A Hybrid PSO Algorithm for Multi-robot Target Search and Decision Awareness

Julia T. Ebert[1], Florian Berlinger[2], Bahar Haghighat[2], and Radhika Nagpal[2]

***Abstract*— Groups of robots can be tasked with identifying a location in an environment where a feature cue is past a threshold, then disseminating this information throughout the group – such as identifying a high-enough elevation location to place a communications tower. This is a continuous-cue target search, where multi-robot search algorithms like particle swarm optimization (PSO) can improve search time through parallelization. However, many robots lack global communication in large spaces, and PSO-based algorithms often fail to consider how robots disseminate target knowledge after a single robot locates it. We present a two-stage hybrid algorithm to solve this task: (1) locating a target with a variation of PSO, and (2) moving to maximize target knowledge across the group. We conducted parameter sweep simulations of up to 32 robots in a grid-based grayscale environment. Pre-decision, we find that PSO with a variable velocity update interval improves target localization. In the post-decision phase, we show that dispersion is the fastest strategy to communicate with all other robots. Our algorithm is also competitive with a coverage sweep benchmark, while requiring significantly less inter-individual coordination.**

I. INTRODUCTION

Robot collectives can work together to investigate features of an unknown environment, such as determining the highest elevation or most stable location in an environment. When robots in a swarm identify such locations and share the information across the whole group, the group can chain together actions into complex behavior. For example, a group of robots could be tasked with building a human habitat on Mars: first, they could collectively classify an environment to determine whether it is a suitable region (as in [1]), then identify a specific location in the area with stable enough ground, then begin construction. This problem appears in many scenarios, such as identifying a high enough elevation to place a communications tower or a weak point in a structure for repair. In each case, there is scalar feature that can be detected at all points in the environment, and robots can identify a location that is close enough to optimal (i.e., past a given threshold) to complete their search task.

This problem is difficult: the feature under investigation, such as elevation or ground stability, often has a non-convex distribution or cannot be well-modeled. To guarantee locating the global minimum, the optimal solution is exhaustive coverage [2]. In many real-world situations, though, a location with a value past a threshold can be identified with non-coverage search algorithms. A robot collective can then speed target localization by adding sensors. However, without global communication, robots would need to independently find the target, wasting time and energy. Instead, robots can move to facilitate communication, which enables collective decision-awareness in the group and allows them to chain collective tasks such as habitat construction.

In this paper, we consider a specific case of this collective search problem: simulated robots seek to (1) locate a position in an environment where a value is below a threshold and (2) disseminate that information to the entire group, completing their task when all robots are aware of the target and return to their deployment location. We model the environment as terrain generated by 2D Perlin noise to create varying difficulties. We present a hybrid algorithm designed to balance exploration of the environment, exploitation of previous observations, and communication with other robots in the swarm. We show that this algorithm operates successfully within realistic constraints of real robot systems, such as limited run duration due to battery constraints, minimizing use of energy-intensive communication, and not requiring tight inter-individual coordination.

II. RELATED WORK

Multi-robot target search algorithms take a variety of approaches, depending on the cue a target propagates through the environment.

When a cue is sparse, approaches typically rely on efficiently coordinating individual searches, such as discrete Markov chains [3] or variations on frontier search [4], [5]. If guarantees about target localization are required, this is best achieved by a complete coverage algorithm [6] like multi-robot boustrophedon coverage (i.e., a sweeping pattern) [7]. If the intermittent cue propagation can be modeled, as in odor or chemical source localization, this model can be employed in probabilistic search algorithms [8], including following an information gradient in infotaxis [9], [10], [11], or using a hidden Markov methods [12]. However, these rely on accurate models of the environmental feature and tend to fail if reality diverges from this [13].

Alternatively, when a cue for the target can be continuously sensed, it is easier to search without a model. If a gradient of the cue is available, gradient descent or chemotaxis allows a group or robots to find a global optimum, as in [14]. When a gradient alone is insufficient and the feature's distribution cannot be accurately modeled, this leaves heuristic search approaches to locating an optimum.

This work was supported by a Department of Energy Computational Science Graduate Fellowship (grant number DE-FG02-97ER25308) and the Office of Naval Research (grant number N00014-20-1-2320).

[1]Julia T. Ebert is with the John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts ebert@g.harvard.edu. [2]Florian Berlinger, Bahar Haghighat, and Radhika Nagpal are with the departments of Mechanical and Aerospace Engineering and Computer Science, Princeton University, Princeton, New Jersey.

The most common multi-robot heuristic search strategies are variations of particle swarm optimization (PSO). In the original form of PSO, abstract particle agents employ a biased random walk toward their individual and collective best observations, resulting in non-guaranteed convergence at a global optimum [15]. PSO has been applied to real multi-robot systems; [16], [17] demonstrated that PSO could be used to identify a source with convex feature distribution, despite limited movement speed and communication range. There are also hybrid algorithms that incorporate PSO into their search, including hybrid ant colony optimization (ACO/PSO), where virtual pheromone deposits augment direct communication [18]; PSO plus fruit fly optimization (MFPSO) to avoid local minima and improve search speed [19]; and adaptive robotics PSO (AR-PSO), which considers obstacle avoidance and a mechanism to escape local minima.

These search strategies typically have one of two possible stopping conditions: (1) when a single robot has located the target, or (2) in PSO, when all robots converge at the target. Without global communication, the first termination condition does not consider knowledge across the rest of the collective. PSO-based convergence requires all robots to identify the target by travelling to the location, rather than learning indirectly by communication. Without global communication, this limits the reach of the already-converged robots to aid convergence of the remaining robots.

To disseminate knowledge of the target without requiring convergence, robots could travel as a connected network. This can be accomplished with distributed spanning trees [20] or creating a k-connected network to maintain at least k neighbors [21]. Both of these algorithms can guarantee maintaining a network, but they are computationally heavy and unrealistic to implement on simple robots. A simpler alternative is Boids flocking [22], where robots create a flock with a target neighbor distance smaller than their communication range. This creates a looser, easier to implement network, though without guarantees.

III. METHODS

A. Problem Definition

We present a problem in which a group of simulated robots identify a location with a value below a threshold value v^*. Robots move in a bounded, monochrome arena composed of grid cells with values $v \in \mathbb{N}\ \{0..255\}$, as shown in Fig. 1. The goal is for each robot to identify a target grid location X where the value $v \leq v^*$; this knowledge obtained from its own observations or communicated by other robots.

Robots begin at a home position in the corner of the environment, equivalent to being deployed together, and the simulation is considered complete when all robots have returned to the home position. A robot will only return to its home position when it knows of a target location, and when it believes that all other robots also know it. This collective awareness is essential for any task where the robots must collectively change their action after a decision is made, including returning to a home position for collection. If robots return before all others make a decision, this leaves the remaining robot(s) to locate the source alone. Note that there can be more than one location with $v \leq v^*$. These discrepancies can be reconciled when robots collect themselves, because they return to the same location.

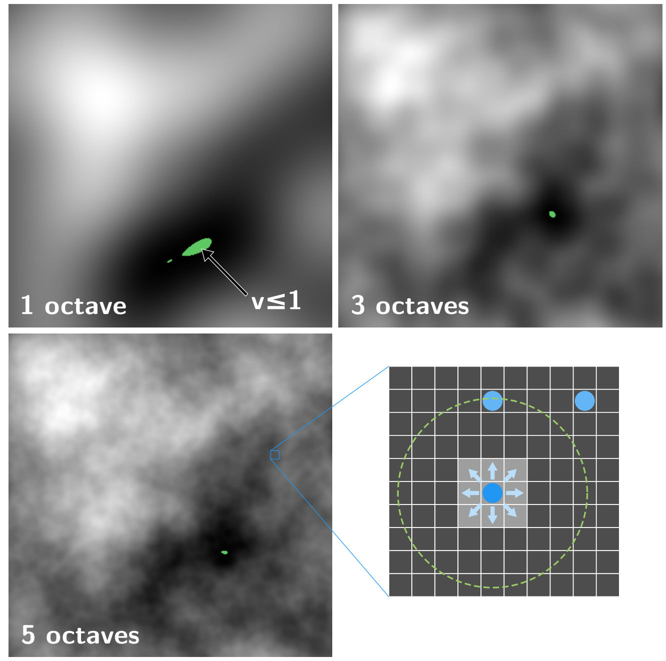

Fig. 1. Examples of simulated environments generated by 1, 3, and 5 octaves of Perlin noise, with overlay of values $v \leq v^*$ for decision threshold $v^* = 1$. Robots must identify a location with a value below v^*. Bottom right: 10×10 cell segment of environment, showing robots (blue dots), movement allowed in a single step (blue arrows), and example communication range $d_c = 4$ (green circle).

B. Robot Model

We use an abstract robot model that can move and observe in the grid world. Each robot occupies a single cell, such that the grid discretization represents the sensing resolution of a robot; multiple robots may occupy the same cell. At each time step, a robot can move to any of the neighboring eight grid cells, while maintaining knowledge of its position. Accurate localization is a reasonable assumption, given the ubiquity of GPS outdoors, increasing accuracy of SLAM algorithms. While [16] demonstrated that it is possible to perform PSO on robots without global positioning, its presence enables accurate memory and reporting of the target location. Each robot can sense the value in its current cell and the neighboring eight cells, enabling gradient estimates. Robots can also communicate with neighbors within a communication range d_c, which we vary in the experiments. This reflects a variety of communication with different ranges, from local line-of-sight light beaconing (as in Kilobots [23]) to Bluetooth to cellular communication.

C. Environment Model

Environments are generated by monochrome, multi-octave 2D Perlin noise [24], which is normalized to cover $[0, 256)$. This tunable procedure generates smooth, multi-scale textures and is used in computer graphics to model naturally occurring phenomena, such as terrain and smoke. We chose it as an abstract representation for many possible types of features that could be investigated using the algorithms presented in this paper. The multi-scale nature of Perlin noise allows us to easily tune the environment complexity by adjusting the number of octaves (layers) to add smaller-scale noise in the environment.

IV. ALGORITHMS

We created a two-stage algorithm, where robots switch from target localization (pre-decision) to dissemination (post-decision). In addition, we compare performance to a benchmark in which robots conduct a pre-defined sweep over the environment.

A. Decision Algorithm

Movement: Each robot moves from the home corner to a random location in the environment and is then assigned a random virtual velocity V^t. Robots then switch to the movement algorithm described in Section IV-B. As long as the robot has not yet found a suitable target location X with $v \leq v^*$, the robot continues executing the pre-decision movement algorithm.

After identifying a location X with value $v \leq v^*$, a robot switches to one of three possible movement algorithms to disseminate the decision, as described in Section IV-C: (1) Flocking behavior with other robots, (2) dispersing away from nearby robots, or (3) continuing the pre-decision movement.

As robots can only move one cell per tick, executed paths are generated from V^t by Bresenham's line algorithm [25].

Observation: Every tick, robots observe the value v at their current position X^t. If the observed value is lower than any previous observation the robot made, it updates its personal best value v^{p}, its position X^{p}, and the observation time t^{p}; if lower than any value that it knows, the robot updates its global best value v^{g}, position X^{g}, and time t^{g}.

Communication: Throughout each trial, robots continuously communicate with any robots within the communication range d_{c}. Each robot maintains a table of the information received, containing the following for each transmitting robot: its ID, best value $v^{\mathrm{g}\prime}$, value's location $X^{\mathrm{g}\prime}$, tick the entry was added t_{add}, and the position X' and velocity V' of the transmitting robot when the message was received. Entries in the table expire and are removed after a duration t_{rx}, increases the robustness of the collective decision; if one or more robots prematurely returns to the origin or entered a failure state, the remaining robots would still be able to complete their collective decision without needing to hear from that robot again.

Robots send messages with the values described above, as well as their own neighbor table. When receiving a message, a robot adds or updates its table entry for the sending robot and incorporates the neighbor's table by updating with any newer values. Note that t_{add} does not change for entries in the received table, as this represents the time when the observation was first transmitted. For any new values, the receiving robot will update v^{g}, X^{g}, and t^{g} if a lower value was received.

Ending Conditions: There are two different experimental conditions determining when robots complete their run. In the collective awareness condition, robots conclude their run and return to the collection point when they believe that all robots know a target location. This is done by checking whether all robots in their current neighbor table know a location with a value below v^*. When all robots have returned, the trial is finished. In the second case, the robots have a fixed maximum duration $t_{\max}$. They may return to the collection point early if the collective awareness condition is met, but they will always return to the collection point by $t_{\max}$, regardless of collective awareness.

B. Pre-decision Movement

Robots balance exploration and exploitation with a combination of PSO and gradient descent, with a variable velocity update interval. For each update interval Δt, a robot generates a new intermediate velocity $V^{(*)}$:

$$V^{t+\Delta t(*)} = \omega V^t + c_{\mathrm{p}} r_{\mathrm{p}}^t \left(X^{\mathrm{p}} - X^t\right) + c_{\mathrm{g}} r_{\mathrm{g}}^t \left(X^{\mathrm{g}} - X^t\right) + c_{\mathrm{GD}} r_{\mathrm{GD}} \nabla f(X^t) \tag{1}$$

The first term, with inertia coefficient ω, limits the change in velocity that can occur in each update. The second term moves a robot toward its personally observed best location X^{p} from its current location X^t, while the third term does the same for the best location known to the robot X^{g}, either from its own observations or communication. These two terms have random coefficients $r_{\mathrm{p}}, r_{\mathrm{g}} \sim U(0,1)$, which make this a random walk biased toward the best known locations. If the fixed coefficients $c_{\mathrm{p}}, c_{\mathrm{g}}$ are too large, this walk will be biased too strongly toward a local minimum; if too small, the observed optima play a negligible role in the movement. The final gradient term allows for exploitation of local observations to move toward a minimum. For these grid-based robots, the gradient is the direction of the neighboring cell with the smallest value. This term also employs a random coefficient $r_{\mathrm{GD}} \sim U(0,1)$.

To generate the virtual velocity used for future updates, the intermediate velocity $V^{(*)}$ is normalized to a maximum speed $V_{\max}$:

$$V^{t+\Delta t} = \min\left(V^{t+\Delta t(*)}, V_{\max} \frac{V^{t+\Delta t(*)}}{\left\| V^{t+\Delta t(*)} \right\|}\right) \tag{2}$$

Each robot updates its virtual velocity after Δt ticks. Preliminary experiments showed that traditional PSO frequently failed by getting stuck in local minima. We therefore added this variable update interval Δt to escape local minima.

$$\Delta t = \min\left(128, t^{\mathrm{p}}\right) \tag{3}$$

If v^{p} was observed recently, this generates a local search around where the value was observed. To prevent capture in local minima, Δt increases when no better values are observed, which increases exploration. The maximum Δt of 128 ticks was selected from pilot experiments.

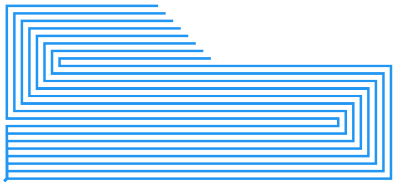

Fig. 2. Example paths of robots in benchmark coverage sweep. Robots sweep until locating where $v \leq v^*$ and all robots know the location.

C. Post-decision Movement

During the initial pre-decision search, robots spread out to explore different regions. We investigate whether a specialized movement strategy following an individual decision improves dissemination of this knowledge through the group. We propose two options: (1) flocking, which creates a loosely connected network and (2) dispersion, which causes mixing.

1) Flocking: Robots update their velocity according to Boids flocking strategy [22], which was created to simulate bird flocking. It has since been used extensively to create flocking behavior in robot swarms [26], [27], [28] because robots can prioritize maintaining a connected network without strict enforcement by a central controller. The Boids model updates the agents' velocities based on the combination of alignment toward, cohesion with, and separation from neighboring agents. Here, neighbors include all robots communicated with since the last velocity update. We define alignment as matching the sum of the neighbors' velocity vectors V_k, and model cohesion and separation by the Lennard-Jones force F_{LJ} [29]. A robot i updates its velocity based on the positions and velocities of its k neighbors:

$$V_i^{t+1(*)} = 2V_i^t + \tfrac{1}{N}\sum_{k=1}^{N} V_k^t + F_{i\text{LJ}} \quad (4)$$

and apply the normalization in Eq. 2.

$$F_{i\text{LJ}} = \tfrac{1}{N}\sum_{k=1}^{N}\left(-\left[a\left(\frac{d_{\text{t}}}{|r_{ik}|}\right)^{a} - 2b\left(\frac{d_{\text{t}}}{|r_{ik}|}\right)^{b}\right]\right)\hat{r}_{ik} \quad (5)$$

where r_{ik} is the vector between robots i and k, and d_{t} is the target distance between robots. We fix d_{t} at 75% of the communication range to maintain a communication network. a and b are constants that determine the intensity of the forces; we used the standard values of $a = 12$ and $b = 6$.

2) Dispersion: To disperse robots, we use only the separation/cohesion term of the velocity update in Eq. 4, but set the target distance d_t to $10 \cdot d_{\text{c}}$. Robots become close enough to communicate for a single tick and exchange messages, then separate again.

D. Benchmark Movement: Coverage Sweep

In addition to the algorithms described above, we also implemented a benchmark in which robots perform a collective lawnmower sweep over the environment, as shown in Fig. 2. Robots begin in the corner, then spread into a line separated by 2 cells to maximize coverage without overlapping sensing ranges. They then sweep over the whole environment until a target location is identified. If all robots are in a connected network (which occurs for all cases with $d_{\text{c}} > 4$), this information is disseminated to the group within a few ticks, and all return to the origin. If not in communication, the robots must sweep the entire environment before returning.

V. EXPERIMENTS

We conducted experiments in Kilosim, an open-source simulator we developed for high-throughput robot swarm simulations [30]. Additional code for this paper is available on GitHub [31]. All simulations were run in a 384×384 cell arena. This is large enough to allow a sparse density of robots, with large-scale features and local noise. The precise dimensions were chosen as a multiple of the robot group size to easily generate benchmark sweep paths.

In all simulations sets, we varied the following, which allow us to understand the effect of the swarm and environment on the algorithm:

- Number of robots n: $\{8, 16, 32\}$
- Communication range d_{c}: $\{4, 8, 16, 32, \text{global}\}$
- Environment octaves: $\{1, 3, 5\}$

The varied octaves used to generate the environments correspond to three different difficulties constructed from Perlin noise, as shown in Fig. 1, where each cell is a pixel in the generated image. The parameters of the texture generation were a frequency (scale) of 100, lacunarity (change of scale per octave) of 2.1, and persistence (change of intensity per octave) of 0.5. These were selected from pilot experiments to provide a variety of feature scales that influenced robot behavior. We generated 50 images per difficulty, to be used with the corresponding trial. In all conditions, we used a fixed threshold of $v^* = 1$ to simplify experiments.

A. Pre-decision Simulations

We first conducted a parameter sweep to choose parameter values for the pre-decision movement (Eq. 1). The goal was to identify values that minimized the time for a first robot to locate where $v \leq v^*$. In addition to the variables described above, our parameter sweep covered the following:

- PSO inertia ω: $\{0, 0.5, 0.75, 1.0, 10\}$
- PSO weights $c_{\text{p}}, c_{\text{g}}$: $\{0, 0.01, 0.025, 0.05, 0.1, 1\}$
- Gradient weight c_{GD}: $\{0, 4, 8, 16\}$
- Maximum virtual speed $V_{\max}$: $\{2, 25\}$

Parameter ranges were selected from pilot experiments. Note that the personal and collective PSO weights are paired, to constrain the size of the parameter sweep. We conducted 50 trials for each of the resulting 10,800 parameter combinations. Each trial was capped at 5,000 ticks; if a source was not found in that time, we considered it a failure.

B. Post-decision Simulations

After selecting the parameters for a single robot to locate the target, we conducted a parameter sweep for the post-decision strategy. We varied the movement strategy, as described in Section IV-C. The neighbor table timeout t_{rx} was fixed at 512 ticks to balance hearing from neighbors while avoiding unnecessary delays. For the collective awareness

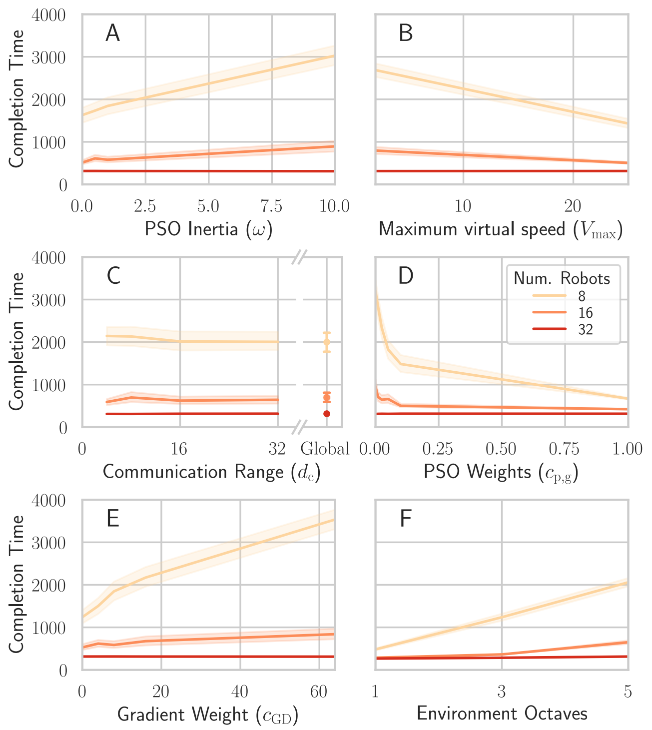

Fig. 3. Parameter effects on time for first robot to locate target, for 5 octave environment. Lines and shading show median and 25th/75th percentiles, respectively.

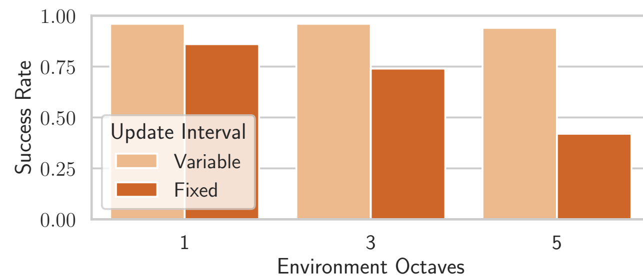

Fig. 4. Success rate of locating the target within 5,000 ticks, for our algorithm (variable update interval) and traditional PSO (fixed updated interval). The fixed update interval did not allow robots to escape local minima in the higher noise (higher octave) environments.

ending condition, trials were capped at 20,000 ticks. For the time-based ending condition, we set the maximum duration $t_{\max} = 8000$ ticks.

VI. RESULTS

A. Pre-decision

We first look at the effect of parameters on time for a single robot to locate a target, seen in Fig. 3. Across all conditions, the performance of a 32 robot collective was hardly impacted, likely due to the density of robots; regardless of the parameter selection, at least one robot was close enough to a target location to quickly identify it. This demonstrates that the algorithm is scalable. Increasing the number of robots is therefore the best way to improve performance, creating group that is robust to parameter selection.

The trends in parameter effects held across all environments, so here we present the results for environments with five octaves of Perlin noise, which is the most challenging, with small-scale noise and local minima. The parameter effects also become more pronounced for smaller groups. We found that inertia reduced performance (Fig. 3A), likely by minimizing the responsiveness of robots to locally observed information. In physical robots, inertia is often inevitable, and we see that plausible real-world inertia of $\omega = 1$ had a small impact on localization time. A higher maximum speed $V_{\max}$ (Fig. 3B) allows more variation in velocities, particularly when inertia exists while preventing runaway values that occur if velocity is unbounded.

Surprisingly, communication did not aid the search (Fig. 3C). As robots are exploring different regions, receiving information from robots exploring elsewhere can interfere with the local search. In fact, it could be advantageous to reduce or turn off communication before a robot makes a decision, as long-range communication is energy-intensive. This differs from traditional PSO, where global communication is assumed. Here, robots are not trying to congregate at the global maximum, but only identifying it. More significantly, robots have limited speed (unlike abstract particles), so physically distant information cannot be acted upon without a significant time delay to move to that location.

We see the most significant parameter benefit from increasing the PSO weights $c_{p,g}$ (Fig. 3D). Given that long-range communication was not beneficial, this shows that robots benefit most from acting on local observations. However, if robots only moved toward their best observed position, they could become trapped in local minima. We show below how our algorithm prevents this complication.

We also see that increasing the octaves of Perlin noise in the environment increased the difficulty of the task (Fig. 3F), likely due to more spatial noise (therefore increasing the number of local minima) and often fewer positions below the decision threshold, seen in Fig. 1. However, in all environments, we found that employing the gradient in the search strategy did not improve search times (Fig. 3E). We hypothesize that this reactive component did not provide additional benefit beyond PSO; when paired with the variable update interval, PSO already allowed robots to react quickly to local information. This also demonstrates that robots do not need the more-advanced ability to detect or estimate gradients to complete this type of search task.

In Fig. 4, we also see the benefits of our approach by comparing to conventional PSO, where the velocity update interval is fixed. We ran a subset of our experiments ($n = 8$, $d_c = 32$) with a fixed update interval and no gradient, and selected the parameters with the lowest median time to first target localization ($c_{p,g} = 1$, $\omega = 1$, $V_{\max} = 25$, $\Delta t = 1$). Our approach allowed robots to successfully identify the target in noisier environments by allowing them to escape local minima. For traditional PSO, the short update interval trapped robots in these minima, while longer update intervals

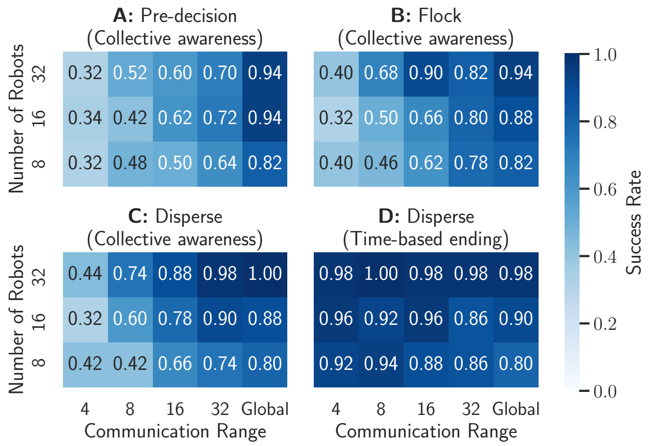

Fig. 5. **A-C:** Success rate by post-decision movement type. Dispersion resulted in higher success by allowing robots to communicate with more robots. **D:** Success rate for time-based ending condition using dispersion, with $t_{\max} = 8000$ ticks. Allowing time-based termination improved success, especially for low-communication regimes.

created overshooting instead of local investigation.

B. Post-decision

From the pre-decision results, we selected the best set of parameters to use for the post-decision simulations: $c_{\mathrm{p,g}} = 1$, $c_{\mathrm{GD}} = 0$, $\omega = 0$, $V_{\max} = 25$.

In Fig. 5A-C, we can compare the success of different post-decision strategies. While initially locating a target did not require large-scale communication, we see that the small groups with limited communication failed to consistently disseminate target information within the 20,000 tick time limit. Overall, communicating target information while continuing to perform the search algorithm yielded the worst performance; communication does not factor into this movement approach. Flocking performs better, adding a communication component that allows robots to maintain a loose network once they meet. This means that any information obtained by one robot in the flock will be known to the whole group. Fig. 6 shows that flocking maintained the highest count of neighbors heard from each tick because of the network created. However, this results in the robots covering a smaller area of the environment, meaning they are less likely to encounter individuals unaware of the target. While a pre-existing flock will allow information to be transmitted within the group, the limitation in this scenario is that the robots must *form* a flock, which is non-trivial for sparse robots.

Dispersion had the highest success rate, as the algorithm does not prioritize *maintaining* communication with the group, but communicating with as many individuals as possible. This can be inferred from the higher success rate; communicating with more individuals resulted in spreading target knowledge to more unique individuals, thus completing the trial within the time limit.

In Fig. 5D, we see that adding the additional constraint of a time limit to the post-decision dispersion improved the success. If one robot knows of the target location, it can be

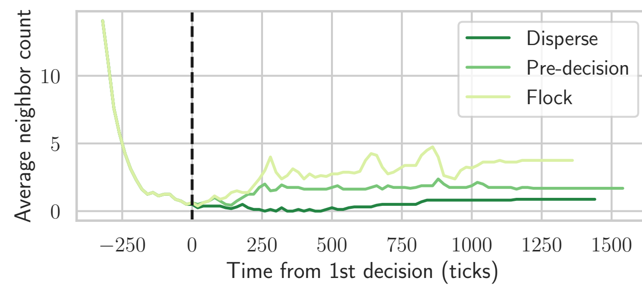

Fig. 6. Example average number of neighbors for different post-decision dissemination movement strategies. Dispersion had the lowest average neighbor count, and the highest success rate.

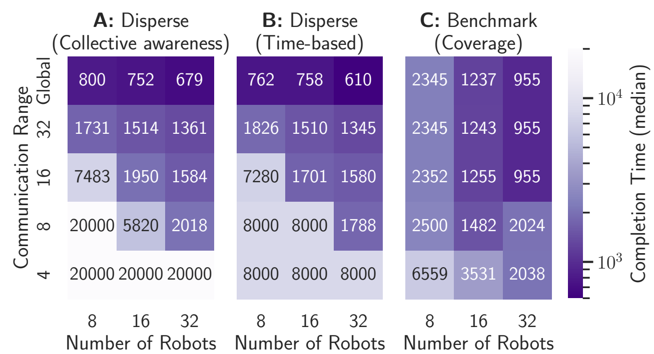

Fig. 7. Comparison of median task completion time for 3-octave environments, on a logarithmic scale. Collective awareness and time-base ending conditions both used dispersion post-decision, but adding a time limit to the search reduced average search time.

disseminated when all the robots are collected at the origin. This prevents cases where the task fails to complete because a small number of robots are not communicated with before the 20,000 tick time limit, which was most likely to occur in settings with small communication ranges. This time-based ending condition also represents a realistic system constraint, as robots typically have a limited mission duration due to battery constraints. In contrast to employing *only* a time-based ending condition 8,000 ticks, in Fig. 7B we see that allowing completion with the collective awareness condition allows for rapid decisions when communication is better – comparable to the collective awareness ending condition seen in Fig. 7A – but creates a backstop to prevent failures where a subset of robots do not learn of the target in the field.

In Fig. 7C, we see that the benchmark sweep is typically fastest, though for larger groups of robots, the hybrid algorithm is competitive. Because the benchmark is a coverage algorithm, it will also always locate a target. By maintaining a formation, any case with $d_{\mathrm{c}} > 4$ maintains a connected communication network; if one robot finds the target, all robots will quickly learn it and the task can be terminated. However, even with small communication ranges, we find that the sweep completes the task faster; the robots only need to cover the environment, rather than continuing to wander to communicate. With global communication, robots in Fig. 7A and B will complete the task as soon as a single robot locates the target, meaning that they only utilize the variable-update-interval PSO stage of the algorithm. Here they are faster than the benchmark because they more quickly explore different areas of the environment, while robots in the sweep are always in the same region. Despite the apparent success of the benchmark, it assumes perfect, synchronous motion of the robots, which is difficult to achieve in groups of physical robots. In contrast, our hybrid algorithm does not require synchronized movement, and unlike flocking, the post-decision dispersion requires no coordinated movement.

VII. CONCLUSION

We have shown an algorithm for the multi-stage task of robot target-searching with a continuous cue: locating the target, communicating this information to the rest of the group, and concluding the task by returning to their deployment position. This demonstrates the potential of creating complex behavior by thoughtfully combining variations on existing algorithms. In turn, this ability opens the possibility to employ simple robot collectives in autonomous tasks like inspection: robots in this scenario were able to complete the task without centralized control, global communication, or inter-robot motion coordination.

We found that a form of PSO with variable update intervals allowed robots to locate a target without large-scale communication. To disseminate this, dispersion proved best at spreading information, rather than forming a flock to create and maintain a communication network. This counter-intuitive result stems from two challenges: forming a flock is challenging for physically distributed robots, and maintaining network limits the ability to spread information across in a large environment. When we also included a realistic time constraint representative of battery limitations, we were able to nearly double the success rate for low-communication regimes, while maintaining fast decision-making for larger groups with larger communication ranges. For these groups, our algorithm was competitive with the benchmark sweep algorithm, but without tight constraints on coordination.

While this algorithm was demonstrated in simulation with an abstract environment model, we expect the results will hold on physical robots because it does not require complex coordination, movement, or sensing by robots. In future work, we plan to implement this algorithm on physical robots and extend it to more complex environments and cues beyond our Perlin-based terrain model. [32] has demonstrated that PSO can be conducted with obstacles and inconsistent signal sensing. We intend to apply our algorithm to fault inspection tasks containing similar challenges, with the goal of creating a system applicable to inspecting infrastructure such as bridges or space stations.

VIII. ACKNOWLEDGMENTS

We thank Bahar Haghaighat for discussions of PSO-based algorithms, and motivation and applications of the work.

REFERENCES

[1] J. Ebert, M. Gauci, F. Mallmann-Trenn, and R. Nagpal, "Bayes Bots: Collective Bayesian Decision-Making in Decentralized Robot Swarms," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2020, iSSN: 10504729.

[2] L. D. Stone, *Theory of optimal search*. New York, NY, USA: Academic Press, 1976, vol. 118.

[3] A. Shirsat, K. Elamvazhuthi, and S. Berman, "Multi-Robot Target Search using Probabilistic Consensus on Discrete Markov Chains," in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. Abu Dhabi, United Arab Emirates: IEEE, Nov. 2020, pp. 108–115. [Online]. Available: https://ieeexplore.ieee.org/document/9292589/

[4] A. Marjovi, J. G. Nunes, L. Marques, and A. de Almeida, "Multi-Robot Fire Searching in Unknown Environment," in *Field and Service Robotics*, B. Siciliano, O. Khatib, F. Groen, A. Howard, K. Iagnemma, and A. Kelly, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, vol. 62, pp. 341–351, series Title: Springer Tracts in Advanced Robotics. [Online]. Available: http://link.springer.com/10.1007/978-3-642-13408-1_31

[5] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376–386, June 2005. [Online]. Available: http://ieeexplore.ieee.org/document/1435481/

[6] C. Robin and S. Lacroix, "Multi-robot target detection and tracking: taxonomy and survey," *Autonomous Robots*, vol. 40, no. 4, pp. 729–760, Apr. 2016. [Online]. Available: http://link.springer.com/10.1007/s10514-015-9491-7

[7] I. Rekleitis, A. P. New, E. S. Rankin, and H. Choset, "Efficient Boustrophedon Multi-Robot Coverage: an algorithmic approach," *Annals of Mathematics and Artificial Intelligence*, vol. 52, no. 2-4, pp. 109–142, Apr. 2008. [Online]. Available: http://link.springer.com/10.1007/s10472-009-9120-2

[8] G. Kowadlo and R. A. Russell, "Robot Odor Localization: A Taxonomy and Survey," *The International Journal of Robotics Research*, vol. 27, no. 8, pp. 869–894, Aug. 2008. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0278364908095118

[9] M. Vergassola, E. Villermaux, and B. I. Shraiman, "'Infotaxis' as a strategy for searching without gradients." *Nature*, vol. 445, no. 7126, pp. 406–409, 2007, iSBN: 1476-4687 (Electronic)\r0028-0836 (Linking).

[10] C. Song, Y. He, B. Ristic, and X. Lei, "Collaborative infotaxis: Searching for a signal-emitting source based on particle filter and Gaussian fitting," *Robotics and Autonomous Systems*, vol. 125, p. 103414, Mar. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0921889019306451

[11] N. Voges, A. Chaffiol, P. Lucas, and D. Martinez, "Reactive Searching and Infotaxis in Odor Source Localization," *PLoS Computational Biology*, vol. 10, no. 10, p. e1003861, Oct. 2014. [Online]. Available: https://dx.plos.org/10.1371/journal.pcbi.1003861

[12] J. Farrell, Shuo Pang, and Wei Li, "Plume mapping via hidden markov methods," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 6, pp. 850–863, Dec. 2003. [Online]. Available: http://ieeexplore.ieee.org/document/1245262/

[13] B. Bayat, N. Crasta, A. Crespi, A. M. Pascoal, and A. Ijspeert, "Environmental monitoring using autonomous vehicles: a survey of recent searching techniques," *Current Opinion in Biotechnology*, vol. 45, pp. 76–84, June 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0958166916302312

[14] G. Sandini, G. Lucarini, and M. Varoli, "Gradient driven self-organizing systems," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '93)*, vol. 1. Yokohama, Japan: IEEE, 1993, pp. 429–432. [Online]. Available: http://ieeexplore.ieee.org/document/583132/

[15] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[16] J. Pugh and A. Martinoli, "Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization," in *2007 IEEE Swarm Intelligence Symposium*. Honolulu, HI, USA: IEEE, Apr. 2007, pp. 332–339. [Online]. Available: http://ieeexplore.ieee.org/document/4223193/

[17] J. Hereford, "A Distributed Particle Swarm Optimization Algorithm for Swarm Robotic Applications," in *2006 IEEE International Conference on Evolutionary Computation*. Vancouver, BC, Canada:

IEEE, 2006, pp. 1678–1685. [Online]. Available: http://ieeexplore.ieee.org/document/1688510/

[18] Y. Meng, O. Kazeem, and J. C. Muller, "A Hybrid ACO/PSO Control Algorithm for Distributed Swarm Robots," in *2007 IEEE Swarm Intelligence Symposium*. Honolulu, HI, USA: IEEE, Apr. 2007, pp. 273–280. [Online]. Available: http://ieeexplore.ieee.org/document/4223185/

[19] H. Tang, W. Sun, H. Yu, A. Lin, M. Xue, and Y. Song, "A novel hybrid algorithm based on PSO and FOA for target searching in unknown environments," *Applied Intelligence*, vol. 49, no. 7, pp. 2603–2622, July 2019. [Online]. Available: http://link.springer.com/10.1007/s10489-018-1390-0

[20] A. Cornejo, F. Kuhn, R. Ley-Wild, and N. Lynch, "Keeping Mobile Robot Swarms Connected," in *Distributed Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 5805, pp. 496–511, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-642-04355-0_50

[21] W. Luo and K. Sycara, "Minimum k-Connectivity Maintenance for Robust Multi-Robot Systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Macau, China: IEEE, Nov. 2019, pp. 7370–7377. [Online]. Available: https://ieeexplore.ieee.org/document/8968058/

[22] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987, arXiv: cond-mat/0208573 ISBN: 0897912276. [Online]. Available: http://portal.acm.org/citation.cfm?doid=37402.37406

[23] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *Proceedings - IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., 2012, pp. 3293–3298, iSSN: 10504729.

[24] K. Perlin, "An image synthesizer," *ACM SIGGRAPH Computer Graphics*, vol. 19, no. 3, pp. 287–296, July 1985. [Online]. Available: https://dl.acm.org/doi/10.1145/325165.325247

[25] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965. [Online]. Available: http://ieeexplore.ieee.org/document/5388473/

[26] F. Berlinger, M. Gauci, and R. Nagpal, "Implicit coordination for 3D underwater collective behaviors in a fish-inspired robot swarm," *Science Robotics*, vol. 6, no. 50, p. eabd8668, Jan. 2021. [Online]. Available: https://www.science.org/doi/10.1126/scirobotics.abd8668

[27] A. E. Turgut, H. Çelikkanat, F. Gökçe, and E. Şahin, "Self-organized flocking in mobile robot swarms," *Swarm Intelligence*, vol. 2, no. 2-4, pp. 97–120, Dec. 2008. [Online]. Available: http://link.springer.com/10.1007/s11721-008-0016-2

[28] G. Vasarhelyi, C. Viragh, G. Somorjai, N. Tarcai, T. Szorenyi, T. Nepusz, and T. Vicsek, "Outdoor flocking and formation flight with autonomous aerial robots," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Chicago, IL, USA: IEEE, Sept. 2014, pp. 3866–3873. [Online]. Available: http://ieeexplore.ieee.org/document/6943105/

[29] J. E. Jones, "On the determination of molecular fields.—I. From the variation of the viscosity of a gas with temperature," *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 106, no. 738, pp. 441–462, Oct. 1924. [Online]. Available: https://royalsocietypublishing.org/doi/10.1098/rspa.1924.0081

[30] J. Ebert and R. Barnes, "Kilosim," Sept. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3406865

[31] J. Ebert, "Hybrid PSO Multi-robot Target Search with Kilosim Gridbots," Feb. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.6311164

[32] B. P. Duisterhof, S. Li, J. Burgues, V. J. Reddi, and G. C. H. E. de Croon, "Sniffy Bug: A Fully Autonomous Swarm of Gas-Seeking Nano Quadcopters in Cluttered Environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Prague, Czech Republic: IEEE, Sept. 2021, pp. 9099–9106. [Online]. Available: https://ieeexplore.ieee.org/document/9636217/