VOLUME 17, N° 2 2023
Journal of Automation, Mobile Robotics and Intelligent Systems

Quantifying Swarm Resilience with Simulated Exploration of Maze-Like Environments

Submitted: 11th November 2021; accepted: 24th March 2022

Megan Emmons, Anthony A. Maciejewski

DOI: 10.14313/JAMRIS/2-2023/10

Abstract:
Artificial swarms have the potential to provide robust, efficient solutions for a broad range of applications from assisting search and rescue operations to exploring remote planets. However, many fundamental obstacles still need to be overcome to bridge the gap between theory and application. In this characterization work, we demonstrate how a human rescuer can leverage minimal local observations of emergent swarm behavior to locate a lone survivor in a maze-like environment. The simulated robots and rescuer have limited sensing and no communication capabilities to model a worst-case scenario. We then explore the impact of fundamental properties at the individual robot level on the utility of the emergent behavior to direct swarm design choices. We further demonstrate the relative robustness of the simulated robotic swarm by quantifying how reasonable probabilistic failure affects the rescue time in a complex environment. These results are compared to the theoretical performance of a single wall-following robot to further demonstrate the potential benefits of utilizing robotic swarms for rescue operations.

Keywords: *Swarm robotics*

1. Introduction

Swarm robotics is a relatively new domain of research that takes inspiration from cooperative biological systems such as flocking birds, ant colonies, and schools of fish. Like their biological counterparts, robots in a swarm are governed by local rules, but frequent interactions with other robots and the environment generate a more complex, emergent behavior. Sophisticated foraging strategies and complex colony construction by ants reveal just a few of the potential advantages of emergent behavior because these actions are accomplished in a distributed and robust manner. Similar emergent behaviors in artificial systems will be extremely useful in many exploration tasks, particularly in harsh environments with a high probability of robot failure. Disaster scenarios are an especially relevant application domain with an increasing rise in occurrence and economic impact [1] and, currently, no viable robotic solutions.

Despite the many important applications and advancements in theory, research in robotic swarms has not yet matured to the point of reliable, real-world deployment. Two fundamental hurdles between the potential and reality of swarms are (1) determining what collective behavior will emerge from the swarm and (2) identifying the influence of local parameters. These challenges are not decoupled. Each robot within the swarm may be equipped with a variety of sensors and control strategies that can be considered as parameters. Swarms are necessarily composed of a large number of agents [2, 3] so the choice of parameters is immediately scaled by the size of the swarm, which imposes some minor difficulties. The true difficulty comes from the collective behavior that emerges after the robots interact with each other as well as the environment. Robot interactions propagate aspects of the local parameters, but there is no closed-form method for determining how individual behaviors will affect the emergent behavior.

Physical implementations would best reveal the full emergent behavior. Swarm test platforms are surfacing and offer improved research development, but current realizations are restricted to table-top environments, like the kilobot [4] or Zooid [5], often require overhead vision systems for robot positioning such as in [6], and are necessarily constrained by the physical number of robots. For example, Georgia Tech's impressive Robotarium offers a remotely accessible testbed for swarm algorithms but only includes 25 physical robots [7]. Beyond environment and size limitations, the initial choices of robot physicality already constrains the range of potential emergent behaviors the swarm can exhibit [8]. Simulation therefore remains a very necessary design step because it can be used to investigate the baseline impact of local parameters on collective properties. The investigation can inform choices for the base physicality of individual robots within the swarm and is less constrained by swarm size and environmental constraints.

Arguably the most fundamental local parameters dictate how a robot moves and interacts with the environment. Specifically, a robot's speed, motion ability, and sensing range control how the robot will move through an environment even in the presence of more sophisticated trajectory control like [9]. The effect of these parameters will be amplified in a swarm and impact properties of the emergent behavior. A human observing the swarm will likely not be able to identify subtle changes in the emergent behavior, but group studies where human participants are asked to classify swarm behaviors by observing simulated results have shown that people can

2023 © Emmons and Maciejewski. This is an open access article licensed under the Creative Commons Attribution-Attribution 4.0 International (CC BY 4.0) (http://creativecommons.org/licenses/by-nc-nd/4.0/)

recognize general patterns [10–12]. Robot density, velocity, and relative cohesion were all notable properties that helped human participants classify simulated emergent behavior.

In this foundational study, we consider a human rescuer who can only rely on local observations of the emergent swarm velocity to navigate a simulated disaster environment in an attempt to locate a lone survivor. Robots within the swarm have no communication or localization ability. This model allows the influence of fundamental robot features related to motion and sensing to be identified. Variations in local swarm parameters are evaluated qualitatively, by observing the resulting motion and area coverage, as well as quantitatively, by the impact of the parameter variation on the rescuer's ability to successfully locate the lone survivor. From this work, we begin to establish important swarm characterization. The robustness of swarms in terms of parameter variation, environmental features, and robot failure is quantified. We also demonstrate the value of employing swarms in disaster scenarios to motivate additional development.

We first place our work in the grander scope of swarm robotics by presenting related work in Section 2 before presenting the simulation framework in Section 3. Section 4 presents a summary of results from our investigation of emergent behavior. We also discuss important observations from our research in Section 5 before summarizing our work in Section 6.

2. Related Work

Our work is motivated by the absence of robot solutions for disaster scenarios including mine rescue [13] and building collapse [14]. Ongoing DARPA challenges and reports [15] indicate a need for more reliable physical robot implementations. There is promising progress in this domain for single robots [16, 17] and improved path planning to mitigate risk [18]. Swarms can potentially leverage these advancements and further reliability efforts by utilizing a large number of robots for increased robustness. Experiments have shown that emergent swarm patterns can serve as an information storage mechanism [19, 20] which supports the robustness paradigm. Yet, to the best of our knowledge, no work has actually quantified the resilience of a swarm.

In our investigation, we explore a minimalist swarm to establish a baseline performance to which additional functionality can then be compared. Our approach considers the interests of potential swarm users as indictated in an important study conducted by Carrillo-Zapta et al. where target users, including firefighters, identified areas where swarms can support current activities [21]. Study participants considered swarms beneficial for gathering information and supporting communication but emphasized that an actual person should remain 'in the loop' for decision making.

Decision-making by the human can be informed by observations of the emergent swarm behavior. As demonstrated by works in swarm expressivity, human participants were able to classify swarm behavior by 'unfocusing' and instead looking at collective behaviors such as robot spacing, velocity, and relative cohesion [10–12]. We maintain our baseline approach by having our simulated human rescuer move using only the observed swarm velocity.

The results from our work focus on maintaining a minimal approach, which can represent a worst-case scenario in robot functionality but also serve as a baseline. More sophisticated algorithms can then be compared to these results for design and application purposes. For example, the potential benefits of improved odometry [22], environment analysis [23], velocity control [24], shape formation [8], and local communication for risk mitigation [25] can be evaluated.

3. Simulation Framework

3.1. Swarm Behavior Model

We model a minimalist robot moving in an unknown, two-dimensional environment. The robot is unable to communicate, has no means for localization, and relies on limited sensing. This model allows us to characterize baseline performance for the swarm but also represents a worst-case scenario for physical robot deployments.

The swarm is composed of M such robots where M is a user-defined parameter allowing the significance of swarm size to be explored. Robots are initially distributed randomly within a user-defined radius around the *start_center*. The position of robot i at iteration k is denoted as $\mathbf{x}(i,k)$. Using discrete-time steps, robots attempt to move in a straight line with general desired velocity

$$\mathbf{v}_d(k) = s_s[\cos(\alpha_k), \sin(\alpha_k)], \tag{1}$$

with s_s representing the maximum robot speed, while also striving to avoid collisions. Each robot is initialized with a uniformly distributed starting angle α_1. Subsequent trajectory angles are a combination of the previous n trajectory angles and a Gaussian noise calculated as

$$\alpha_k = \begin{cases} \frac{1}{n}\sum_{a=k-n}^{k-1} \alpha_a + \gamma & \text{if } n < k \\ \alpha_1 & \text{otherwise} \end{cases} \tag{2}$$

where γ is a normally distributed random variable centered at zero with user-defined standard deviation, σ. We explore the impact of heading control on swarm resiliency by varying σ. The size of the moving window, n, is also varied to determine the importance of *memory* on the emergent swarm behavior.

Once a desired velocity is determined, we ensure the robot does not collide with any objects in its sensing radius, r_s, by using a generalized forcing model. The actual velocity for robot i at time step k is calculated as

$$\mathbf{v}_s(i,k) = \mathbf{v}_d(i,k) + \sum_{w=1}^{W} \mathbf{f}_w(w) + \sum_{j=1, j\neq i}^{M} \mathbf{f}_s(i,j) \tag{3}$$

where $\mathbf{f}_w$ and $\mathbf{f}_s$ denote the effect of the environment and other robots on robot i, respectively. We next define the repulsive wall force for each of the W walls creating our environment as

$$\mathbf{f}_w(w) = \begin{cases} A_s e^{(r_s - d(w))/B_s}\hat{\mathbf{n}} & \text{if } d(w) \leq r_s \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{4}$$

with $\hat{\mathbf{n}}$ as a unit vector pointing into the environment, perpendicular to the wall, and $d(w)$ as the minimum distance to the wall segment w. The choice of coefficients A_s and B_s dictates how strongly robots will be repelled from obstacles. Robots in the swarm also need to avoid collisions with each other, so $\mathbf{f}_s$ is a similarly repulsive force imposed on robot i by all other robots. Specifically, we define

$$\mathbf{f}_s(i,j) = \begin{cases} C_s e^{(r_s - d(i,j))/F_s}\hat{\mathbf{k}} & \text{if } d(i,j) \leq r_s \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{5}$$

where $d(i,j)$ is now the scalar distance between robot i and robot j, calculated as

$$d(i,j) = \|\mathbf{x}(i,k) - \mathbf{x}(j,k)\|. \tag{6}$$

We maintain C_s and F_s as exploration parameters and $\hat{\mathbf{k}}$ as a unit vector in the direction of the line of impact for the collision. Here we note, (5) also serves as a dispersion model to ensure robots are generally directed further into new regions of the environment.

3.2. Rescuer Model

Maintaining our base assumptions of a disaster scenario, the human rescuer likewise has a limited visibility range and cannot modify the behavior of the robots. The rescuer can travel at a maximum speed of s_{max}, has an observation radius of r_h, and position $\mathbf{x}_h(k)$ at time step k. During time step k, the rescuer calculates their desired velocity as

$$\mathbf{v}_{d,h}(k) = \mathbf{v}_{obs}(k) + \sum_{w=1}^{W} f_{hw}(w) \tag{7}$$

where $\mathbf{v}_{obs}(k)$ is a velocity inferred from local observations of the swarm behavior. The second term in (7) prevents the rescuer from colliding with the wall, analogous to (4) but with defined scalars

$$A_h = 0.1 s_{max} \tag{8}$$

$$B_h = 2s_{max}/\log\frac{2s_{max}}{A_h}. \tag{9}$$

By the formulation of (7), the rescuer's desired velocity is primarily determined using local observations of the swarm. We implement a simple flow model for the rescuer to explore how changes in the emergent swarm behavior influence the utility of observable swarm properties. More specifically,

$$\mathbf{v}_{obs}(k) = s_{max}\frac{\mathbf{v}_e(k)}{\|\mathbf{v}_e(k)\|} \tag{10}$$

where

$$\mathbf{v}_e(k) = \sum_{i=1}^{M} \mathbf{v}_s(i,k)t(i) \tag{11}$$

and

$$t(i) = \begin{cases} 1 & \text{if } \|\mathbf{x}(i,k) - \mathbf{x}_h(k)\| \leq r_h \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

is an indicator function ensuring the rescuer only uses swarm velocities which are observable.

Finally, the rescuer cannot exceed their maximum speed, s_{max}, nor will they act if the inferred velocity from swarm observation is below some threshold magnitude, s_{min}, so the actual implemented velocity at iteration k, denoted as $\mathbf{v}_h(k)$, becomes

$$\mathbf{v}_h(k) = \begin{cases} s_{max}\frac{\mathbf{v}_{d,h}(k)}{\|\mathbf{v}_{d,h}(k)\|} & \text{if } \|\mathbf{v}_{d,h}(k)\| > s_{max} \\ \mathbf{0} & \text{if } \|\mathbf{v}_{d,h}(k)\| < s_{min} \\ \mathbf{v}_{d,h}(k) & \text{otherwise} \end{cases} \tag{13}$$

with an additional limit of $\pi/4$ on the maximum angle change the rescuer will experience between time steps unless they are near a wall boundary.

Equation (11) leverages the pattern recognition ability of people as demonstrated in studies like those by Walker et al. [10]. It also introduces an important but challenging aspect of autonomous exploration, which is determining the 'optimal' amount of time to wait for information before acting.

Swarm interactions encode important environmental information, like the presence of openings or obstacles [20]. The encoding process changes properties of the emergent behavior but takes an amount of time depending on the robot density and distance to obstacles. We introduced two simple parameters to explore when the rescuer should begin moving in the environment using local observations of the swarm. The s_{min} parameter specifies a minimum magnitude of observed velocity the rescuer must observe before acting and represents a coarse consensus within the swarm about a desired direction. The second parameter, PT, is a specified pause time wherein the rescuer allows the robots time to interact before observing the emergent velocity.

3.3. Environment Description

Our goal in this work is to initiate a baseline quantification of swarm parameters, so we defined a starting environment, explored the simulated performance, then systematically added features to gain insight into how different parameters influenced the emergent behavior. Figure 1 presents the four main environments discussed in this work. All of the environments are built around a central, square room that is four units wide. The size of the environment serves as a scaling factor that directs the choice of other exploratory parameters like maximum speed and sensor radius. The simulated robot swarm and rescuer are initially positioned centrally in this room and cannot pass through environment boundaries. Robots continue to move about the environment according to

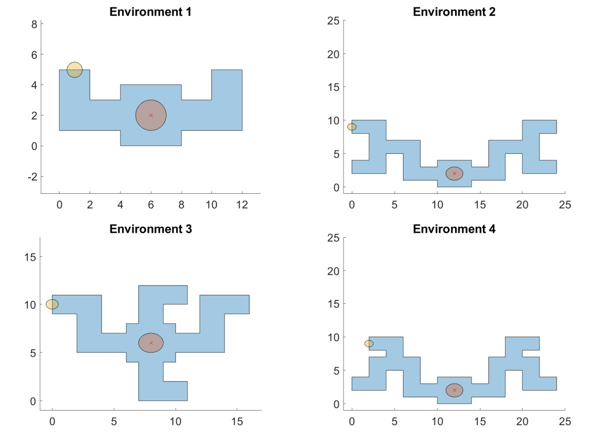

Figure 1. A lone survivor is placed at the end of the top-left-most hallway for each of the four simulated disaster environments, denoted by the yellow circle. The rescuer begins in the center of the environment but has limited sensing, as indicated by the red ellipse

(3) unless they experience a failure or are within a sensor radius of the survivor's location, at which point the robot is removed from the simulation. The robots themselves are not attempting to locate the survivor, but the collective flow created by having robots no longer explore once they reach the survivor directs the rescuer toward the survivor, as the remaining work shows. Removing robots from the simulation can be physically recreated by having the survivor turn the robots off or using another equivalent strategy.

4. Results

4.1. Preliminary Parameter Characterization

For the initial swarm parameter evaluation, we focused on qualitatively assessing the reasonableness of the emergent behavior by animating the swarm exploration process and quantitatively recording the percentage of area explored within a fixed time. We initially simulated the robots in Environment 1. As indicated in Figure 1, the survivor is placed at the end of the west hallway. Robots stop when they are in the vicinity of the survivor. After identifying a reasonable initial operating range for all parameters, we proceeded to systematically vary one parameter at a time and evaluate the resulting swarm performance. The experiments confirmed several intuitive correlations:

- Increasing the number of robots increased the area explored, but with diminishing rate of return, so that a very large number of robots was needed to generate even a small increase in exploration area
- Increased robot speed similarly increased area explored, but again with diminishing returns
  - Too large a speed resulted in unnatural motion, disproportionate to environment size
- The number of past movements stored had a negligible effect on the area explored when there were robot interactions
- More area was explored when the robots' motion was less random
  - To represent realistic motion, a small value of σ should be incorporated into the robots' motion
  - The presence of other robots acted to restrict sporadic motion due to collision avoidance
- Larger sensor radius generally improved performance

From this preliminary set of experiments, we defined base values for the swarm parameters, summarized in Table 1, that created a relatively fluid and efficient dispersion in the environment.

Table 1. Summary of simulation parameters

Param.	Significance	Base Value
M	Number of robots in swarm	300
s_s	Maximum robot speed	0.4
n	Number of stored movements	1
σ	Std. dev. for Gaussian noise	0.1
r_s	Robot's sensing radius	0.8
A_s, B_s	Force coeff. for obstacles	1, 0.5
C_s, F_s	Force coeff. for neighbors	1, 0.5

4.2. Resilience to Rescuer Variation

Using the baseline swarm parameter values summarized in Table 1, we ran 100 different swarm dispersion scenarios for the environments in Figure 1. A maximum exploration time of 3000 steps was imposed on all scenarios to ensure the simulation time remained tractable. We then simulated a rescuer who navigates the unknown environment according to (13) in an effort to reach the lone survivor. Each rescuer parameter – threshold speed (s_{min}), pause time (PT), sensor radius (r_h), and maximum speed (s_{max}) – was systematically varied to explore the resiliency of the swarm and resulting rescue strategy with respect to variations in the fundamental rescuer behavior. Our priorities in evaluating the rescuer performance align with real-world disaster scenarios: ensuring the rescuer successfully locates the survivor as reliably as possible, minimizing the danger to the rescuer by reducing the number of steps they take in a potentially dangerous environment, and reducing the time it takes for the rescuer to locate the survivor.

Systematically varying each rescuer parameter revealed fundamental relationships between the properties of a successful rescue and the local emergent behavior. Variation in s_{min} had negligible impact on our evaluation criteria, but the rescuer often required fewer steps to locate the lone survivor when the PT was first increased. The median rescue times for each environment and a range of PTs are plotted in Figure 2 along with a fitted curve to help visualize the overall trend. Pause times around 300 were generally most beneficial for reducing the number of steps the rescuer needed to locate the survivor, independent of environment. Further increases to the PT did not notably reduce the number of steps and instead only increased the rescue time.

Decreasing the number of rescuer steps is partially correlated with reduced danger to the rescuer, an important priority in disaster scenarios, but waiting

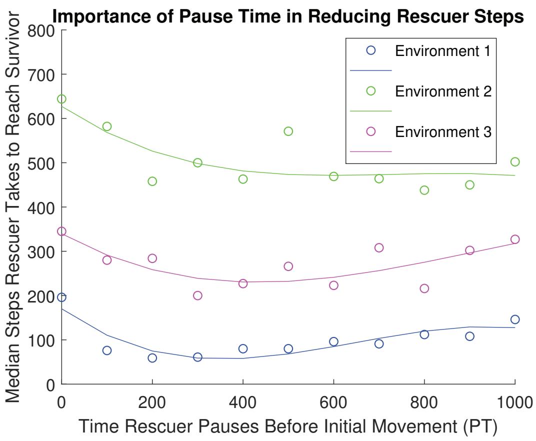

Figure 2. The rescuer requires fewer steps to locate the lone survivor if they first pause and allow the robots to do some preliminary exploration. The median number of steps taken by the rescuer over 100 simulations shows that a PT of 300 was generally the most beneficial, but there is a range of comparable values indicating resilience to rescuer PT

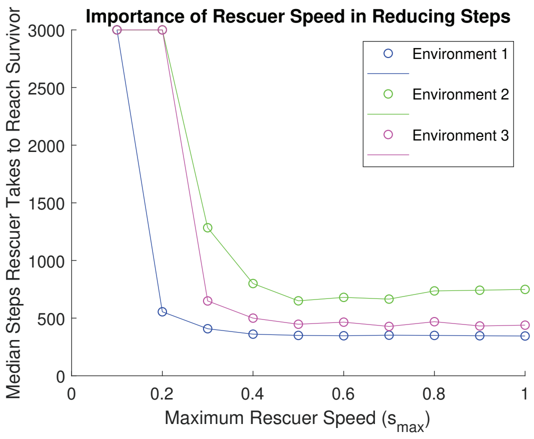

Figure 3. The rescuer often requires the fewest number of steps to locate the survivor when they have the same maximum speed as the robots. Here the median rescuer step count is shown when robots have a maximum speed of 0.4. Travelling faster than the swarm does not offer improved performance but indicates a resiliency to variations in rescuer speed

for the robots to do preliminary exploration also adds time to the overall rescue, which is undesirable. The trade-off between acting early or waiting for more information is not new, but disaster scenarios add the challenge of a potentially dynamic environment where uncertainty is never fully resolved. Fortunately, rescuers are highly trained to make decisions based on the specific situation, so the results of Figure 2 can be used to primarily inform real-world rescue strategies. The informative role of the swarm exploration also aligns with the preferred swarm interaction expressed by participants in the survey by Carrillo-Zapata et al. [21].

Speed is a similarly important parameter to ensure the rescuer fully leverages information from the swarm. Figure 3 shows the median time required to locate the survivor as a function of rescuer speed for the first three environments from Figure 1. The survivor is located a specific distance away from the center of the environment where the rescuer is initially positioned so, theoretically, a faster speed would result in decreasing the lower bound on rescue time. We see that the benefits of travelling faster are initially present in Figure 3 but speeds faster than the swarm speed of 0.4 do not significantly improve performance in any of the environments.

While the utility of the swarm was resilient to variations in rescuer parameters, the rescuer generally required fewer steps to locate the survivor if they implemented a sufficient pause time (PT) and a reasonable speed (s_{max}). Another important parameter is the number of robots present in the swarm to ensure an informative behavior emerges. Figure 4 demonstrates the effect of increasing the number of robots in Environment 4 on the average rescue time. A sufficient number of robots are clearly needed – the rescuer never located the survivor when only one robot was present, and only found the survivor in

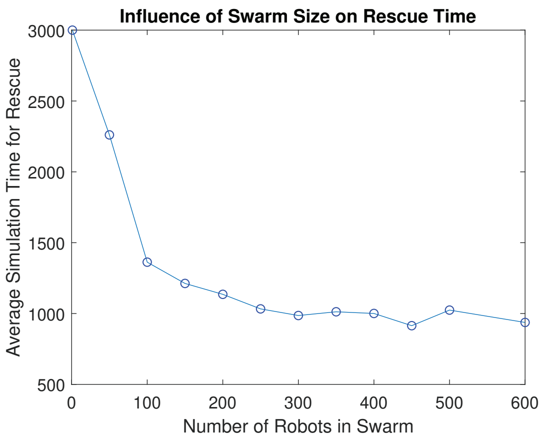

Figure 4. The rescuer required less time on average once more than 300 robots were initially distributed because a sufficient number of robot interactions occurred for informative behaviors to emerge. Using more than 300 robots did not significantly reduce the rescue time, as demonstrated by the average rescue times for Environment 4

47% of the simulations when relying on 50 robots, which resulted in an average of 2400 time steps – but *sufficient* is difficult to define in advance, particularly when the environment itself is unknown, as is the case for most disaster scenarios. Fortunately, the rescue performance for our minimalist scenario is relatively robust with respect to variations in swarm size. Swarm sizes of 200 – 600 robots all resulted in very similar average rescue times as summarized in Figure 4 for Environment 4.

4.3. Resilience to Environment Hazards

The harsh terrain of many disaster scenarios results in a high probability of failure for autonomous systems. We have shown that the utility of the emergent behavior is robust with respect to initial swarm

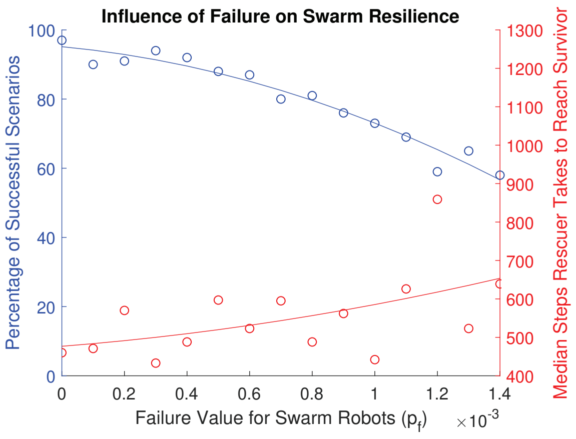

Figure 5. The median rescue time in Environment 4 (lower red curve) increased as robots experienced larger failure rates, but the rescuer still successfully located the survivor in the majority of scenarios (top blue curve) even when as few as 12 robots remained functional

size but, alternatively, these results can be extended to scenarios where a large number of robots fail during the exploration task. To more fully characterize the impact of robot failure on our hypothetical disaster rescue scenario, we introduce a failure parameter, p_f, to our simulation model. Prior to moving, a robot will be assigned a uniformly generated random number R. The robot is removed from the simulation if $R \leq p_f$ to simulate a realistic, immobilizing failure.

The lower curve in Figure 5 shows the median time required to locate the survivor in Environment 4 as p_f was systematically increased. The top curve indicates how many of the 100 scenarios resulted in the rescuer successfully locating the survivor. For all 100 scenarios, 300 robots are initially dispersed, travelling at 0.4 units per time step, with a rescuer PT of 300 time steps.

It is instructive to compare the performance of the swarm in the presence of catastrophic failure from Figure 5 to the results for varying the initial swarm size in Figure 4. A p_f of 0.0014 resulted in the rescuer taking a median of 639 steps. The swarm experienced a 75% failure rate over the course of the simulation, so 225 of the original 300 robots failed by the time the rescuer successfully located the survivor. Although there were only 75 robots operating in the environment near the end of the simulation, the surviving robot behaviors had still been influenced by interactions with other robots pre-failure. Information about the environment was distributed through the swarm by these interactions so that, even with mass failure, the rescuer could still leverage the swarm knowledge to locate the survivor. Interpolating the values from Figure 4, the median time for the rescuer would have been about 2000 steps if the swarm was initially composed of only 75 robots and there were no failures. Amazingly, the rescuer successfully reached the survivor as long as at least 12 of the original 300 robots were still present in the environment, which occurred in 58 of the 100 scenarios.

To further put the swarm resilience into context, we consider the theoretical performance of a single robot that primarily operates using a wall-following strategy. Environment 4 has a perimeter of 104 units, so a wall-following robot would have an expected path length of 52 units, thus enforcing the same speed, we expect the wall-follower to need 130 steps on average to locate the lone survivor. The wall-following robot is also a singular entity, so the rescuer will only successfully locate the survivor if the robot itself does not fail, but each move in a harsh environment adds a chance for failure. We model the probability, p_n, of the wall-following robot successfully moving n steps by using the exponential decay model

$$p_n = e^{-cn} \tag{14}$$

where c is the probability of failure per unit step. Evaluating (14) with $n = 130$, the wall-following robot would theoretically need to achieve a 99.58% reliability per step to locate the lone survivor in Environment 4 in as many trials as our minimalist swarm, where 75% of the robots failed and the survivor was still successfully located.

Even with 100% step reliability, a wall-following strategy may simply be ineffective for the particular environment of interest due to a physical lack of boundaries or extremely reduced visibility, which even limits the abilities of human rescuers [21]. We further explore the swarm's resilience to environmental variations by adding a room, or cavern, to one end of Environment 4 and placing the lone survivor at the center of this room. The room dimensions are such that neither the robots nor the rescuer can sense the survivor without losing contact with a wall. Figure 6 shows the new environment as well as the resulting rescuer trajectory for one illustrative simulation. The trajectory shows that the rescuer sometimes circles back on their previous path, which adds steps to the total that would not occur with a wall-following algorithm; however, the trajectory also shows that the rescuer avoided entering several unoccupied regions of the environment, instead spending extra time in areas that must necessarily be entered in order to reach the survivor.

Our minimalist rescuer model only considers the velocities of robots within the rescuer's sensing radius, but this simple algorithm leverages the emergent swarm behavior to reduce the likelihood of entering unnecessary regions of the environment. Consider the rescuer's decision making process when they approach the final hallway junction before entering the large room where the survivor is located. Figure 7 zooms in on this region of the environment and shows the rescuer's position as a red 'x' with the sensing area illustrated with the purple circle. At this point, the rescuer has been unable to detect the upper hallway. A wall-following strategy would lead the rescuer into the dead-end hallway, but several robots have already explored this region and are moving back up, creating a small repulsive wave that directs other robots as well as the rescuer away from the dead end in general.

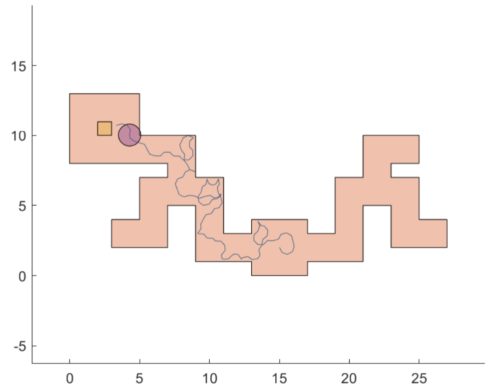

Figure 6. In the cavern environment, the survivor (yellow square) is positioned in the middle of a large room where limited sensing prevents both the rescuer and robots from seeing the survivor while maintaining contact with the wall. The survivor would therefore not be successfully located if the rescuer was alone, or reliant on a wall-following robot with comparable sensing radius, but the dispersion of robots in the swarm directs the rescuer away from the wall and to the survivor. The rescuer also avoids entering unnecessary regions of the environment in this scenario

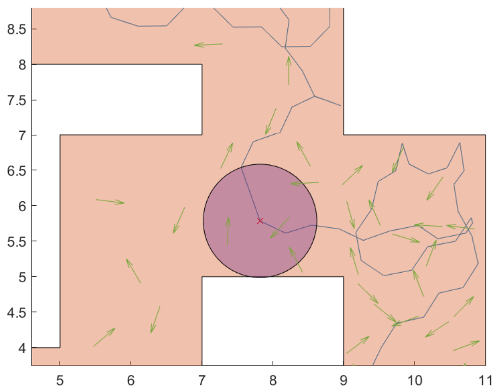

Figure 7. Some robots have already entered the left-most branch of the cavern environment and encountered a dead end. Zooming in on the final junction of the cavern environment and showing the robot velocities as green arrows, the returning robots exert a pressure that helps direct the rescuer, who can only detect objects located within the large purple circle, into the correct hallway and away from the unnecessary environment segment

The velocity for each robot is shown as a green arrow. The repulsive force of the wall, combined with the observable swarm motion at the single time step, direct the rescuer away from a potentially hazardous region of the environment and up into the correct hallway.

More sophisticated path-planning algorithms would reduce the amount of rescuer back-tracking, but our focus in this work was on quantifying the resiliency of the emergent swarm behavior to perturbations. Even with the minimalist model used, the rescuer successfully located the lone survivor in our cavern environment in 83 out of 100 simulations and required a median of 676 steps. The rescuer also frequently avoided entering unnecessary regions of the environment, thereby further reducing the potential hazards encountered in real disaster scenarios. No change in the swarm behavior was needed to account for a significant change in the environment, which further illustrates the swarm's resiliency, this time with respect to environmental features.

5. Discussion

While we present specific parameter values in Section 4, we are by no means proposing these as ideal scalars for swarm exploration. Nor are we advocating a simplistic interaction between swarm and simulated rescuer. The real contribution of this work is establishing a baseline quantification of swarm resilience, demonstrating important relationships between robots and a minimalist human rescuer, and illustrating the feasible benefits of leveraging emergent properties for important applications like locating survivors in disaster environments.

Interactions are the primary force driving emergent swarm behavior, so a sufficient number of robots are necessary to ensure that collective properties evolve. While this statement is partially intuitive and also supported by Figure 4, the influence of swarm size on individual robot properties is potentially less clear but an extremely important design consideration. We used values from Table 1 to govern the dispersion algorithm. The choice of coefficients was impressively resilient to variations, but testing minimum swarm sizes highlights the value of interactions for producing more complex behaviors.

For example, during our preliminary investigations, we found that an individual robot did not need to store past steps or implement sophisticated heading control for efficient area coverage if other robots were in the sensing area. Neighboring robots in the swarm essentially serve as a motion memory and enforce more direct trajectories. When one robot moved, other robots could fill in the space to prevent the robot from undoing its move, which is the traditional role of 'memory'. Similarly, an individual robot may have very random motion (large value of σ in our simulation) but the presence of other robots combined with a simple collision avoidance mechanism reduced the robot's erratic trajectory compared to the robot's trajectory in isolation. The collective behavior generally resulted in more efficient local behaviors without any change to the individual robot parameters. Figure 8 qualitatively demonstrates the improved trajectory for a robot in a swarm with $M = 300$ robots (red line) compared to a robot operating on its own (blue line) with the same values from Table 1.

Robot interactions also contribute to the desired robustness frequently associated with swarms

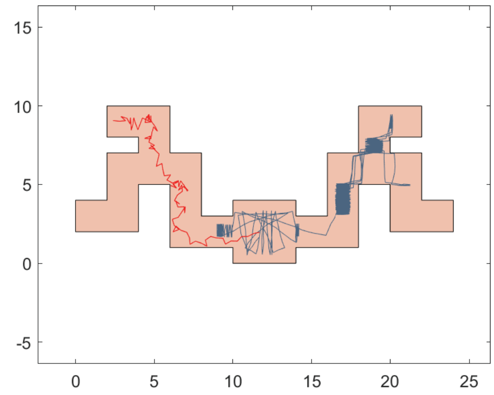

Figure 8. A robot operating with the parameters of Table 1 with swarm size $M = 1$ ineffectively navigates the environment (blue line) but increasing the swarm size improves exploration trajectories without any change to the local rules. The red line is an illustrative robot trajectory when $M = 300$

because information is stored implicitly in the behaviors rather than explicitly with any single robot. Figure 5 demonstrates that the swarm could experience catastrophic failure but still retain sufficient information for the rescuer to leverage and successfully locate the lone survivor. Of the 58 scenarios with $p_f = 0.0014$ where the rescuer successfully reached the survivor, the fewest number of robots still functioning in the environment was just 12, but the motion of those 12 robots had benefitted from the initial distribution of 300 robots.

The surviving robots were generally guided along a more direct path to the survivor by other robots that had entered dead end hallways and were attempting to return before failing. The rescuer similarly benefitted from the exploration of the failed robots before they were immobilized. One challenge from our implementation is that the rescuer *required* a minimum observed velocity from the swarm before moving. Thus, in the event that all robots within the rescuer's sensing radius failed, the rescuer was unable to move. In a real-world disaster scenario, the swarm would most likely only inform the rescuer's search strategy so that even in the extreme cases where no robots are in the rescuer's sensing radius, the rescuer can still navigate the environment.

6. Conclusion

The distributed nature of swarms offers an intuitive promise of robustness and efficiency that is especially desirable for exploration-based tasks in harsh environments like locating disaster survivors. In this work, we establish a baseline quantification of swarm robustness by first evaluating the impact of fundamental robot properties on the utility of the emergent behavior. A simulated human rescuer relies on the locally observable swarm velocity to navigate an unknown disaster environment while attempting to locate a lone survivor. We affirmed that a minimalist swarm governed by a simple collision avoidance algorithm had sufficiently complex interactions to encode important environmental information that the rescuer successfully leveraged. The utility of the swarm was robust with respect to variations in fundamental rescuer parameters and changes to the environmental structure. More impressively, the rescuer still found the survivor when the swarm underwent catastrophic failure and lost up to 288 of the original 300 robots.

Although simple, the robust success of a simulated rescuer locating a lone survivor affirms the benefits of robot swarms. Local observations of the emergent swarm behavior often reduced the number of regions the rescuer entered and even consistently directed the rescuer into empty spaces that would be inaccessible if relying on limited-sensing strategies like wall-following. Properties of the swarm do not need to be optimized for the swarm to still be effective in a variety of environments. While our simulation established an important baseline, we believe more information can be extracted by a human observing the emergent behavior and the additional information will further improve the swarm performance in all interactive applications.

AUTHORS

Megan Emmons* – Colorado State University, Fort Collins, CO 80523, USA, e-mail: mremmon@rams.colostate.edu.

Anthony A. Maciejewski – Colorado State University, Fort Collins, CO 80523, USA, e-mail: aam@colostate.edu.

*Corresponding author

References

[1] M. Coronese, F. Lamperti, K. Keller, F. Chiaromonte, and A. Roventini. "Evidence for sharp increase in the economic damages of extreme natural disasters," in *Proceedings of the National Academy of Sciences*, vol. 116, no. 43, 2019, pp. 21 450–21 455.

[2] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo. "Swarm robotics: A review from the swarm engineering perspective," in *Swarm Intelligence*, vol. 7, no. 1, 2013, pp. 1–41.

[3] E. Şahin. "Swarm robotics: From sources of inspiration to domains of application," in *International Workshop on Swarm Robotics*. Springer, 2004, pp. 10–20.

[4] G. Valentini, A. Antoun, M. Trabattoni, B. Wiandt, Y. Tamura, E. Hacquard, V. Trianni, and M. Dorigo. "Kilogrid: A novel experimental environment for the kilobot robot," in *Swarm Intelligence*, vol. 12, 2018, pp. 245–266.

[5] M. L. Goc, L. H. Kim, A. Parsaei, J.-D. Fekete, P. Dragicevic, and S. Follmer. "Zooids: Building blocks for swarm user interfaces," in *Proceedings of the 29th Annual Symposium on User Interface*

Software and Technology. Association for Computing Machinery, 2016, p. 97–109.

[6] J. Wiech and Z. Hendzel. "Overhead vision system for testing swarms and groups of wheeled robots," in *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 13, no. 2, Jul. 2019, pp. 15–22.

[7] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt. "The robotarium: A remotely accessible swarm robotics research testbed," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 1699–1706.

[8] I. Buckley and M. Egerstedt. "Infinitesimal shape-similarity for characterization and control of bearing-only multirobot formations," in *IEEE Transactions on Robotics*, 2021, pp. 1–15.

[9] R. Piotrowski, B. Maciąg, W. Makohoń, and K. Milewski. "Design of control algorithms for mobile robots in an environment with static and dynamic obstacles," in *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 13, no. 4, July 2019, pp. 22–30. [Online]. Available: https://www.jamris.org/index.php/JAMRIS/article/view/525

[10] P. Walker, M. Lewis, and K. Sycara. "Characterizing human perception of emergent swarm behaviors," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC) 2016*, 2016, pp. 002 436–002 441.

[11] D. St-Onge, F. Levillain, E. Zibetti, and G. Beltrame. "Collective expression: How robotic swarms convey information with group motion," in *Paladyn, Journal of Behavioral Robotics*, vol. 10, no. 1, 2019, pp. 418–435.

[12] M. Santos and M. Egerstedt. "From motions to emotions: Can the fundamental emotions be expressed in a robot swarm," *International Journal of Social Robotics*, 2020.

[13] R. R. Murphy, J. Kravitz, S. L. Stover, and R. Shoureshi. "Mobile robots in mine rescue and recovery," in *IEEE Robotics Automation Magazine*, vol. 16, no. 2, 2009, pp. 91–103.

[14] E. Ackerman. "Why robots can't be counted on to find survivors in the florida building collapse," Jul 2021. [Online]. Available: https://spectrum.ieee.org/why-robots-cant-help-find-survivors-in-the-florida-building-collapse

[15] J. Carlson, R. R. Murphy, and A. Nelson. "Follow-up analysis of mobile robot failures," in *IEEE International Conference on Robotics and Automation*, vol. 5, 2004, pp. 4987–4994.

[16] A. Bouman, M. F. Ginting, N. Alatur, M. Palieri, D. D. Fan, T. Touma, T. Pailevanian, S.-K. Kim, K. Otsu, J. Burdick, and A. Akbar Agha-Mohammadi. "Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 2518–2525.

[17] E. Takane, K. Tadakuma, M. Watanabe, M. Konyo, and S. Tadokoro. "Design and control method of a planar omnidirectional crawler mechanism," *Journal of Mechanical Design*, vol. 144, no. 1, July 2021.

[18] X. Xiao, J. Dufek, and R. R. Murphy. "Robot risk-awareness by formal risk reasoning and planning," in *IEEE Robotics and Automation Letters*, vol. 5, no. 2, 2020, pp. 2856–2863.

[19] M. T. Jack, S. Khuman, and K. Owa. "Spatio-temporal patterns act as computational mechanisms governing emergent behavior in robotic swarms," *International Journal of Swarm Intelligence and Evolutionary Computation*, vol. 8, no. 1, 2019.

[20] M. Emmons, A. A. Maciejewski, C. Anderson, and E. K. P. Chong. "Classifying environmental features from local observations of emergent swarm behavior," in *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 3, 2020, pp. 674–682.

[21] D. Carrillo-Zapata, E. Milner, J. Hird, G. Tzoumas, P. J. Vardanega, M. Sooriyabandara, M. Giuliani, A. F. T. Winfield, and S. Hauert. "Mutual shaping in swarm robotics: User studies in fire and rescue, storage organization, and bridge inspection," in *Frontiers in Robotics and AI*, vol. 7, no. 53, 2020.

[22] M. Palieri, B. Morrell, A. Thakur, K. Ebadi, J. Nash, A. Chatterjee, C. Kanellakis, L. Carlone, C. Guaragnella, and A. akbar Agha-mohammadi. "Locus: A multi-sensor lidar-centric solution for high-precision odometry and 3d mapping in real-time," in *IEEE Robotics and Automation Letters*, vol. 6, no. 2, 2021, pp. 421–428.

[23] N. I. Giannoccaro and T. Nishida. "Analysis of the surrounding environment using an innovative algorithm based on lidar data on a modular mobile robot," in *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 14, no. 4, March 2021, pp. 25–34. [Online]. Available: https://www.jamris.org/index.php/JAMRIS/article/view/574

[24] S. A. Kumar, J. Vanualailai, B. Sharma, and A. Prasad. "Velocity controllers for a swarm of unmanned aerial vehicles," *Journal of Industrial Information Integration*, vol. 22, p. 100198, 2021.

[25] E. R. Hunt, C. B. Cullen, and S. Hauert. "Value at Risk Strategies for Robot Swarms in Hazardous Environments," in *Unmanned Systems Technology XXIII*, H. G. Nguyen, P. L. Muench, and B. K. Skibba, Eds., vol. 11758, International Society for Optics and Photonics. SPIE, 2021, pp. 158–177. [Online]. Available: https://doi.org/10.1117/12.2585760