflects the dynamic character of the target.

At each time-step, each agent i seeks to bring its objective function value to the tracking value -1 by taking into account its current direction of travel and the best position of an agent within its neighborhood, denoted by $\mathbf{N}_{\text{best}}$ for "Neighborhood best." Most classical variants of PSO dealing with a static objective function also includes the influence of the personal best position—know as the cognitive term, which essentially acts as the agent's memory. Given that we consider the much more challenging case of a dynamic objective function, such memory effects are known to be completely counter-effective and are therefore discarded [8, 14]. The velocity and position of the agents are updated based on Eqs. (1) & (2). The parameter ω is known as the velocity inertial weight, and c as the velocity social weight, while r is a number randomly drawn from the unit interval.

$$\mathbf{v}_{\text{pso},i}[t+1] = \omega \mathbf{v}_i[t] + cr\big(\mathbf{N}_{\text{best}}[t+1] - \mathbf{x}_i[t+1]\big) \quad (1)$$

$$\mathbf{x}_i[t+1] = \mathbf{x}_i[t] + \mathbf{v}_i[t]. \quad (2)$$

It should also be noted that, $\mathbf{N}_{\text{best}}[t]$ is assigned the position of its neighbor with the best position at any particular time-step. This is unlike the original PSO algorithm where $\mathbf{N}_{\text{best}}[t]$ takes on the historical best position found by the particle's neighbors. It is important note that in our framework, the concept of a neighbor is understood in the network sense. This means that any agent i has as many neighbors as its degree k_i, and that these neighbors are changing given that we consider time-varying network topologies. The specifics are given in the following section.

2.1.2 Swarm Connectivity. The interaction network among swarming agents plays a key role in the effectiveness of our EED strategy. In particular, the level of connectivity—i.e. the degree—can be adjusted with significant effects on the collective dynamics [5]. As previously mentioned, higher levels of connectivity lead to higher levels of local exploitation, and consequently a higher degree of aggregation within the search space, while a low degree causes the swarm to prioritize exploration of the environment.

It has been shown that in flocks of starlings, birds interact with only 7 to 8 nearest neighbors on average, regardless of the distance between them [2]. Such a topological neighborhood concept is conveniently represented by a k-nearest neighbor graph. Here, the value of the degree k—identical for all agents—characterizes the level of connectivity within the swarm. At each time-step, any agent i identifies its k-nearest neighbors in the topological sense, and sets its $\mathbf{N}_{\text{best}}$ position to be the position of the nearest neighbor on target. Should the agent detect the target itself, or should none of the agent's neighbors detect the target, $\mathbf{N}_{\text{best}}$ is then set to the agent's own position $\mathbf{x}_i$ at that given time-step. At this point, we define $\mathcal{N}_i[t]$ as the set of k indices corresponding to the topological neighbors of agent i at time-step t. In the sequel, the reference to the time-step is dropped to simplify the notations.

2.1.3 Adaptive Repulsion. There are two main reasons for the implementation of inter-agent repulsion as part of our EED approach. Firstly, repulsion encourages a more exploratory behavior of the search-space and prevents the swarm from congregating within a small area—an undesirable herding dynamics. Secondly, from the practical swarm robotics standpoint, this behavior offers a welcome collision-avoidance measure.

The repulsive behavior adopted is based on the one used in another swarming system [28, 31]. For any agent i with topological neighbors j, the repulsion velocity can be expressed as:

$$\mathbf{v}_{\text{rep},i}[t] = -\sum_{j\in\mathcal{N}_i}\left(\frac{a_R[t]}{r_{ij}[t]}\right)^d \frac{\mathbf{r}_{ij}[t]}{r_{ij}[t]}, \tag{3}$$

where $\mathbf{r}_{ij}$ is the vector from agent i to agent j. This inter-agent repulsion is controlled by two parameters: the repulsion strength a_R that affects the agents' distance from each other at equilibrium, and the exponent d in the pre-factor term (a_R/r_{ij}). In what follows, d is fixed at 6 given that this value has very moderate effects on the performance of the EED strategy. At large (a_R/r_{ij}) and d values, the repulsion strength of the agents is approximately equal to the nearest-neighbor distance in equilibrium configuration [28]. As such, these two parameters were set so that the swarm was able to cover the entire search area when there was no target found.

A central aspect of this repulsive behavior is the ability of an agent to tune its level of repulsion based on the information it gathers from the environment and its neighbors. Based on this gathered local information, each agent has the ability to adjust the value of its own parameter a_R within a set range, thereby yielding adaptive repulsion. In our numerical tests, a_R was allowed to vary between 0.375 and 1.5. If an agent or its neighbors are not within range of the target, this agent enters an exploratory state, in which it gradually increases a_R until it reaches the maximum prescribed value. On the other hand, if an agent or at least one of its neighbors is within range of the target, the agent adopts a tracking behavior in which it gradually reduces the strength of its repulsion toward the minimum value. The adaptive repulsion is summarized in Algorithm 1.

2.1.4 Exploration Exploitation Dynamics. Given the elements reported in the previous sections, we can detail the inner workings of the proposed EED search and tracking strategy in Algorithm 2.

Algorithm 1 Adaptive Repulsion

```
Set a_{R,min} = 0.375, a_{R,max} = 1.5, d = 6, and δ = 0.01
while System active do
  if f(x_i[t], t) = −1 or ∃j ∈ N_i s.t. f(x_j[t], t) = −1 then
    if a_R > a_{R,min} then
      a_R ← a_R − δ
    end if
  else
    if a_R < a_{R,max} then
      a_R ← a_R + δ
    end if
  end if
  Calculate v_{rep,i} using Eq. (3)
end while
```

Algorithm 2 Dynamic k-Nearest Network PSO with Adaptive Repulsion

```
Set t = 0, k ∈ [2, N − 1], ω = 1, and c = 0.5
while System active do
  for All agents i ∈ [1, N] do
    Calculate f(x_i[t], t)
    if f(x_i[t], t) = −1 then
      N_best ← x_i[t]
    else
      Determine N_i = {j ∈ [1, N] s.t. agent j is a topological
      k-nearest neighbor of agent i}
      if ∃j ∈ N_i s.t. f(x_j[t], t) = −1 then
        N_best ← x_j[t]
      else
        N_best ← x_i[t]
      end if
    end if
    Calculate v_{pso,i} using Eq. (1)
    Calculate v_{rep,i} using Algorithm 1
    v_i[t] ← v_{pso,i}[t] + v_{rep,i}[t]
    v_i[t] ← v_i[t]/v_max
    x_i[t + 1] ← x_i[t] + v_i[t]
  end for
  t ← t + 1
end while
```

2.2 Swarm Performance Metrics

In this paper, three distinct metrics are used to characterize the performance of the swarm while searching and tracking the target. These are the cumulative velocity fluctuation magnitude, the correlation between an agent's heading and its bearing relative to the target's location, and the proportion of time the swarm has at least one agent in detection range of the target.

The choice of a particular metric is an important one as it conditions the effectiveness of our assessment of the performance of the system for a given task. Unfortunately, there is no general agreement on a particular metric in the literature. Most of the metrics reported hitherto tend to focus on the action of one or a few particular agents, and not the swarm as a whole. This hinders our ability to compare the respective performances of various approaches and

justifies our aim to consider new swarm performance metrics. The three metrics, which are introduced in the following sections, not only offer different insights into the overall complex collective dynamics of the swarm, but also reveal the influence of the exploitative and exploratory components on the agents' behavior.

2.2.1 Cumulative Velocity Fluctuation Magnitude. In the seminal work done by the STARFLAG group, Cavagna *et al.* found that in flocks of starlings, the effectiveness of the transmission of information between swarming agents is closely connected to the level of velocity fluctuations of individual agents and their correlations [6]. The same conclusion was reached when studying swarms of midges [1]. Therefore, as a proxy for its collective activity, the level of velocity fluctuations in our artificial swarm is quantified.

The velocity fluctuation of an agent i is given by the difference between its velocity and the mean velocity of the entire swarm:

$$\mathbf{u}_i[t] = \mathbf{v}_i[t] - \langle \mathbf{v}_j[t] \rangle_{j=1,\dots,N} = \mathbf{v}_i[t] - \frac{1}{N}\sum_{j=1}^{N} \mathbf{v}_j[t]. \quad (4)$$

The overall response of the swarm to the dynamic environment can then be calculated by taking the time-averaged sum of all of the swarming agents' velocity fluctuations, divided by the number of agents in the swarm, and then normalizing the value with respect to the agents' maximum speed:

$$\Xi = \frac{1}{T_f}\sum_{t=1}^{T_f} \left\langle \frac{\mathbf{u}_j[t]}{v_{\max}} \right\rangle_{j=1,\dots,N} = \frac{1}{NT_f v_{\max}} \sum_{t=1}^{T_f}\sum_{j=1}^{N} \mathbf{u}_j[t], \quad (5)$$

where T_f is the total number of time-steps considered.

2.2.2 Heading-Bearing Correlation. To quantify the exploitative activity of the swarm, we introduce the correlation between the heading of any agent i and the target's bearing relative to this agent. This correlation can be calculated as follows

$$\phi_i[t] = \frac{\mathbf{v}_i[t] \cdot \mathbf{t}_i[t]}{\|\mathbf{v}_i[t]\|} = \cos\theta_i[t], \quad (6)$$

where $\mathbf{t}_i[t]$ is the bearing of the target in relation to the agent and $\theta_i[t]$ is the angle between the two unit vectors (see Fig 1). The quantity ϕ_i therefore has a value between -1 and 1, with large negative values related to exploratory actions, and large positive ones revealing exploitative actions of agents that are indeed tracking the target.

2.2.3 Time on Target. The overall tracking performance of the swarm is classically determined by counting the number of time-steps when there is at least one agent within detection range of the target. This enables us to calculate the percentage of time the system is engaged in tracking the target as it moves around the search-space.

2.3 Target Representation

Most of the research on collective search and tracking considers the tracking of time-varying fields with given spatial variations. Indeed, with a continuous field a classical gradient-descent approach can be used to relatively easily locate the position of the target. Here, we represent the target by means of a binary objective function, with an arbitrary value -1 if an agent is on target and 0 otherwise. This conservative approach is meant to emulate one of the most challenging cases with a near-zero-range sensor tracking a target faster than the agent themselves. We believe that the design of effective swarm strategies benefits from considering such challenging scenarios.

2.4 Swarm Robotic System

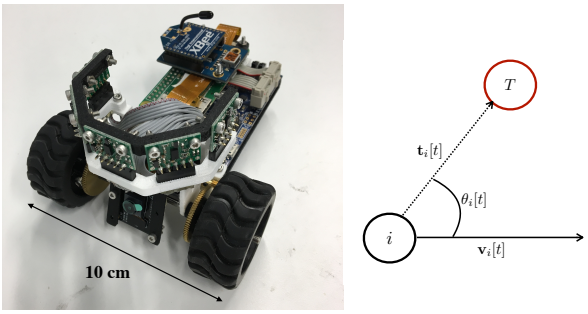

Figure 1: (Left) Robotic platform used for the experiments. (Right) Bearing of the target ($T[t]$) relative to the agent ($\mathbf{t}_i[t]$) and the heading of the agent ($\mathbf{v}_i[t]$) at time t.

For the experimental validations, 8 in-house developed differential-drive land robots were used (see Fig. 1) [11, 18, 27]. Each robot operates through the use of two controlling software layers—a high-level layer and a low-level one by means of an embedded swarm-enabling unit [7]. The low-level layer is tasked with way-point navigation and estimating the robot's position. This is done through the use of two DC motors attached with encoders, as well as an Inertial Measurement Unit (IMU). The IMU allows a robot to estimate its position using dead reckoning. The robot's estimated position is then sent by the low-level layer to the high-level layer that grants the robot its autonomy as part of the swarming system. The high-level layer consists of a low-power and low-bandwidth wireless XBee-PRO module as well as a Raspberry Pi zero that is responsible for implementing and processing the required swarming behaviors and managing the information received from the neighboring robots. From this, it generates a navigational way-point used by the low-level layer [18]. Communications between the robots are facilitated by the XBee module that broadcasts and receives positional information and objective function values.

The high-level layer is also responsible for filtering and only processing the data originating from neighboring units according to the selected value k of the inter-connecting topological network. The degree k has to be selected prior to running the test.

To emulate our conservative scenario with a zero-range sensor capability, we use a fictitious sensing unit. The sensory information—i.e. the value of the objective function—is provided at every instant by the Raspberry Pi zero, based on the estimated position of the robot in relation with the prescribed trajectory of the moving target, which moves randomly in the search-space at speed v.

3 SIMULATION RESULTS

3.1 Simulation Setup

A two-dimensional square search-space (dimensions $L \times L$) was considered with a disc-shape target having a fixed radius $\rho = L/20$

(see Fig. 2). The target moves randomly about the search-space at constant speed v. If an agent falls within the radius of the target, it is assigned an objective function value of -1, and zero otherwise. This models a binary objective function, in which agents either have are fully informed of the target's position or have no information at all. In our simulations, the speed of the target can vary between 10 and 50 non-dimensional units, compared to the baseline velocity value of 10 for the particles. All simulation runs lasted for a total of 100,000 time-steps to ensure statistically steady conditions. With such long simulation runs, we were able to calculate the three performance metrics with a very low variability between two different runs (below 0.3% for the tracking performance and 0.4% for the cumulative velocity fluctuation magnitude). The code and generated data-sets are available on Github [1].

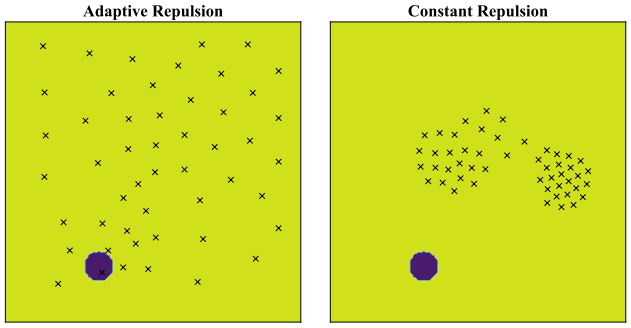

Figure 2: (Left) Agents using adaptive repulsion being blocked by the search-space boundaries and neighboring agents; (Right) agents remaining in place after losing track of the target with the low exploratory behavior in the presence of low constant repulsion.

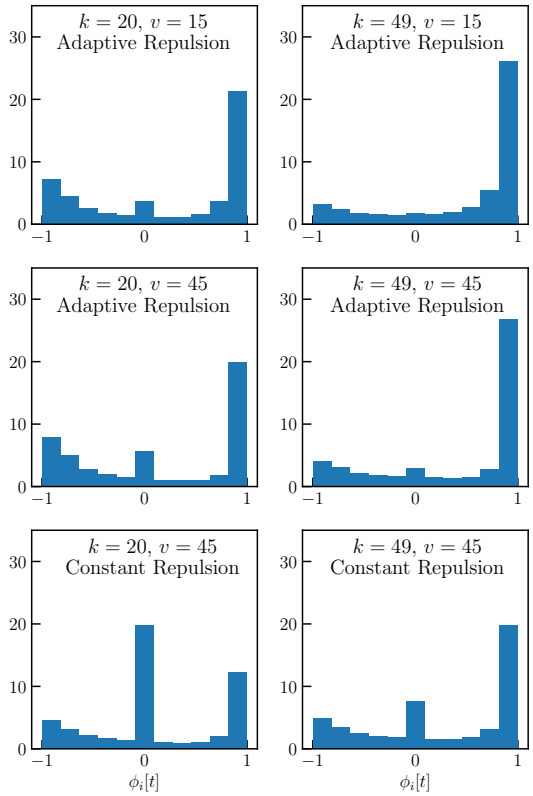

Figure 3: Distribution of the heading-bearing correlations $\phi_i[t] = \cos\theta_i[t]$ (x-axis) for different levels of the swarm connectivity (k), target speeds (v). The "Adaptive Repulsion" corresponds to our EED strategy, while the "Constant Repulsion" considers the minimal static repulsion at $a_{R,\min}$.

3.2 Heading-Bearing Correlation

We consider the distribution of the heading-bearing correlation $\phi_i[t] = \cos\theta_i[t]$ for all agents i in the swarm and over the entire duration of the simulation (100,000 time-steps). The histograms of this heading-bearing correlation allows for the visualization of the swarm's EED as discussed in Sec. 2.2.2. This was done by recording the heading-bearing correlations for all agents and iterations. The weight of each bin was then divided by the total number of iterations to give a time-averaged histogram for the entire swarm.

Three observations can be made from Fig 3. First, it can be seen that as the level of connectivity k increases, the level of exploration of the environment decreases in favor of an increase in exploitative actions, regardless of the type of repulsion used. The "Adaptive Repulsion" corresponds to our EED strategy, while the "Constant Repulsion" refers to a static repulsion with the minimum level of repulsion $a_{R,\min}$. This shift toward a more exploitative collective behavior when increasing k is robust and observed for all speeds v of the target considered. It is characterized by a higher proportion of agents having large and positive heading-bearing correlations. This indicates that more agents are moving toward the target rather than exploring the environment. Although this higher exploitation of the target may at first seem preferable, it is actually detrimental as the simulation unfolds. Indeed, it leads to an aggregation of the agents near the target, thereby leaving large swaths of the search-space empty of units when the target moves away from the "clustered" swarm (see Fig. 2 (Right)).

Second, the amount of exploitation carried out by the swarm decreases while the amount of exploration increases with increasing target speed v. This is characterized by the reduction in the number of agents with a positive heading-bearing correlation. The increased exploration is caused by the swarm losing track of the target more frequently, causing the swarming agents to disperse from the target's last known location.

Finally, it can be seen that there are agents with a heading-bearing correlation value of 0. This is more prominent at higher velocities and when using constant repulsion. This is caused by agents that are stationary or travelling at very low speeds. The reason for the near stagnation of the agents depends on the type of repulsion used. With the constant repulsion strategy, when the swarm loses track of the target, the agents hold their positions and remain in the vicinity of the target's last known location unless the randomly moving target crosses the path of one agent. In contrast,

[1] https://github.com/hianlee/kNNTargetSearch

while using the adaptive repulsion strategy, the agents disperse from the target's last known location. However, their movement is eventually restricted by the search-space boundaries and the blocking effect caused by a combination of a high repulsion strength and the presence of neighboring agents. The stagnant agents using the two repulsion strategies can be seen in Fig. 2.

3.3 Time on Target

As expected, the overall tracking performance of the swarm, as measured by the time on target (see Sec. 2.2.3) decreases with the target speed v for all values k of the swarm connectivity, and for both repulsion strategies (see Fig 4). This is a direct consequence of the target's competitive advantage in traveling faster than any individual agent—the constant speed of all agents equals 10, thus allowing it to "outrun" the searching agents.

The adaptive repulsion behavior adopted as part of our EED strategy yields a significant improvement over the constant repulsion case (see Fig. 5). For instance, at the highest target speed $v = 50$ and for $k = 20$, the tracking performance improves by approximately 40% (see Fig. 4). This enhancement of the tracking capability stems from to the increased exploratory behavior promoted by our adaptive repulsion scheme. Thanks to this, the swarm exhibits an improved collective search behavior upon losing track of the target. In turn, this notably shortens the time required to re-engage the target. As already highlighted, the reduced exploratory dynamics with the constant repulsion scheme leads to the swarm aggregating and a reduced ability to re-engage with the target.

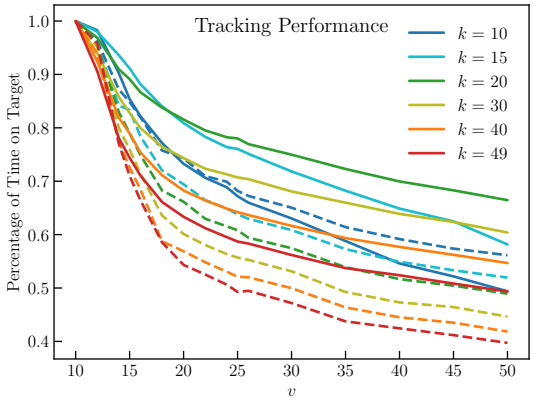

Figure 4: Tracking performance with varying target speed v and for different levels of connectivity k while using adaptive repulsion (solid lines) and constant repulsion (dashed lines). Particles travel at a speed of $v = 10$.

Figure 5 reveals the existence of an optimal degree $k^* \sim 20$ for the interaction network at which the tracking performance peaks. Starting from $k = 10$, an increase in the level of connectivity among swarming agents improves up to k^*. Beyond that value, more connectivity ends up being detrimental to the collective tracking performance. In the limit of the classically used all-to-all connectivity ($k = N - 1 = 49$), the collective tracking performance happens to be the lowest. Interestingly, the dynamics with the constant repulsion scheme fails to exhibit such an optimal connectivity for the

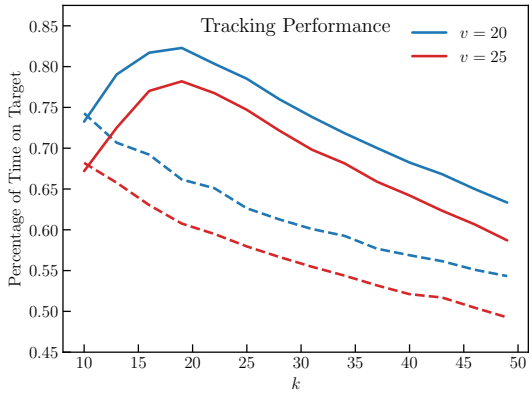

Figure 5: Tracking performance with varying levels of connectivity k for two different target velocities while using adaptive repulsion (solid lines) and constant repulsion (dashed lines).

tracking performance. Essentially, in both cases, a too high level of connectivity induces a stronger exploitative behavior—e.g. herding behavior—detrimental to the overall performance of the swarm.

3.4 Cumulative Velocity Fluctuations

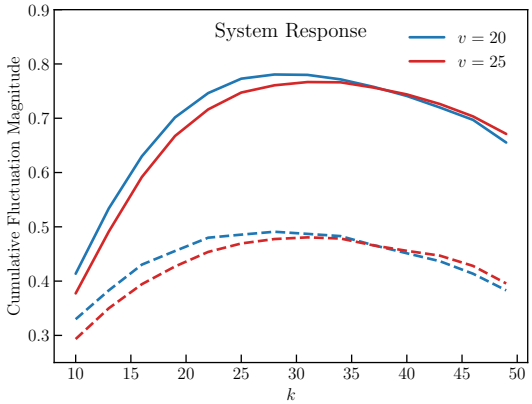

Figure 6: Cumulative velocity fluctuation magnitude plot with varying levels of connectivity k and at two different target speeds v while using adaptive repulsion (solid line) and constant repulsion (dashed line).

As explained in Sec. 2.2.1, the cumulative velocity fluctuation magnitude quantifies the collective response of the swarm to dynamic changes in the search-space; in this case the temporal variations of the objective function represent the target. It can therefore be considered as a proxy for a quantitative measure of the swarm intelligence activity.

The results for this more encompassing metric are consistent with those for the tracking performance. Indeed, Fig. 6 shows a significant increase in the swarm's activity with the adaptive repulsion scheme as compared to constant repulsion case.

The curves in Fig. 6 also present a peak in system's response at intermediate levels of connectivity. The peak is more clearly marked for the collective response obtained with the adaptive repulsion scheme. However, these peaks occur at values of $k > k^*$ previously identified for the collective tracking performance. When the swarm is connected for optimal response, the entire swarm is attracted to the target, leaving very few to no agents exploring the environment. Eventually, the pursuing swarm is unable to keep up and loses track of the target. In contrast, when the swarm is connected for optimal tracking performance, there is still a portion of the agents carrying out exploration of the environment. As such, when the initial group of agents are outrun by the target, the agents carrying out exploration are able to continue the tracking task, leading to a better overall system tracking performance.

Lastly, Fig. 6 reveals that there is a slight increase in the k value required for optimal system's response when the target velocity increases. This observation seems sensible as the level of connectivity k influences the effectiveness of information exchanges leading to the collective response. When the swarm encounters dynamic changes at different time-scales—here the two different target velocities—it needs to adjust its communication channel to optimally respond. Moreover, to generate the highest level of fluctuations in the swarm, as many agents as possible must be switching as often as possible from exploitative actions to exploratory ones.

4 SWARM ROBOTIC EXPERIMENTS

4.1 Experimental Conditions

At the start of each experimental test, 8 robots were arranged in two lines in the middle of an open search-space (see Fig. 7, left column). All the robots were able to broadcast their estimated position within the search space, as well as their objective function value. A target shaped as a disc of radius 1 m was simulated to be in constant random motion around the search area. The robots and the target traveled at the same speed of 10 cm $\cdot$ s^{-1}. It is important to note, however, that the fictitious target has an infinite maneuverability compared to the finite one of our robots. Hence, even if the individual robots have the same speed as the target, they are effectively much slower to respond changes in their heading owing to their low maneuverability. In practice, the dynamic target is implemented by generating a list of target positions and having the robots calculate the target's position based on the elapsed time. The robots then compare their respective position against the target's position. If a robot is within the target area, it would assign itself an objective function value of -1 and 0 otherwise.

The experiments were run for two different connectivity levels ($k = 3$ and $k = 6$, keeping in mind that $k = N - 1 = 7$ results in all-to-all connectivity) and using both repulsion schemes tested numerically. Each test lasted for slightly over a minute each and was repeated 5 times for each of the considered cases. This allows us to assume that the robots are able to perfectly estimate their position within the search space.

It is important stressing that, aside from the logging of data for the path reconstruction of the robots, our swarm robotic testbed operates in the complete absence of any supporting infrastructure [18]. In all the experiments carried out, the localization, cooperative control, and computing tasks were performed by the robotic units in a fully decentralized manner.

4.2 Experimental Results

Our swarm robotic setup is capable of monitoring and recording the robots' and target's positions and movements throughout the experiment (see Fig. 7). The robots are first initialized in the middle of the search-space (Fig. 7 (Left)). Subsequently, under the influence of the adaptive repulsion scheme, they start dispersing from their original locations. When a robot engages the target, its k topological neighbors start moving toward the target (Fig. 7 (Center)). If a robot is outrun by the target and one of its neighbors is still within the target region, it is able to change its heading and move towards the target's location (Fig. 7 (Right)).

4.2.1 Tracking Performance. Table 1 shows the percentage of time when there is at least one robot engaging the target, along with the 99% confidence intervals. This allows us to quantify the tracking performance of the robotic swarm. From the results, we can see that there is no significant difference between the two repulsion schemes. There seems to be a decrease in the tracking performance when going from $k = 3$ to $k = 6$, but given the relatively high variability in the results for $k = 3$, it is impossible to come up with any definite conclusions.

Although our visual observations of the collective dynamics showed noticeable differences for different k values, and between different repulsion schemes, the quantitative results for the tracking performance are not congruent with the simulations. This can be attributed to four reasons: (1) the small number of robots used within the swarm, (2) the short time duration of each test, (3) the relatively short amount of time the swarm is able to engage the target, (4) the relative high speed of the target given that the finite maneuverability of the robots as compared with the particles in the simulations.

Table 1: Tracking performance of the swarm robotic system.

Repulsion Type	$k = 3$	$k = 6$
Adaptive Repulsion	0.26 ± 0.20	0.17 ± 0.05
Constant Repulsion	0.31 ± 0.34	0.24 ± 0.13

4.2.2 System Response. From the same experimental results used to quantify the tracking performance, the time-averaged cumulative velocity fluctuations of the swarm was calculated, and hence its level of response to the dynamic target. Table 2 contains the calculated response levels, along with the 99% confidence intervals. These results unambiguously show that the adaptive repulsion scheme significantly increases the response of the swarm to the target ($p < 0.01$) for both connectivity levels considered. In addition, we measure a noticeable drop in collective response with both repulsion schemes when increasing the network connectivity from $k = 3$ to $k = 6$. This latter observation is fully consistent with the simulation results (see Fig. 6).

It should also be noted that the time-averaged cumulative velocity fluctuation magnitudes obtained experimentally are higher

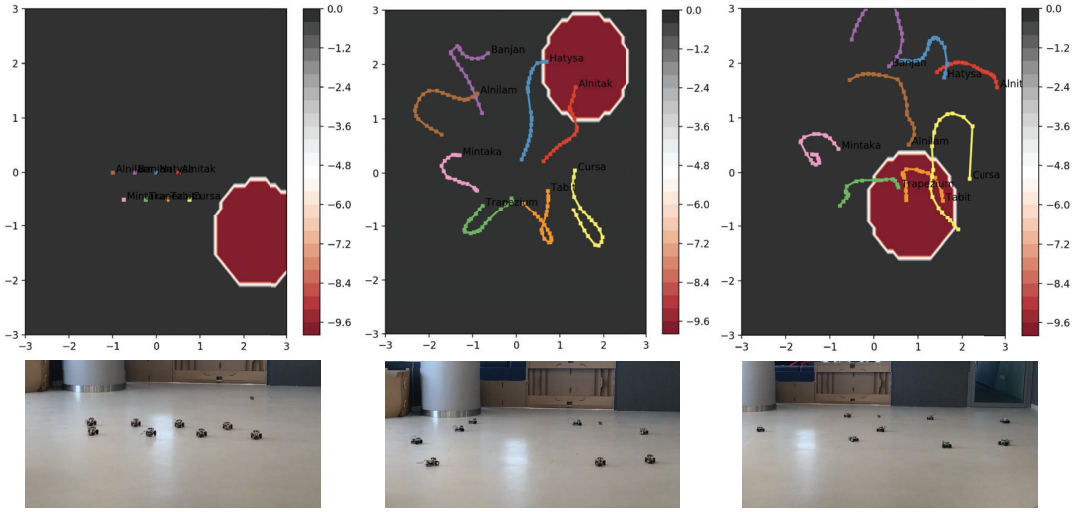

Figure 7: (Top) Positions of the target, the robots, and path reconstruction of the robots as observed through our monitoring system. (Bottom) Photos from the physical experiment using a connectivity of $k = 6$ and adaptive repulsion. Each column corresponds to a different instant during the collective search and tracking run.

Table 2: Response of the swarm robotic system to a dynamic target.

Repulsion Type	$k = 3$	$k = 6$
Adaptive Repulsion	1.36 ± 0.07	1.23 ± 0.03
Constant Repulsion	1.23 ± 0.03	1.05 ± 0.08

than those calculated for the simulations. This is expected given that the robots are unable to instantaneously change their speed and direction of travel unlike the agents in the simulations.

5 DISCUSSION

The application of robots to a dynamic search and tracking task is a challenging problem, in particular when considering a target moving faster than the individual units. While several strategies involving swarming MRS have been explored, the effect of changing a swarm's connectivity has not been made use of. In this work, we presented a decentralized swarming strategy with an adjustable balance of exploratory/exploitative collective actions. This balance in the EED is achieved by means of an adaptive repulsion scheme combined with a PSO-based algorithm using a topological distance k-nearest neighbor graph. This EED is tested on the challenging problem of search and tracking of a fast moving target.

To characterize the performance of the swarm, we proposed three swarm performance metrics. These are the cumulative velocity fluctuation magnitude, quantifying the amount of collective activity in the swarm, the heading-bearing correlation offering insights into the exploration/exploitation balance, and the time on target, which measures the proportion of time during which the swarm is engaged with the target.

Using these metrics, we showed that increasing k within the swarm causes it to favor exploitative dynamics over exploratory ones. This high amount of exploitation is detrimental to the swarm's tracking performance as it leads to agents clustering near the target. This leaves large swaths of empty space when the target outruns the aggregated swarm. By implementing the adaptive repulsion strategy, the amount of exploration carried out by the swarm is increased, thus leading to improved search dynamics upon losing track of the target. This reduces the amount of time required for the swarm to re-engage with the target, thereby improving its tracking performance.

The performance metrics also revealed the presence of an optimal degree $k^* \sim 20$ for maximum tracking performance, which is different from the degree required for maximum collective response. The decreased tracking performance at high levels of system response is due to the entire swarm being attracted to the target when the network is adjusted for maximum collective activity. This leaves few to no agents exploring the environment, leading the swarm unable to keep up with a faster moving target.

Physical tests were carried out using a swarm robotic system comprising 8 units. No firm conclusions could be drawn from the tracking performance of the swarm due to the high variability in the results. However, the ability of the adaptive repulsion strategy to significantly improve the swarm's response to a dynamic target was clearly demonstrated. In addition, a reduction in the swarm's collective response was observed when increasing its connectivity level k, which is in complete agreement with the simulation results.

ACKNOWLEDGMENTS

This research is funded by Thales Solutions Asia Pte Ltd, under the Economic Development Board - Industrial Postgraduate Programme (EDB-IPP). We thank H. Shankar for his help and support in performing the preliminary simulations and robot tests.

REFERENCES

[1] Alessandro Attanasi, Andrea Cavagna, Lorenzo Del Castello, Irene Giardina, Stefania Melillo, Leonardo Parisi, Oliver Pohl, Bruno Rossaro, Edward Shen, Edmondo Silvestri, et al. 2014. Collective behaviour without collective order in wild swarms of midges. *PLoS Comput. Biol.* 10, 7 (2014), e1003697.

[2] Michele Ballerini, Nicola Cabibbo, Raphael Candelier, Andrea Cavagna, Evaristo Cisbani, Irene Giardina, Vivien Lecomte, Alberto Orlandi, Giorgio Parisi, Andrea Procaccini, Massimiliano Viale, and Vladimir Zdravkovic. 2008. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the National Academy of Sciences* 105, 4 (2008), 1232–1237. https://doi.org/10.1073/pnas.0711437105

[3] Tim Blackwell and Peter Bentley. 2002. Dynamic Search with Charged Swarms. In *Proceedings of the Genetic and Evolutionary Computation Conference*. New York City, NY, USA, 19–26. https://doi.org/10.1.1.97.9635

[4] Tim Blackwell and Jim Kennedy. 2018. Impact of communication topology in particle swarm optimization. *IEEE Transactions on Evolutionary Computation* 23, 4 (2018), 689–702. https://doi.org/10.1109/TEVC.2018.2880894

[5] Roland Bouffanais. 2016. *Design and Control of Swarm Dynamics*. Springer, Heidelberg, Germany. https://doi.org/10.1007/978-981-287-751-2

[6] A. Cavagna, A. Cimarelli, I. Giardina, G. Parisi, R. Santagati, F. Stefanini, and M. Viale. 2010. Scale-free correlations in starling flocks. *Proceedings of the National Academy of Sciences* 107, 26 (2010), 11865–11870. https://doi.org/10.1073/pnas.1005766107

[7] Mohammadreza Chamanbaz, David Mateo, Brandon M. Zoss, Grgur Tokić, Erik Wilhelm, Roland Bouffanais, and Dick K. P. Yue. 2017. Swarm-Enabling Technology for Multi-Robot Systems. *Front. Robot. AI* 4 (2017), 12. https://doi.org/10.3389/frobt.2017.00012

[8] Charles Coquet, Clement Aubry, Andreas Arnold, and Pierre-jean Bouvet. 2019. A Local Charged Particle Swarm Optimization to Track an Underwater Mobile Source. In *OCEANS 2019 - Marseille*. IEEE, Marseille, France. https://doi.org/10.1109/OCEANSE.2019.8867527

[9] Micael S. Couceiro, Rui P. Rocha, and Nuno M.F. Ferreira. 2011. A Novel Multi-Robot Exploration Approach based on Particle Swarm Optimization Algorithms. In *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. IEEE, Kyoto, Japan, 327–332. https://doi.org/10.1109/SSRR.2011.6106751

[10] Micael S. Couceiro, Patricia A. Vargas, Rui P. Rocha, and Nuno M.F. Ferreira. 2014. Benchmark of swarm robotics distributed techniques in a search task. *Robotics and Autonomous Systems* 62, 2 (2014), 200–213. https://doi.org/10.1016/j.robot.2013.10.004

[11] Audelia G. Dharmawan, Priti Xavier, Hassan H. Hariri, Gim Song Soh, Avinash Baji, Roland Bouffanais, Shaohui Foong, Hong Yee Low, and Kristin L. Wood. 2019. Design, Modeling, and Experimentation of a Bio-Inspired Miniature Climbing Robot With Bilayer Dry Adhesives. *J. Mech. Rob.* 11, 2 (02 2019). https://doi.org/10.1115/1.4042457 020902.

[12] Russell Eberhart and Yuhui Shi. 2001. Tracking and Optimizing Dynamic Systems with Particle Swarms. In *Proceedings of the 2001 Congress on Evolutionary Computation*. IEEE, Seoul, South Korea, 94–100. https://doi.org/10.1109/cec.2001.934376

[13] Tomonari Furukawa, Frederic Bourgault, Benjamin Lavis, and Hugh F. Durrant-Whyte. 2006. Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets. In *Proceedings - IEEE International Conference on Robotics and Automation*. Orlando, Florida, 2521–2526. https://doi.org/10.1109/ROBOT.2006.1642081

[14] Xiaohui Hu and Russell C. Eberhart. 2002. Adaptive particle swarm optimization: Detection and response to dynamic systems. In *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, Vol. 2. Honolulu, HI, USA, 1666–1670. https://doi.org/10.1109/CEC.2002.1004492

[15] Wisnu Jatmiko, Kosuke Sekiyama, and Toshio Fukuda. 2007. A PSO-Based Mobile Robot for Odor Source Localization in Dynamic Advection-Diffusion with Obstacles Environment: Theory, Simulation and Measurement. *IEEE Computational Intelligence Magazine* 2, 2 (2007), 37–51. https://doi.org/10.1109/MCI.2007.353419

[16] Ahmad Rezaee Jordehi. 2014. Particle Swarm Optimisation for Dynamic Optimisation Problems: A Review. *Neural Computing and Applications* 25 (2014), 1507–1516. https://doi.org/10.1007/s00521-014-1661-6

[17] James Kennedy and Russell Eberhart. 1995. Particle Swarm Optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*. IEEE, Perth, WA, Australia, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

[18] Jabez Leong Kit, Audelia G Dharmawan, David Mateo, Shaohui Foong, Gim Song Soh, Roland Bouffanais, and Kristin L Wood. 2019. Decentralized Multi-Floor Exploration by a Swarm of Miniature Robots Teaming with Wall-Climbing Units. In *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, New Brunswick, NJ, USA, 195–201. https://doi.org/10.1109/MRS.2019.8901058

[19] Qunfeng Liu, Wenhong Wei, Huaqiang Yuan, and Zhi-hui Zhan. 2016. Topology Selection for Particle Swarm Optimization. *Information Sciences* 363 (2016). https://doi.org/10.1016/j.ins.2016.04.050

[20] Ruochen Liu, Jianxia Li, Jing Fan, Caihong Mu, and Licheng Jiao. 2017. A co-evolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization. *European Journal of Operational Research* 261, 3 (2017), 1028–1051. https://doi.org/10.1016/j.ejor.2017.03.048

[21] David Mateo, Nikolaj Horsevad, Vahid Hassani, Mohammadreza Chamanbaz, and Roland Bouffanais. 2019. Optimal Network Topology for Effective Collective Response. *Science Advances* 5, 4 (2019), eaau0999. https://doi.org/10.1126/sciadv.aau099

[22] David Mateo, Yoke Kong Kuan, and Roland Bouffanais. 2017. Effect of Correlations in Swarms on Collective Response. *Scientific Reports* 7, 10388 (2017). https://doi.org/10.1038/s41598-017-09830-w

[23] Jose Medina, Andrey Khlapov, Stephanie Forero, Scott Jagolinzer, and Sabri Tosunoglu. 2017. NASA Swarmathon Search and Rescue. In *30th Florida Conference on Recent Advances in Robotics*. Boca Raton, FL, USA. http://www.eng.fiu.edu/mme/robotics/elib/2017FCRAR/FCRAR2017-Medina-SwarmSearchRescue.pdf

[24] Malika Meghjani, Sandeep Manjanna, and Gregory Dudek. 2016. Multi-target rendezvous search. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'16)*. Daejeon, South Korea. https://doi.org/10.1109/IROS.2016.7759404

[25] Jim Pugh and Alcherio Martinoli. 2007. Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization. In *2007 IEEE Swarm Intelligence Symposium*. IEEE, Honolulu, HI, USA, 332–339. https://doi.org/10.1109/SIS.2007.367956

[26] Sara Brin Rosenthal, Colin R Twomey, Andrew T Hartnett, Hai Shan Wu, and Iain D Couzin. 2015. Revealing the hidden networks of interaction in mobile animal groups allows prediction of complex behavioral contagion. *Proceedings of the National Academy of Sciences* 112, 15 (2015), 4690–4695.

[27] Jugesh Sundram, Duong Van Nguyen, Gim Song Soh, Roland Bouffanais, and Kristin Wood. 2019. Development of a Miniature Robot for Multi-robot Occupancy Grid Mapping. In *ICARM 2018 - 2018 3rd International Conference on Advanced Robotics and Mechatronics*. IEEE, Singapore, 414–419. https://doi.org/10.1109/ICARM.2018.8610745

[28] Francesco Vallegra, David Mateo, Grgur Tokić, Roland Bouffanais, and Dick K. P. Yue. 2018. Gradual Collective Upgrade of a Swarm of Autonomous Buoys for Dynamic Ocean Monitoring. In *IEEE-MTS OCEANS 2018*. Charleston, SC, USA. https://doi.org/10.1109/OCEANS.2018.8604642

[29] Yuting Yan, Ru Zhang, Ji Wang, and Juming Li. 2018. Modified PSO algorithms with "Request and Reset " for leak source localization using multiple robots. *Neurocomputing* 292 (2018), 82–90. https://doi.org/10.1016/j.neucom.2018.02.078

[30] Jian Yang, Xin Wang, and Peter Bauer. 2019. Extended PSO Based Collaborative Searching for Robotic Swarms With Practical Constraints. *IEEE Access* 7 (2019), 76328–76341. https://doi.org/10.1109/access.2019.2921621

[31] Brandon M. Zoss, David Mateo, Yoke Kong Kuan, Grgur Tokić, Mohammadreza Chamanbaz, Louis Goh, Francesco Vallegra, Roland Bouffanais, and Dick K. P. Yue. 2018. Distributed system of autonomous buoys for scalable deployment and monitoring of large waterbodies. *Autonomous Robots* 42, 8 (2018), 1669–1689. https://doi.org/10.1007/s10514-018-9702-0