Evolving a Repertoire of Controllers for a Multi-Function Swarm

Source code available: https://goo.gl/Vmp2s

Sondre A. Engebråten, Jonas Moen, Oleg Yakimenko, and Kyrre Glette
sondre.engebraten@ffi.no

Motivation

1. Top-down generation of swarm behaviors
2. New behavior primitives for multi-UAV operations
3. Adaptation to human preferences

Our UAV swarm explores an area from most visited squares (darker) to least visited squares (lighter). You can do the same with this poster!

A swarm controller

Weighted

$$\mathbf{v}_{sp} = \frac{1}{4}\sum_{i}^{N} \frac{\mathbf{F}_i}{||\mathbf{F}_i||} * w_i$$

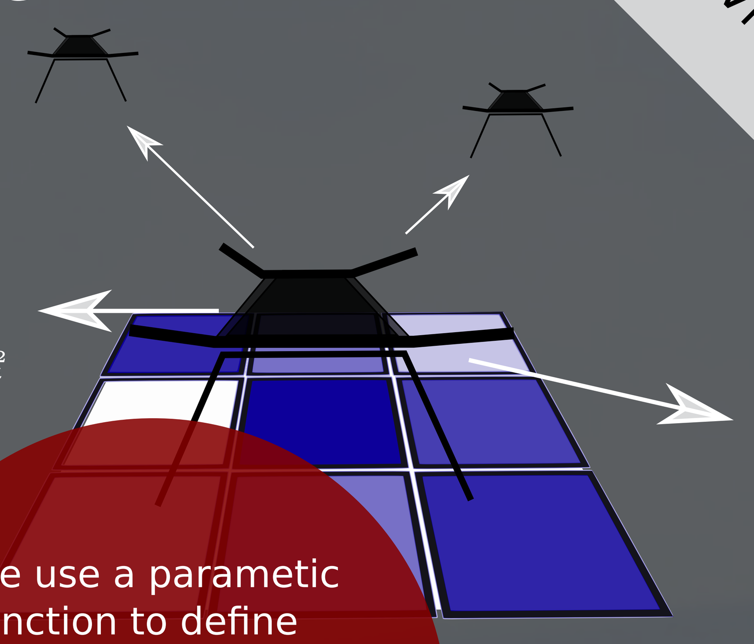
Parametric

$$g_i(d_i) = -t_i * 2 * (d_i - c_i) * e^{-(d_i-c_i)^2/\sigma_i^2}$$

$$a_i(d_i) = k_i * \left(\frac{2}{1+e^{-(d_i-c_i)/\sigma_i}} - 1\right)$$

$$w_{p,i}(d_i) = a_i(d_i) + g_i(d_i)$$

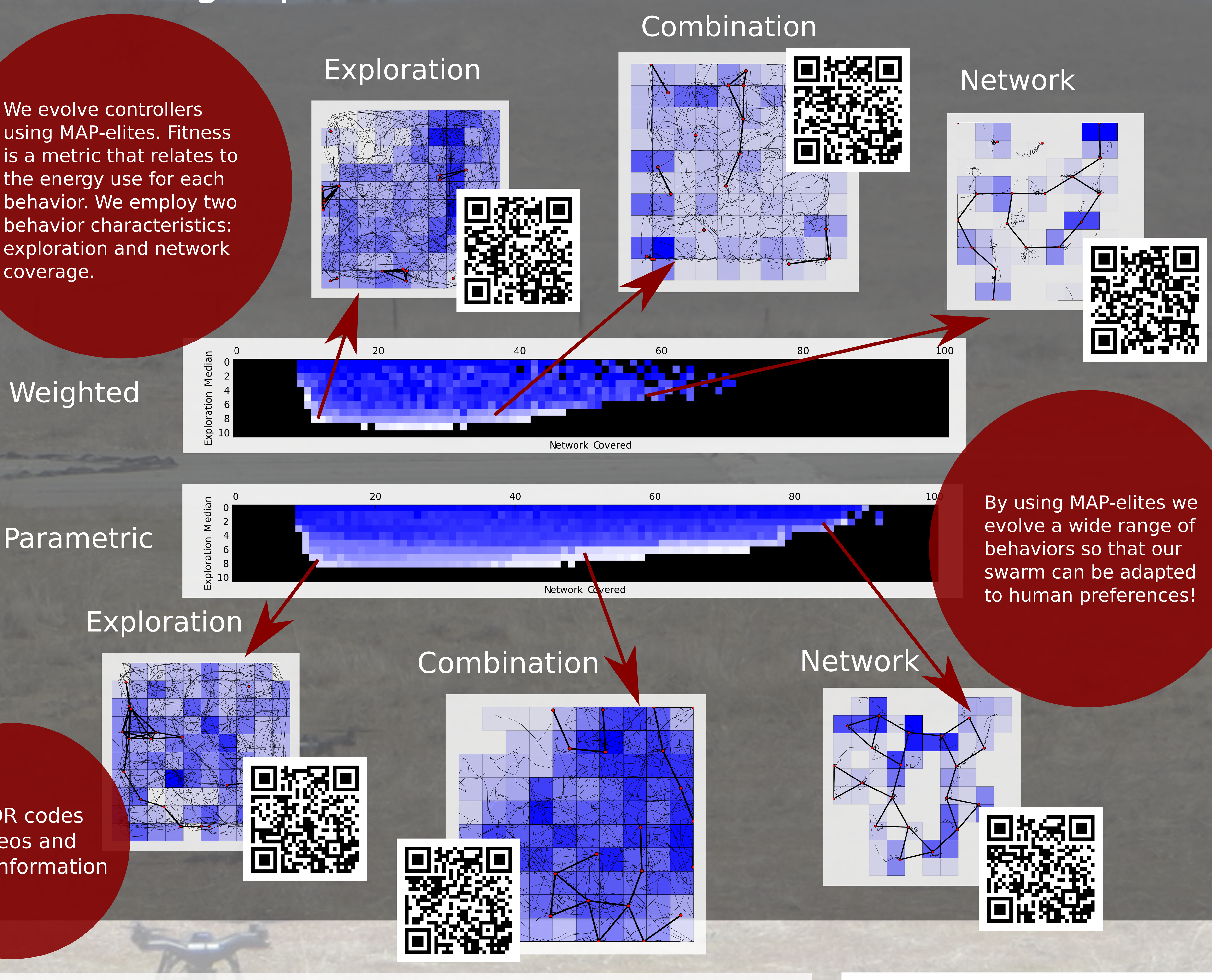
$$\mathbf{v}_{sp} = \frac{1}{4}\sum_{i}^{4} \frac{\mathbf{F}_i}{||\mathbf{F}_i||} * w_p(d_i)$$



We use a parametric function to define swarm behaviors. Our controller is defined by 16 real-coded values, that in turn define the swarm behavior

Evolving repertoires of controllers

We evolve controllers using MAP-elites. Fitness is a metric that relates to the energy use for each behavior. We employ two behavior characteristics: exploration and network coverage.



By using MAP-elites we evolve a wide range of behaviors so that our swarm can be adapted to human preferences!

Scan QR codes for videos and more information

Challenges

1. Holding distance

Our initial controller used a single scalar to weight each input force. This made it hard to evolve behaviors that would hold a distance to neighboring agents. For this reason we extended the controller structure to a parametric function that depends on distance allowing agents to hold a distance to other agents.

2. Stochastic fitness

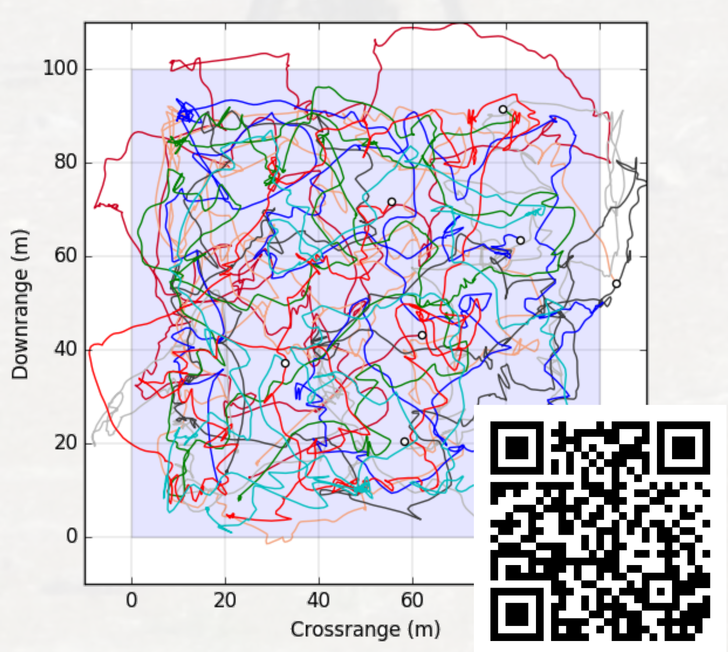
The fitness evaluation is based on a simulator. This results in a stochastic fitness evaluation. Variance in fitness estimates is problematic when using a greedy algorithm such as MAP-elites as it allows behaviors a chance to get lucky.

3. Reality gap

We designed the controller structure at a high abstraction level (using velocity setpoints for control) in the hopes that this would ease the transition to real UAVs in the future. However, this does not fully address the reality gap for these experiemts. The difference in simulated vehicle response and real UAVs is still a challenge for future work.

Field-Testing of High-Level Decentralized Controllers for a Multi-Function Drone Swarm

Real flight log 8 UAVs



Our proposed controller structure was ported to a swarm of real UAVs. Due to the fact that our sensor input can be generated with minimal communication between agents there were no real obstacles to converting the proposed controller to UAVs. Our greatest challenge with the real-world swarm today, is the difference between simulated UAV responce and the real response.

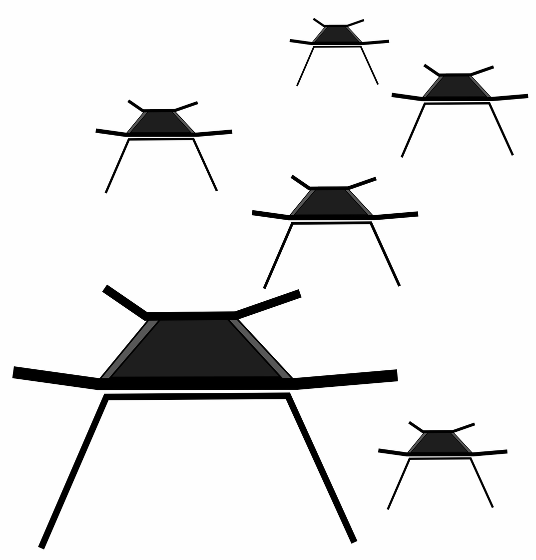
By making use of available COTS hardware we have managed to develop a real swarm to verify our experiments. You can read more about this in our upcoming publications at IEEE ICCA 2018

Conclusion

1. We show that it is possible to evolve a repertoire of controllers for a multi-function swarm.

2. By using a parametric function we are able to evolve widely varied behaviors in combination with any desired tradeoff between multiple applications.

3. In future work, we wish to explore more varied behaviors for new applications, transition dynamics and more test with real UAVs. Initial experiments show promise in allowing this controller structure to be used on a real UAV swarm.



A simple parametric function is sufficient to make a wide repertoire of behaviors that can be used on a real-world swarm

FFI Forsvarets forskningsinstitutt
NPS PRAESTANTIA PER SCIENTIAM
UiO : Department of Informatics
University of Oslo