Hunt, E. R., Jenkinson, G. P., Wilsher, M. J. G., Dettmann, C. P., & Hauert, S. (2020). SPIDER: a Bioinspired Swarm Algorithm for Adaptive Risk-Taking. In *ALIFE 2020: The 2020 Conference on Artificial Life* (Vol. 32). Massachusetts Institute of Technology (MIT) Press. https://doi.org/10.1162/isal_a_00279

SPIDER: a Bioinspired Swarm Algorithm for Adaptive Risk-Taking

Edmund R. Hunt[1*], George Jenkinson[2*], Michael Wilsher[3], Carl P. Dettmann[3] and Sabine Hauert[1]

[1]School of Computer Science, Electrical and Electronic Engineering, and Engineering Maths, University of Bristol, UK
[2]School of Civil, Aerospace and Mechanical Engineering, University of Bristol, UK
[3]School of Mathematics, University of Bristol, UK
*contributed equally. edmund.hunt@bristol.ac.uk

Abstract

Robot swarms can solve tasks that are impossible or too hazardous for single robots. For example, following a nuclear radiation leak, a user may wish to establish a distributed communication chain that partly extends into the most dangerous areas to gather new information. The challenge is to create long chains while maintaining chain connectivity ('connected reach'), where those at the distant end of the chain are more likely to be disconnected. Here we take the concept of dynamic 'boldness' levels from animal behavior (*Stegodyphus* social spiders) to explore such risky environments in a way that adapts to the size of the group. Boldness is implemented as a continuous variable associated with the risk appetite of individuals to explore regions more distant from a central base. We present a decentralized mechanism for robots, based on the frequency of their social interactions, to adaptively take on 'bold' and 'shy' behaviors. Using this new bioinspired algorithm, which we call SPIDER, swarms are shown to adapt rapidly to the loss of bold individuals by regenerating a suitable shy–bold distribution, with fewer bolder individuals in smaller groups. This allows them to dynamically trade-off the benefits and costs of long chains (information retrieval versus loss of robots) and demonstrates the particular advantage of this approach in hazardous or adversarial environments.

Background

In biology there is increasing awareness of the ecological and evolutionary significance of individual differences in behavior within groups; such differences include that owing to 'personality' variation (Dall et al., 2004). The social spider *Stegodyphus dumicola* lives in shared nests and engages in collection predation; group living helps their survival (Bilde et al., 2007). Recent research found that social interactions between group members shape individual 'personality' types (shy or bold) such that colonies can regenerate a suitable distribution of behaviors within the group following a perturbation (Hunt et al., 2018). Essentially, the density of the group provides a proxy for how much risk the group can afford to take – how many bold individuals it ought to sustain. This is an example of phenotypic plasticity which could be instructive for minimal robot swarms (Hunt, 2020). Here, we take bio-inspiration from *S. dumicola* to develop swarm robot controllers that can achieve a decentralized assignment of individual-level risk appetite that is appropriate to swarm-level capacity to sustain potential losses. We present a new algorithm we call SPIDER (**S***warm* **P***ersonal***I***ty for* **DE***nsity* **R***esponse*), for allocating risk appetite – propensity to explore – to a swarm of agents in a hazardous environment. This responds to a risk–return trade-off, whereby robots on the periphery of a swarm are both more likely to gain new environmental information, while also being more likely to become disconnected from the swarm or encounter hazards.

Engineering a robot swarm with the ability to dynamically set a risk appetite (boldness) role distribution that is adaptive to the task at hand has real-world applications, especially for difficult and dangerous scenarios (Schranz et al., 2020). These include the establishment of ad-hoc networks in, or exploration of, unknown, unpredictable, dangerous areas. Examples include following a natural disaster like an earthquake or an anthropogenic disaster such as a nuclear radiation leak.

In our scenario a swarm begins with a deployment around a central 'base' area. The swarm task is to establish connected chains of robots (with only very limited communication range) into more radially distant areas. With very simple robots that have severely limited navigation abilities, as robots become more radially distant they are more likely to become isolated from the swarm, or even lost altogether. Chain formation is a common mechanism in swarm robotics for searching the environment and facilitating navigation (Garattoni and Birattari, 2018; Nouyan and Dorigo, 2006; Sperati et al., 2011).

A homogeneous swarm may be able to perform its tasks more effectively if its members have a division of labor into different work roles. Pre-allocating roles to a proportion of the swarm as suitable to the given task can be an appropriate approach (Yun and Rus, 2014; Krieger and Billeter, 2000). However, this compromises the ability of the swarm to generalise to unexpected environments, or to adapt to robot failure or malfunction. Thus, methods by which decentralized agents can coordinate desired emergent distributions of roles adaptive to the task at hand are desirable. Here, our algorithm adapts boldness dynamically to achieve an allocation of near or distant 'place roles' in an emergent communication chain.

Exchange of information within a swarm relies on individuals maintaining regular, if not constant, contact with the bulk of the swarm; minimising the time that agents spend disconnected from the bulk of the swarm ensures a high level of swarm cohesion (Hauert et al., 2008, 2009). Maintaining connectivity is usually achieved by attaching a cost to becoming disconnected: varying this cost over swarm agents stratifies the swarm with individuals who occupy different functional niches or roles. The cost can be adapted online to reflect the depletion of energy or battery levels (Bénichou and Redner, 2014; Li et al., 2019). All of this poses a crucial trade-off for exploratory swarms: new areas are best discovered by robots that actively seek areas they have not covered yet. This may however lead to them losing communication with the swarm. Exploration–connectivity trade-offs are a common challenge for swarm robotics (Hauert et al., 2014).

We develop a decentralized swarm algorithm that approaches the exploration–connectivity trade-off in a novel way, using local swarm density as a proxy for the capacity of the swarm to engage in risky behavior. Our approach may be particularly suited to hazardous environments where agents are removed either temporarily or permanently.

Methods

Robots and experimental arena

The Kilobot (Rubenstein et al., 2012) has become a popular swarm robotics research platform that allows conceptual demonstration of swarm algorithms in very large numbers of robots. It is relatively small, with a diameter of 33mm and height of 34mm, and has two vibrating motors instead of wheels. It has a communication range of up to 10cm, and very limited sensing and localisation capabilities. Therefore, for the purposes of facilitating radial navigation the arena is marked with a series of concentric rings the width of a Kilobot that the robots can detect as being labelled 1, 2, or 3. This provides a gradient to count up or down to a desired travel distance, which ranges in our experiments from 0 to 26 rings. In real-world environments, an environmental gradient could be provided by received signal strength intensity (RSSI) from a base station radio beacon or other robots (e.g. McGuire et al. (2019)).

Robots are initialised in a ring around a central 'base' robot and move radially inwards or outwards depending on their boldness.

Kilobox (Jones et al., 2018) is a 2-Dimensional simulator adapted from Box2D written in C++ demonstrated to give accurate and fast simulations of Kilobots (Rubenstein et al., 2012). All experimental data presented in this study has been obtained through Kilobox simulations.

Experimental trials were all simulated for the equivalent of 5000s. This provides ample time for the swarm to reach a steady state, and is a realistic amount of time for an experiment on real Kilobots with respect to battery life.

Task and performance metric

Main Task The task for the swarm is to arrange connected chains of robots from a central 'base' area into more radially distant positions in the arena. It does this by arriving at a stable distribution of boldness-related roles that balances the need to occupy the 'safe' central region with the opportunity to cover more distant regions at and beyond the periphery of the swarm. It is important to manage the risk that peripheral robots lose contact with the swarm for extended periods of time, which would impair swarm cohesion. To manage this risk, the dynamic, self-organized swarm boldness distribution is associated with a spatial swarm distribution over a circular arena. The key principle is that a larger swarm can afford to be more risky in how it allocates boldness roles, because the disconnection of one robot (temporarily or permanently) is on balance less likely to compromise the swarm's connected reach.

Performance metric: maximum 'connected reach' We assign a reward to connected chains of robots reaching into distant regions, which is inherently penalised (reduced) if robots become disconnected from the swarm. To assess the performance of the task we measure the maximum distance reading received at the swarm center from peripheral robots. Each member of the swarm broadcasts its current distance from the central circle as a message towards the center of the swarm, which percolates inwards via connected robots. Every 0.5s, the robots reset their broadcast message to convey their current distance, and overwrite this if they receive a message from another robot that conveys a larger distance, such that they act as a signal repeater for any robot that is further out. In a given time frame, the base robot at the center of the swarm records the distance of the message which has arrived from the furthest point to the center (Figure 1). This simple performance metric can be recorded on-board the 'base' Kilobot for experimental analysis, though it is not required for the controller.

Shy, medium and bold behaviors For the purposes of our analysis, it is useful to stratify swarm members into 3 categories: shy, medium, and bold, as in Hunt et al. (2018). The three behaviors can be differentiated as follows: shy behavior manifests as remaining within a region that has a very low chance of becoming disconnected, and the geometry of the arena and the dynamics of the boldness mean that in general this is in the central region. Bold robots venture beyond the periphery of the group, into regions that are unlikely to be connected, but that result in a higher incremental performance score (successful transmission of a distant message back to base) if they are connected at any time. Medium boldness robots straddle these two behaviors, often

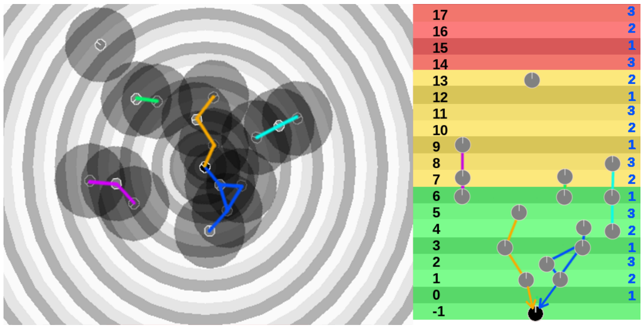

Figure 1: Swarm attempting to optimize ‘connected reach’ in 2D arena (left) and network graph representation (right) of this example swarm layout. The green, yellow and red regions are indicative of the risk of disconnection from the swarm for a given swarm size. Coloured lines show different connected clusters of Kilobots. The blue numbers on the right hand side show the gradient available to the Kilobots’ sensors as they move between the greyscale rings, which they use to track their position. The ‘base’ Kilobot (black) increments the performance metric (furthest connected reach) every 0.5s for the duration of the experiment, which in this instance would add 5 to the score total.

contributing a moderately high score increment themselves and also enabling bolder robots to score occasionally by providing a semi-stable link to the central cluster.

In larger swarms robots are likely more to be connected to the central region even at larger radial distances. This is because there tend to be higher densities in the central region, and hence the swarm would benefit from shifting its boldness distribution to include more bolder robots. In smaller swarms, even at a moderate distance a robot has a risk of being disconnected from the swarm. Therefore, a larger swarm is expected to be able to accommodate more bold individuals because this poses little risk to its connected reach, whereas a small swarm will require a higher proportion of shy individuals in order to remain connected with the base station at all.

A basic boldness mechanism: SPIDER-density

The boldness-based movement behavior of the SPIDER algorithm is described in pseudo-code in Algorithm 1. We describe two variants of the algorithm, a simple, density-based version (SPIDER-density) and a refined version of the algorithm that includes sensitivity to neighbor boldness (SPIDER-bold).

The desired radial distance from the center r_d relates to an individual robot’s boldness, which ranges from 0 to 255: dividing by 10 and rounding gives the distance in rings. For example, a bolder robot with boldness 150 will have a travel distance of 15 rings from the center, whereas a shy robot of boldness 53 will seek a distance of 5. The current distance is r_c. Note that r_d cannot be updated unless and until the destination is reached. To calibrate boldness levels to arena geometry, distances were specified such that a robot with maximum boldness set its distance goal to the outermost region, and a robot with minimum boldness set its travel distance to the distance with index 0, with intermediate boldness levels linearly spaced between them.

Algorithm 1 SPIDER-density algorithm pseudo-code

```
procedure SPIDER-DENSITY($a$, $b$) ▷ input parameters
Setup: (for individual robots separately)
    $r_c \leftarrow$ robots placed in ring formation
    $bold \leftarrow$ set random boldness $\in [0, 255]$
    $r_d \leftarrow$ floor($bold/10$)
Loop: (every $0.5s$)
    $density \leftarrow$ number of neighbors within range
    $bold = bold + a \times density - b$
    $bold =$ limitrange($bold$, $[0, 255]$)
    $r_c \leftarrow$ travel according to gradient ascent/descent
    if $r_c == r_d$
    $r_d \leftarrow$ floor($bold/10$)
    endif
```

A simple way to introduce a density-induced boldness dynamic to the swarm is to have a boldness that increases when a robot experiences a ‘social interaction’ (receives a message at close proximity), and otherwise decreases at a fixed rate. Because the Kilobots send messages with consistent frequency, the message reception frequency is proportional to how many neighbors are within communication range, and therefore the local density.

Following Algorithm 1, robots move outwards from densely populated central areas towards the outer regions, which are likely to be more sparsely populated, resulting in a drop in their boldness level. Due to the steadily applied boldness decrease, robots are always drawn back toward the center of the swarm until they encounter at least one other robot. These countervailing behaviors ensure the maintenance of a good level of connectivity throughout the swarm, and avoid over-packing of the central region. Boldness increases each time-step according to how many neighbors are within range proportional to the constant a and decreases consistently by constant b (Algorithm 1). These can be optimized for swarm performance according to different user requirements, as discussed in a later section.

A refined boldness mechanism: SPIDER-boldness

In this section, additional refinements are made to the controller, which we refer to as SPIDER-boldness. Its mechanism is outlined using pseudo-code. This controller includes each of following four features, which variously make the swarm controller more or less responsive to certain environmental conditions.

Boldness increase with bold neighbors Following behavior observed in *S. dumicola* (Hunt et al., 2018), the controller was adapted so that robots only increase their boldness when in the presence of bolder individuals. This was set to take effect when a Kilobot's boldness level was below the mean of all its neighbors within communication range. As in all other trials, the robots are initialized with random boldness and boldness is dynamic.

'Addictive' boldness increases Also inspired by behavior observed in *S. dumicola* (Hunt et al., 2018), the behavior was updated to include an 'addictive' term, where robots that had recently increased their boldness were more likely to increase it again by a larger amount. By slowing the boldness increase for the boldest individuals in a cluster, this and the previous feature have the combined effect of preventing clusters of robots quickly increasing each other's boldness before spreading out; while also providing a way for the swarm to avoid a falling ceiling on its overall boldness.

Relative boldness increase/decrease To stabilize the distribution of shy and bold individuals over the swarm, the rate at which an individual altered its boldness level was set to be a function of the individual's current boldness level.

In relation to Algorithm 1, the boldness update was amended in SPIDER-boldness to be:

```
if $mean(neighbor\_boldness) > boldness$ then:
    $bold = bold + f(bold) \times addict - g(bold)$
    $addict = addict \times c$
else:
    $bold = bold - g(bold); \quad addict = 1$
```

For the study presented here

$$f(bold) = \frac{a}{bold}; \; g(bold) = \frac{b}{bold}$$

where a and b are positive constants related to the rate of boldness increase and decay, and c is a term indicating the robot's 'addiction rate'. The choice of function presented here meant that individuals had a tendency to remain bold for longer once their boldness level was significant. Choosing another function would yield different behavior. The values for a and b can be interpreted as having the effect of changing the rate at which the boldness of a robot changes, where $a >> 1$ and $b << 1$ result in stability when the robot is bold, $a << 1$ and $b >> 1$ when it is shy, and intermediate combinations resulting in different stable boldness levels, where higher values for a and b tend to manifest in faster changing and more variable boldnesses.

Updating desired distance en-route As a robot moves throughout the arena, it will continue to receive information on its changing local neighborhood density and could act on this to prevent it from straying too far from the swarm and into a very sparse region, or persisting in attempting to reach the center of the swarm when it is prohibitively crowded. In order to update its behavior dynamically, we allow the robot to change its desired distance from the center in transit to accommodate new information. This may increase the cohesion of the swarm and stops overcrowding in the center as the robots are not bound to completing the task of reaching their waypoint destination.

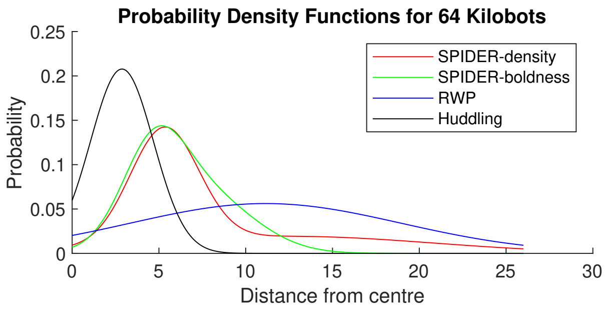

Figure 2: The probability density for robot locations, across the 4 controllers, for a swarm size of 64. Curves are fitted to data collected from 10 simulations using Matlab's Gaussian fit function with 1 peak or 2 peaks for the SPIDER controllers (central bulk and congestion effects). Distance from the center is measured in navigational ring widths (33mm).

We set a threshold difference between the travel distance and the current boldness, which when exceeded causes it to update the desired distance based on the current boldness. In these experiments the threshold was set to update the distance when the boldness had diverged from the goal by more than 2.

Swarms without a boldness mechanism

Two movement behaviors were used to provide a baseline against which to test the effectiveness of the SPIDER algorithm. These had no boldness dynamic or communication between agents, and are random waypoint choice ('RWP'), where each robot sets a random travel distance which is reset upon arrival Aznar et al. (2018); and a 'huddling' behavior where all agents had their distance goal fixed at the center of the arena.

Testing adaptability and robustness

Balancing connectivity and reach Different tasks, or robots with different capabilities, may prioritize the stable connectivity of chains over the extent of their reach into distant areas, or vice versa. For example, maintaining a communication network for emergency services in a difficult to reach, unknown terrain demands a constant, stable, connection, whereas a team of robots searching a less dangerous environment for a resource can operate using only infrequent communication. For this reason it is important that the swarm can easily be adapted to suit the balance of connectivity and reach required.

We anticipate that a swarm prioritising connectivity over exploration is likely to have a boldness that decays very fast when in the presence of few or no neighbors, such that bold robots are infrequent and quickly become shy again; with a slow boldness decay for a swarm that prioritizes exploration over constant connectivity. This hypothesis is readily tested by comparing a short scoring period for the central, 'base' robot, to a long one. We examined a scoring period of 20s rather than 0.5s. This meant that the swarm would increment its score by the most distant connection made between the center and the periphery at any point over the 20s period. This rewards less stable, but more distant connections.

Robustness testing – population catastrophes A desired advantage of a decentralized role distribution mechanism is that it should be robust and consistent in a changing or even hazardous environment. Therefore, for a given environment, the distribution of roles throughout the swarm should stabilize to a similar state that optimizes the reach–connectivity trade-off from a variety of starting conditions. This ensures that the swarm is resilient to noise and can be decentralized in its operation. The swarm should also act to redistribute its roles as appropriate to a changing population, for example following the destruction of agents.

To examine whether the swarms were resilient to extreme change, they underwent a simulated catastrophic population event. This could correspond to the malfunction, removal, or destruction of a significant section of the robots. This motivates the need for a control mechanism suitable for simple, expendable (replaceable) agents. Such an event is simulated by programming half of the swarm to cease communication with the rest of the robots and remove themselves from the arena after a given amount of time. This catastrophe was repeated over the swarm twice over the course of an experiment, changing the population from 128 to 64 to 32, at intervals of 1500s. In a robust system, the remaining swarm should redistribute itself over the boldness space to optimize for connectivity and reach after each catastrophe. Because the boldness distribution changes depending on the swarm size, 20 repeats of 5000 second simulations were run in order to find the stable sub-population distribution that each swarm size converged towards given enough time. The inputs used for the boldness controller were those that had proven to be effective for a population of 64 Kilobots. For a swarm recovering from a catastrophe, the sum of the difference between the stable sub-populations and the current sub-populations gave a measure for the distance of the swarm from equilibrium, and the volatility of the swarm undergoing a catastrophe was calculated as the sum of the standard deviations in each of the sub-populations over 50s intervals. The time taken for a swarm to settle after a population catastrophe was approximated as the time taken for the distance from equilibrium to fall and remain below 0.3 and the volatility to fall and remain below 0.1.

Automated parameter optimisation

Given the complex relationship between input model parameters for individual-level robot behavior, and overall swarm performance for different swarm sizes and user connection stability requirements, we optimize parameters a, b and c using an evolutionary algorithm (EA), with rank-based selection on swarm-level performance. The EA worked over 20 generations on the 3 'genes' (i.e. a, b, and c) with elitism, and a combination of crossover without contamination, crossover with contamination (all 'genes' from each simulation were broken into factors and paired with a corresponding factor from another simulation), with small consistent mutations (value jitter of ±10%).

Results[1]

Random waypoint choice

For the purpose of analysis swarm density is calculated by dividing the combined area reached by the robots' (here, Kilobots') communication range by the area of the arena. The swarms were scored for populations of 8, 16, 32, 64, and 128, corresponding to densities of 0.0329, 0.0657, 0.1315, 0.2630, and 0.5260 respectively.

Up to a certain population density, robots randomly choosing their desired distance produced a symmetric Gaussian-like probability density function (pdf) when the individuals' positions were concatenated and integrated over time (Figure 2). As the population approached a swarm density of 0.5, collisions and crowding at the center significantly skewed the distribution towards the center of the arena. Due to this skew and higher density, performance by larger populations of robots following the random waypoint behavior continued to increase, such that the performance did not differ significantly from swarms with a boldness dynamic (Figure 3). Thus, at high swarm densities, more complex control strategies are no more effective than a simple approach (Hunt et al., 2019).

Huddling behavior

With robot boldness set to a minimum, there was consistent connectivity in the central arena region at the expense of not reaching into further radial distances. Nevertheless, this was sufficient for performance scores comparable with boldness-dependent behaviors at very low populations. As swarm size grew, the performance of risk-taking exploratory behaviors quickly overtook this 'huddling' strategy. However, it is a reasonable approach to reaching a small amount of the area if connectivity is paramount to the task and only a small population is available. It outperforms the random waypoint be-

[1] All data analysed here is available on GitHub: https://github.com/geojenks/adaptive-risk-appetite

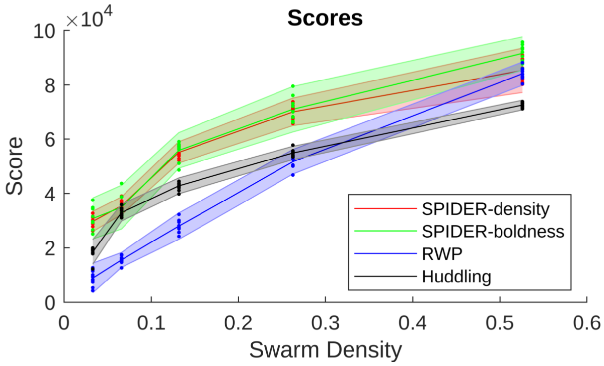

Figure 3: Performance metric score comparison for the 4 controllers (95% confidence). See Fig. 1 for the scoring method: it rewards successful transmission of distant messages back to base. At interim densities the boldness-based controllers outperform huddling and random r_d strategies (SPIDER-boldness robots also travel less distance; see Results).

havior up to a swarm density of ≈ 0.25 presumably owing to stable connectivity (Figure 3).

SPIDER-density controller performance

The controller parameters selected by the EA for small swarms tended to favor low risk strategies whereby robots remain close to the central region where they are likely to be connected to the main cluster. As the population grew, the selected parameters produced swarms containing robots much more likely to be found in riskier regions further from the arena center (N=64 shown in Figure 2).

Performance metric score The basic boldness mechanism outperforms the random waypoint (RWP) behavior beyond a confidence level of 95% until a swarm density of around 0.38. Beyond this, the RWP and basic boldness behaviors score more closely. At a population density of 0.5, the mean score of the basic boldness swarm does not fall outside of a single standard deviation of that of the RWP. The basic boldness behavior also outperformed the huddling behavior across the higher range of population densities but not significantly at the lowest densities (Figure 3). This indicates that for small populations, the strategy that the EA evolves is rather risk-averse.

Population catastrophe The SPIDER-density controller sometimes resulted in moderately fluctuating sub-populations of each boldness class, as can be seen in the initial stages of Figure 4A. This may have had the effect of increasing the speed at which the swarm can react to a change such as a population catastrophe; being the fastest controller to stabilize, the swarm took an average of 542.9s to stabilize at a new distribution (Figure 4A B).

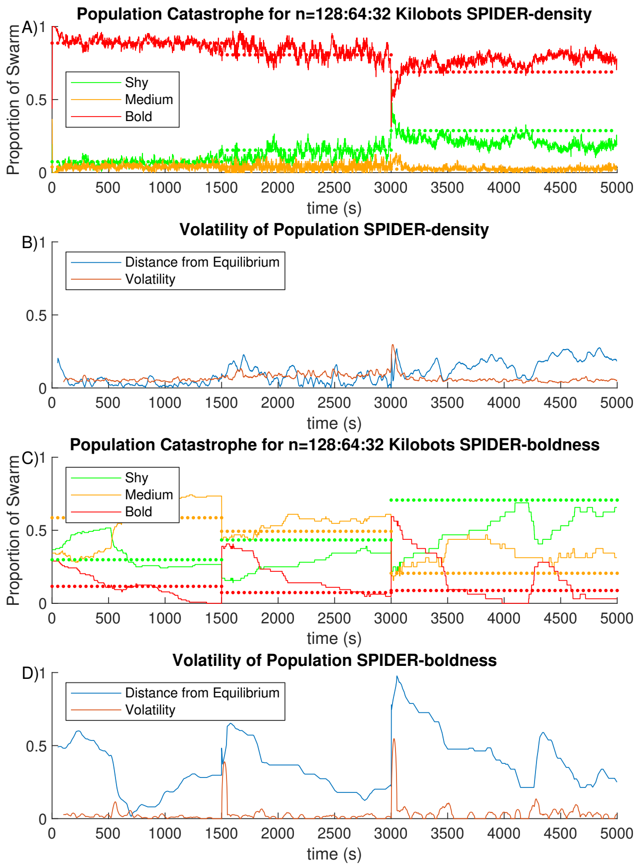

Figure 4: Example trials of the SPIDER-density (panes A and B) and SPIDER-boldness (C and D) controllers with a ‘population catastrophe’. The population is halved twice from 128 to 64 to 32 robots at 1500s and 3000s. Dotted lines show the sub-population proportions towards which longer simulations converge.

SPIDER-boldness controller performance

The SPIDER-boldness controller was particularly effective for larger swarms. This is likely because as the population increases, it becomes more difficult to travel to or from the central region due to blocking, so being able to abandon the destination and select a new desired distance allowed robots to leave the overcrowded central area. The distribution of the robots over the environment is less positively skewed into distant regions than for the more simple SPIDER-density controller. This is a reflection of the decaying local swarm density (Figure 2), as a robot will not continue to move outwards far beyond the periphery of the swarm if it does not encounter another agent. The centroid of the Gaussian fit to the distribution moves through 3.88, 4.21, 5.52, 8.41, as the population redoubles from 16 to 128. The regions toward the edge of the arena are relatively unexplored, despite the swarm having a considerable bold proportion. This is likely because as robots travel toward far regions, their boldness falls and so they adjust their distance to be closer to the center before they reach their previous distance goal.

Performance metric score The SPIDER-boldness controller allows the swarm to significantly outperform random waypoint (RWP) behavior up to a population density of 0.5. For low populations, the swarm-wide behavior was similar to a swarm of minimal boldness, as most robots were required to be close to the center to maintain any connection at all. The swarms controlled by this behavior outperformed the RWP behavior beyond a confidence level of 95% by a factor of 3.2, 2.3, 2.0, 1.4 up to a density of 0.5, at which point it outperformed the RWP behavior by a factor of 1.1 at a confidence level of 90%. The SPIDER-boldness behavior significantly outperformed the huddling behavior with a confidence level above 95% for population densities above 0.11, by a factor of 1.3, 1.3, and 1.3 at the swarm densities tested (Figure 3).

Population catastrophe The sub-populations gradually approached their stable proportion after a catastrophe and did not overshoot or oscillate heavily (Figure 4C). The swarm stabilized after an average of 615s following the population being halved (Figure 4C D). This reliable recovery is possible due to the combination of the stabilizing effects of increasing robots' boldness levels only when they were surrounded by bolder neighbors, at a rate relative to their current level, and the ability adapt their goal en-route.

Suitability for less connection-critical tasks

The EA was used to optimize the SPIDER controllers at a longer scoring time interval of 20s. By recording the furthest successfully relayed message over this longer time period, there was scope to tolerate longer interim time periods where less distant or no messages were received at the central base. This resulted in a distribution of roles was then bolder than when scoring over a shorter time interval of 0.5s (Table 3). This demonstrates that the boldness dynamic can be altered to be suitable for different tasks, or for robots with different capabilities, depending on how the user prioritizes connectivity of chains or further reach – exploration and information gathering – from peripheral regions.

Overall comparison of SPIDER variations

Both the controllers increased the average boldness over the swarm as the population grew (Table 2). The similar stable boldness distributions for the two mechanisms show that the boldness dynamic is a reliable way to reach a role distribution. It also allows the swarm to effectively react to its current state in order to stabilize or alter its role distribution.

The SPIDER-density controller was capable of adapting to population changes far more quickly than the SPIDER-boldness controller (Table 1). The robots controlled by SPIDER-boldness changed their boldness levels relative to their current level, meaning that bold robots remained bold – and in isolation – for much longer than the simple density-based behavior. Therefore they took longer to react to population reduction and the need for more shy robots. The utility of swarm role stability versus adaptability is task-specific.

Population Catastrophe Recovery Time (s)						
N=Pre:Post	SPIDER-density			SPIDER-boldness		
	mean	s.d.	success	mean	s.d.	success
128:64	**62.5**	76.8	100%	361.5	89.3	100%
64:32	**40.0**	21.0	100%	110.5	49.9	100%
128:64:32	**51.3**	48.9	100%	236.0	69.6	100%

Table 1: Swarms' abilities to stabilize their shy-medium-bold distribution after two population catastrophes. Means are from 10 repeat experiments. Both SPIDER controllers stabilized the role distribution after a catastrophe. The fastest recovery times are emboldened.

Average Boldness of SPIDER Algorithms				
Population	SPIDER-density		SPIDER-boldness	
	mean	s.d.	mean	s.d.
16	0.19	0.088	0.19	0.045
32	0.20	0.008	0.21	0.036
64	0.25	0.057	0.25	0.070
128	0.25	0.040	0.31	0.046

Table 2: Average boldness sustained over the last 1500s of 5000s simulations, from the best scoring swarms for SPIDER-density and SPIDER-boldness. Both controllers increase the mean boldness over the swarm for higher populations.

Boldness in Swarms of Size N=64 For Two Scoring Periods				
Score Period	SPIDER-density		SPIDER-boldness	
	mean	95% range	mean	95% range
0.5s (fast)	0.243	0.231:0.255	0.262	0.239:0.284
20s (slow)	0.30	0.273:0.327	0.342	0.319:0.365
slow/fast	1.23		1.31	

Table 3: When scoring the swarm performance metric over longer time periods, the boldness dynamics are optimized for a more risk-taking swarm, because it permits short-term connectivity in the pursuit of a further reach.

The two controller variants achieve similar scores up to a swarm density of 0.263, yet the SPIDER-boldness controller executes the task while consistently travelling significantly less distance, on average covering 85% of the linear distance, and so expending less energy . The en-route updating means that an agent's behavior was closely linked to its boldness. By comparison agents using the SPIDER-density controller often had long periods where their behavior had been set at a time when their boldness was significantly different to its present level. Therefore, it is likely that using a SPIDER-boldness controller with input parameters a, b, c that have been evolved, or otherwise developed, to minimise boldness while achieving a high score, provides a path to minimising the swarm's energy consumption.

Conclusion and further work

Our bio-inspired boldness dynamic provides the foundational theory for a computationally cheap, decentralized mechanism by which a swarm of robots can arrive at a stable distribution of roles in relation to risk appetite (boldness or willingness to explore). This boldness dynamic, which we call the SPIDER algorithm, significantly improves the swarm's ability to reach from a base into a potentially hazardous distant area with suitable levels of connectivity.

Demonstrating these boldness mechanisms in real robots – Kilobots – is a potential next step in validating their effectiveness. Equally, although we have examined very simple robots, the SPIDER algorithm could be implemented in robots with more sophisticated navigational and localisation abilities. There is also an opportunity to deploy the SPIDER algorithm in real-world field trials in hazardous environments, to demonstrate the effectiveness of a density-based risk appetite controller for optimising the connectivity–reach trade-off.

Acknowledgements

EH acknowledges support from the Royal Academy of Engineering and the Office of the Chief Science Adviser for National Security under the UK Intelligence Community Postdoctoral Fellowship Programme. MW acknowledges support from the Engineering and Physical Sciences Research Council (EP/L016656/1 and EP/N002458/1); and the University of Bristol.

References

Aznar, F., Pujol, M., Rizo, R., Pujol, F. A., and Rizo, C. (2018). Energy-efficient swarm behavior for indoor uav ad-hoc network deployment. *Symmetry*, 10(11):632.

Bénichou, O. and Redner, S. (2014). Depletion-controlled starvation of a diffusing forager. *Physical Review Letters*, 113(23).

Bilde, T., Coates, K. S., Birkhofer, K., Bird, T., Maklakov, A. A., Lubin, Y., and Avilés, L. (2007). Survival benefits select for group living in a social spider despite reproductive costs. *Journal of Evolutionary Biology*, 20(6):2412–2426.

Dall, S. R. X., Houston, A. I., and McNamara, J. M. (2004). The behavioural ecology of personality: consistent individual differences from an adaptive perspective. *Ecology Letters*, 7(8):734–739.

Garattoni, L. and Birattari, M. (2018). Autonomous task sequencing in a robot swarm. *Science Robotics*, 3(20):1–12.

Hauert, S., Mitri, S., Keller, L., and Floreano, D. (2014). Evolving cooperation: From biology to engineering. In Vargas, P. A., Di Paolo, E. A., Harvey, I., and Husbands, P., editors, *The Horizons of Evolutionary Robotics*. MIT press.

Hauert, S., Winkler, L., Zufferey, J. C., and Floreano, D. (2008). Ant-based swarming with positionless micro air vehicles for communication relay. *Swarm Intelligence*, 2(2-4):167–188.

Hauert, S., Zufferey, J.-C., and Floreano, D. (2009). Evolved swarming without positioning information: an application in aerial communication relay. *Autonomous Robots*, 26(1):21–32.

Hunt, E. R. (2020). Phenotypic Plasticity Provides a Bioinspiration Framework for Minimal Field Swarm Robotics. *Frontiers in Robotics and AI*, 7(23).

Hunt, E. R., Jones, S., and Hauert, S. (2019). Testing the limits of pheromone stigmergy in high-density robot swarms. *Royal Society Open Science*, 6:190225.

Hunt, E. R., Mi, B., Fernandez, C., Wong, B. M., Pruitt, J. N., and Pinter-Wollman, N. (2018). Social interactions shape individual and collective personality in social spiders. *Proceedings of the Royal Society B*, 285(20181366):12–14.

Jones, S., Studley, M., Hauert, S., and Winfield, A. (2018). Evolving Behaviour Trees for Swarm Robotics. In *Proceedings of the 13th International Symposium on Distributed Autonomous Robotic Systems*, pages 487–501. Springer International Publishing.

Krieger, M. J. and Billeter, J. B. (2000). Call of duty: Self-organized task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems*, 30(1):65–84.

Li, G., Svogor, I., and Beltrame, G. (2019). Long-term pattern formation and maintenance for battery-powered robots. *Swarm Intelligence*, 13(1):21–57.

McGuire, K. N., De Wagter, C., Tuyls, K., Kappen, H. J., and de Croon, G. C. H. E. (2019). Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment. *Science Robotics*, 4(35).

Nouyan, S. and Dorigo, M. (2006). Chain based path formation in swarms of robots. In Dorigo, M., Gambardella, L. M., Birattari, M., Martinoli, A., Poli, R., and Stützle, T., editors, *Ant Colony Optimization and Swarm Intelligence*, pages 120–131, Berlin, Heidelberg. Springer Berlin Heidelberg.

Rubenstein, M., Ahler, C., and Nagpal, R. (2012). Kilobot : A Low Cost Scalable Robot System for Collective Behaviors. In *2012 IEEE International Conference on Robotics and Automation*, Saint Paul, Minnesota, 3293-3298. Washington, D.C. Computer Society Press of the IEEE.

Schranz, M., Umlauft, M., Sende, M., and Elmenreich, W. (2020). Swarm robotic behaviors and current applications. *Frontiers in Robotics and AI*, 7:36.

Sperati, V., Trianni, V., and Nolfi, S. (2011). Self-organised path formation in a swarm of robots. *Swarm Intelligence*, 5(2):97–119.

Yun, S. K. and Rus, D. (2014). Adaptive coordinating construction of truss structures using distributed equal-mass partitioning. *IEEE Transactions on Robotics*, 30(1):188–202.