Novelty and Objective-based Neuroevolution of a Physical Robot Swarm

Forrest Stonedahl[1], Susa H. Stonedahl[2], Nelly Cheboi[1], Danya Tazyeen[1] and David Devore[1]
[1]*Math and Computer Science Department, Augustana College, Rock Island, Illinois, U.S.A.*
[2]*Engineering and Physical Science Department, St. Ambrose University, Davenport, Iowa, U.S.A.*

Keywords: Novelty Search, Neuroevolution, Multi-agent Robotics, Exploration.

Abstract: This paper compares the use of novelty search and objective-based evolution to discover motion controllers for an exploration task wherein mobile robots search for immobile targets inside a bounded polygonal region and stop to mark target locations. We evolved the robots' neural-network controllers in a custom 2-D simulator, selected the best performing neurocontrollers from both novelty search and objective-based search, and compared performance relative to an unevolved (baseline) controller and a simple human-designed controller. The controllers were also transferred onto physical robots, and the real-world tests provided good empirical agreement with simulation results, showing that both novelty search and objective-based search produced controllers that were comparable or superior to the human-designed controller, and that objective-based search slightly outperformed novelty search. The best controllers had surprisingly low genotypic complexity, suggesting that this task may lack the type of deceptive fitness landscape that has previously favored novelty search over objective-based search.

1 INTRODUCTION

Within evolutionary robotics, there has been growing enthusiasm surrounding the concept of novelty search (NS) (Lehman and Stanley, 2011), wherein the evolutionary algorithm focuses solely on generating novel behaviors, without regard to objective measures that quantify robot performance on the desired task. This re-integration of the idea of open-ended evolution into solving performance-based machine learning tasks is intriguing, and promising results have been published for a variety of domains, including 2-D maze navigation and bipedal walking (Lehman and Stanley, 2011), tunable deceptive T-mazes (Risi et al., 2009), and simulated robot swarm aggregation and resource sharing tasks (Gomes et al., 2013). Extending this general line of research, our paper documents one of the first uses of NS for evolving neurocontrollers that are employed in a *physical* swarm robotics exploration experiment. In the remainder of the paper, we will define our swarm robotics task, describe the software simulator, explain the evolutionary search for neurocontrollers, and discuss the results of simulated and physical robot experiments with those controllers.

2 TASK SPECIFICATION/ BACKGROUND

While some have defined swarm robotics as involving the coordination of large numbers of agents, we ascribe to the definition proposed in a recent review paper (Brambilla et al., 2013) wherein the main characteristics of swarm robotics system are that robots are *autonomous*, situated in a changeable environment, possess only *local sensing/communication*, *lack centralized control* and/or global knowledge, and *cooperate* on a given task. Thus, although we only employ eight robots, we prefer to frame this task within the genre of swarm robotics because of the manner in which the robots are allowed to interact.

2.1 Multi-agent Search Task

Disaster recovery has been identified as an important real-world application where collaborative robot teams could provide a great benefit to society (Davids, 2002), and our experimental task is based on a loose analogy to the following search/rescue scenario. We are concerned with the task of collaborative exploration of a region for which the robots will not possess a map or any *a priori* knowledge about the shape of the region. For economic scalability, the individ-

Stonedahl F., Stonedahl S., Cheboi N., Tazyeen D. and Devore D.
Novelty and Objective-based Neuroevolution of a Physical Robot Swarm.
DOI: 10.5220/0006118303820389
In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence (ICAART 2017)*, pages 382-389
ISBN: 978-989-758-220-2

ual robots will each be quite simple and endowed with only limited capacities for sensing, and no form of direct inter-robot communication. The robots will possess basic locomotion (change heading/move forward), distance sensors that can detect obstacles nearby (which could be walls or other robots), and some form of specialized local sensors that can detect "targets" (possibly disaster victims, chemical hazards, etc.), but only at very close range. Once a robot encounters a target, it will remain at that location (e.g. providing aid and/or communication services to the victim, providing a beacon for rescue workers to track, etc.) The robot swarm's goal is to spread quickly throughout the disaster domain and collectively locate as many targets as possible.

We abstract this specific real-world task into the following simplified version of the problem, which we will refer to as the Multi-Agent Multi-Target Search and Stay (MAMTSS) problem. Given a bounded (flat) 2-D polygonal region, N targets are placed within the region, and N robot agents are deployed along one boundary of the region. The robots are launched simultaneously, and after a fixed time limit, the success of the robot team is measured by the fraction of the N targets that were located during that time period. In the present work, we disallow any explicit communication between robots. However, a robot's distance or bump sensor may detect other robots, even though the sensor cannot distinguish whether it has reacted to a wall or another robot. Thus, robots may still influence other robots' behavior without communicating directly, similar to a flock of "boids" (Reynolds, 1987).

2.2 Related Work

The MAMTSS problem is most closely related to the team *coverage task*, where the robots' collective goal is that every location has had a robot pass over it, as in the examples of lawn mowing or vacuuming (Choset, 2001; Rekleitis et al., 2004). Since the targets in MAMTSS are placed randomly in the region, our task could almost be described as a stochastic sampling method for estimating "coverage"; however, it differs slightly since robots that reach a target remain immobile at that spot thereafter, rather than continuing to explore. Some approaches to the coverage task use *a priori* global map knowledge to *guarantee* complete coverage (Rekleitis et al., 2004), while others employ stigmergy, such as using artificial pheremones to mark cells in the environment as explored (Wagner et al., 2008). It has also been shown that with enough robots, even local obstacle avoidance behavior can achieve decent coverage of the space (Ichikawa and

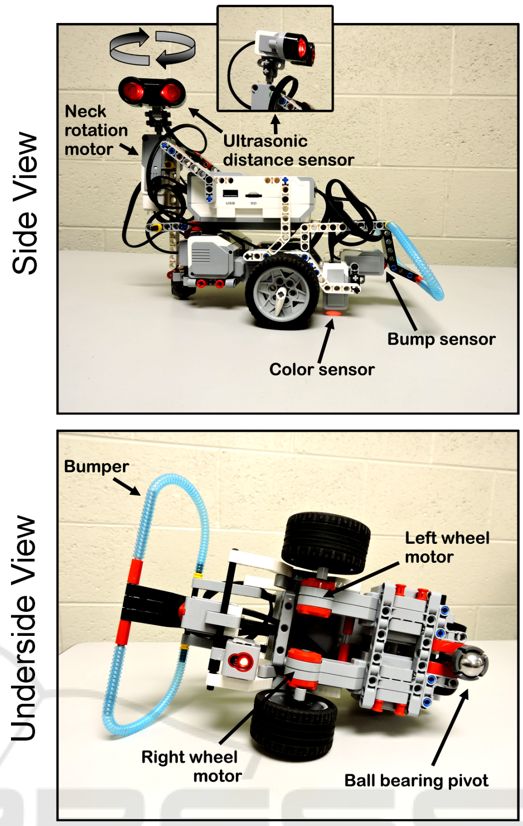

Figure 1: Physical robot design. Two wheel motors provide differential steering, and a third motor swivels the distance sensor to take measurements at 30° increments.

Hara, 1999), which is relevant because our MAMTSS task relies on local sensing/geometry rather than on global knowledge, direct communication, or even indirect communication via stigmergy. In contrast to their specific human-coded navigation strategy, we are *evolving* robot motion controllers. Another related task is *robot dispersion* (McLurkin and Smith, 2007) although there the robots' goal is to spread out evenly throughout the space, rather than to locate as many targets as possible.

2.3 Physical Setup and Robot Specifications

We designed and built 8 identical robots using the LEGO^{TM} Mindstorms EV3 robotics kit, as pictured in Figure 1. While these consumer-grade robots would be inadequate for rugged real-world search and rescue missions, they are well-suited for our simplified task. The ease of modification and configuration makes them a versatile research tool, and Mindstorms robots have been successfully applied in previous evolutionary robotics research studies (Lund,

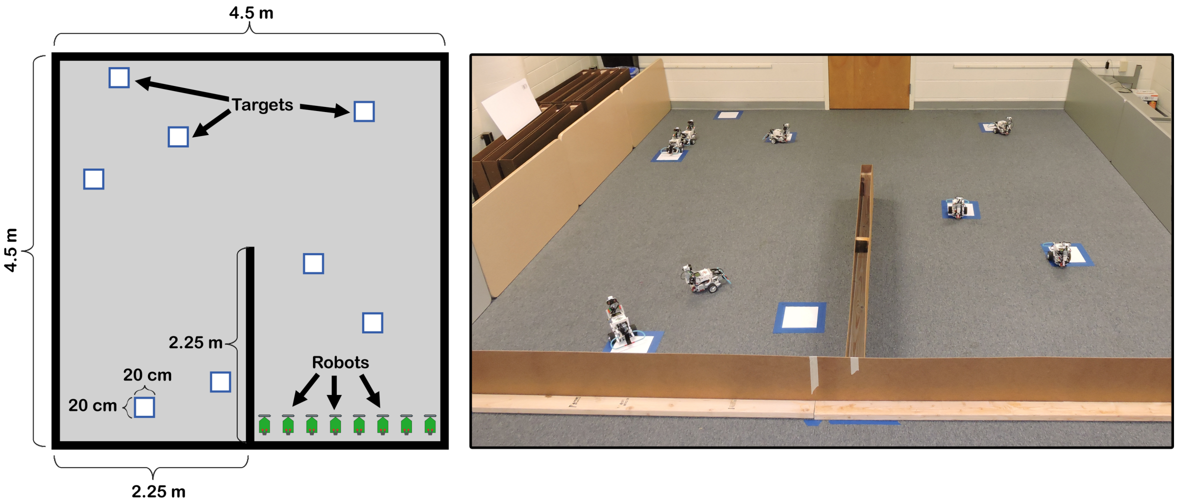

Figure 2: *LEFT:* Initial top-down layout for the MAMTSS task. The interior wall was 10 cm thick and all walls were at least 30 cm tall. *RIGHT:* The physical MAMTSS environment, at the end of an 8-minute trial where the team located 6 of 8 targets.

2003; Parker and Georgescu, 2005). The EV3 model features a 300 Mhz ARM9 processor, 64MB of RAM, runs embedded Linux, and is programmable in Java using the open-source LeJOS framework. Each robot independently repeats three phases: sensing, turning, and movement. During the *sensing phase* it takes distance measurements at 12 regularly-spaced (30°) angles. During the *turning phase* it chooses an angle to rotate based on the 12 distance measurements. During the *movement phase* it moves straight forward 40 cm. If it runs into an obstacle (other robot/wall) that triggers the "bump sensor" on the front of the robot, it moves backward 10 cm, and then starts a new sensing phase. If the robot's color sensor detects a white target beneath it at any point, the robot will cease movement and stay at that location for the remainder of the trial. For our physical implementation of the 8-agent MAMTSS task, we designed a simple orthogonal U-shaped environment, as shown in Figure 2. For each trial, the robots were given 8 minutes to explore the region and locate as many goals as possible.

3 SIMULATION AND NEURO-EVOLUTION

3.1 Neural Network Design

We evolved simple feed-forward artificial neural networks (ANNs) with 12 input neurons (corresponding to the 12 equiangular measurements from the distance sensor), one output neuron (which controls the angle for each robot's turning phase), and a variable number of hidden-layer neurons (added during neuroevolution). The robot's ultra-sonic distance sensor has a maximum range of 2.5 meters, and will report a value of "Infinity" for "out of range", which we translated into 10 m. The distance measurements in meters were normalized and fed into the neural network. Since a sigmoidal activation function would bias the output angle toward sharp turns, the output neurons were assigned a linear activation function. However, hidden layer neurons used a sigmoidal activation function to permit the construction of nonlinear functions. The final neural output was adjusted/scaled to always be between -180 and 180 degrees.

3.2 Simulation Software

The time required for running evolutionary algorithms on the physical robots themselves was prohibitive, so we used the NetLogo platform (Wilensky, 1999) to develop a custom 2-D mobile robot simulator for this task. (The diagram shown in Figure 2 is based on a screenshot from this simulator.) Based on extensive calibration measurements with the physical robots, we incorporated realistic levels of Gaussian noise into the sensor data and actuator error within the simulator.

3.3 Search Algorithm Experiments

We connected the simulator to the AHNI framework (Coleman, 2012) for neuro-evolution, and applied the well-established NEAT search algorithm (Stanley and Miikkulainen, 2002) for both objective-based search and novelty search, to evolve neurocontrollers for the robots. The search parameters, given in Table 1, were chosen from reasonable ranges based on previ-

Table 1: Search algorithm parameters.

Parameter	Value
Population size	100
Simulation trials per fitness eval.	10
Add neuron mutation rate	.03
Add connection mutation rate	.30
Remove connection mutation rate	.01
Weight mutation rate (& stdev)	.80 (1)
Min/max connection weight range	$[-8,8]$
Number of generations	120
Crossover rate	.75
Survival rate	.40
Elitism proportion	.10
Novelty k (nearest neighbors)	15
Novelty threshold	.05

ous work (Stanley and Miikkulainen, 2002; Lehman and Stanley, 2011). Apart from the use of either objective or novelty to guide the evolution process, all other aspects of the two search experiments were identical. For each generation, 10 new random seeds were used to determine the random positions of the 8 targets within the environment, and all individuals within that generation performed trials on those 10 course layouts. An individual's "performance" score was calculated as the average (across the 10 trials) fraction of goals found within the 8 minute trial simulation. For objective-based search, the fitness score was the same as the performance score. For novelty search, the performance scores were calculated for extrinsic record-keeping, but they did not influence the search process in any way. Instead, individuals were selected for reproduction based on the *novelty* of their behavior, with behavior characterized as a high-dimensional vector of the positions of the robots over time. Specifically, normalized robot x and y coordinates (between 0.0 and 1.0) were collected for each robot every 10 (simulated) seconds. The 10 trials produced 10 such histories, and these were condensed by taking the mean and the standard deviation *across trials*, thus storing data estimating each robot's distribution of possible positions over time. Following prior research (Lehman and Stanley, 2011), novelty was calculated using the Euclidean distance between this behavior vector and the vectors already stored in the novelty archive[1]. Extrinsic to the search process, the best-performing individual in each generation was recorded, and its performance was re-evaluated using 30 random seeds in order to obtain a more accurate and unbiased estimate of that individual's true performance level (for plotting and analysis of results).

[1] For an introduction to novelty search and more details about the method, see (Lehman and Stanley, 2011)

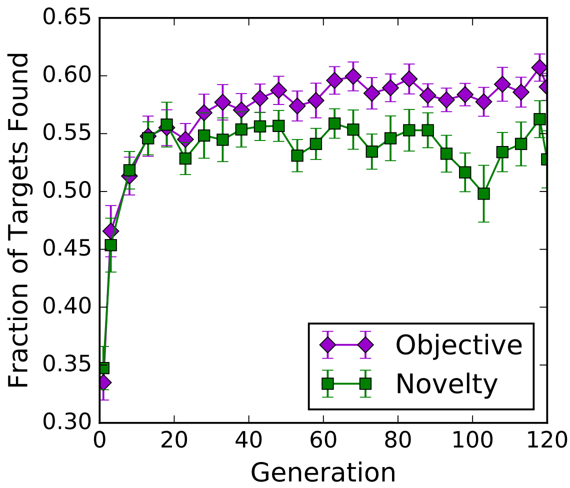

Figure 3: Average performance of the best individuals during neuroevolution. Error bars show 95% confidence intervals for the mean.

4 RESULTS AND DISCUSSION

4.1 Search Algorithm Results

The full search algorithm experiments described above for objective-based and novelty search were each run 30 times. The average performance over evolutionary time is shown in Figure 3. In theory, a perfect solution would have a performance value of 1, indicating that the robots located every target in all simulated trials. There are pragmatic reasons why this theoretical maximum is likely unattainable: a) the robots were only given 8 minutes, which is probably insufficient to completely cover the region b) the robots' lack of communication makes some duplication of coverage unavoidable, and c) the targets are large enough that it is not uncommon for two robots to find and "stay" at the same target (although the hope is that they will sense the other robot's presence and turn away). Given these factors, we judged that both searches were able to discover fairly good solutions relatively quickly, suggesting that the algorithms are working effectively, although the MAMTSS problem as we have posed it may be a less challenging benchmark problem than we had anticipated.

Recall that NS is not guided toward high performance individuals, but is instead guided toward novel behaviors, and is often able to find high performance individuals along the way (Lehman and Stanley, 2011). After finding high-performance individuals and adding their behaviors to the novelty archive, NS's appetite for novelty may lead it toward less-fit behaviors, a phenomenon which likely explains

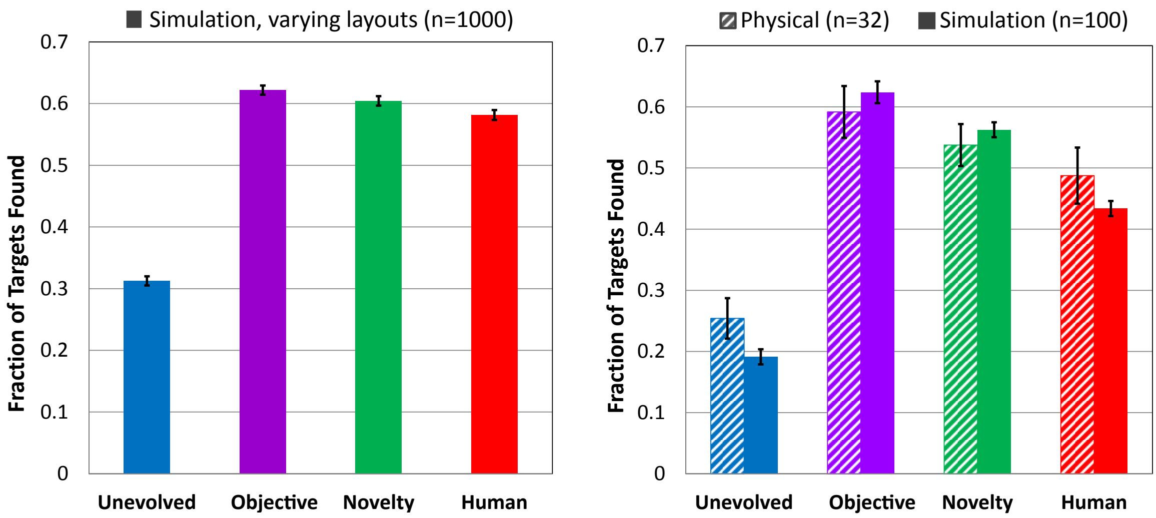

Figure 4: *LEFT:* Performance on the 8-agent MAMTSS task based on 1000 trials with different (random) target layouts. *RIGHT:* Comparison of physical & simulated trials for one specific (fixed) target layout. Error bars show 95% confidence on the mean.

why there is a slight downward slope for novelty search's performance in later generations. The average final-generation performance for objective-based search was statistically higher (t-test, $p < 0.01$) than for novelty search, but this was not the case in generation 35 ($p > 0.6$).

From the 30 objective-based searches, we selected the highest-performing individual from any of the 30 final populations. From the 30 novelty searches, we selected the highest-performing individual from any generation, which happened to be generation 76 of one of the runs.

Because performance varies considerably based on the placement of the targets, we ran a more extensive test of the performance of the best neurocontroller from objective-based search and novelty search. To see the progress evolution had made, as a baseline we also included an "unevolved" neurocontroller which was the best performing controller (out of 100 randomly generated individuals) from the initial population of the same objective-based search that eventually produced the best performer. Finally, we included a human-designed controller in the test, to see how the evolved solutions compared against human ingenuity/intuition. The human-designed controller was not a neurocontroller, since hand-designing neural nets is not an area where humans excel, but rather an algorithm (designed by undergraduate research assistants) that used the same 12 distance measurements as input and produced a movement angle as output. Specifically, the human-designed algorithm was to move forward as long as there was at least 0.5 meters clear in front of it, and otherwise it would choose to turn to face the direction that offered the farthest clear line-of-sight. The preference for moving forward was based on the intuition that it is beneficial to cover as much ground as possible (as opposed to a random walk which diffuses slowly through the space), while also attempting to fill in open spaces. These four controllers were run in simulation for 1000 trials with different random target layouts; the performance results are shown in the left panel of Figure 4.

The key observations (which are all statistically significant at $p < 0.01$) are as follows:

1. The evolved and human-designed controllers substantially outperform the unevolved controller.
2. We confirmed that the neurocontroller from objective-based search slightly outperforms that from novelty search on the MAMTSS task.
3. Both the evolved controllers outperform the human-designed controller.

This last point underscores the effectiveness of neuroevolution in this domain. However, in fairness to the humans, we note that the evolved algorithms are very likely exploiting the fixed geometry of the course (e.g. by preferring left turns over right turns), whereas the human-designed algorithm did not attempt to exploit this geometric bias. To better visualize the collective motion of the robot swarm, Figure 5 shows the probability of each location (discretized at 10 cm resolution) being explored by a robot after 2 and 8 minutes. The slightly lower performance by novelty search may stem from the robots staying a little further away from the walls than with the objective-based controller.

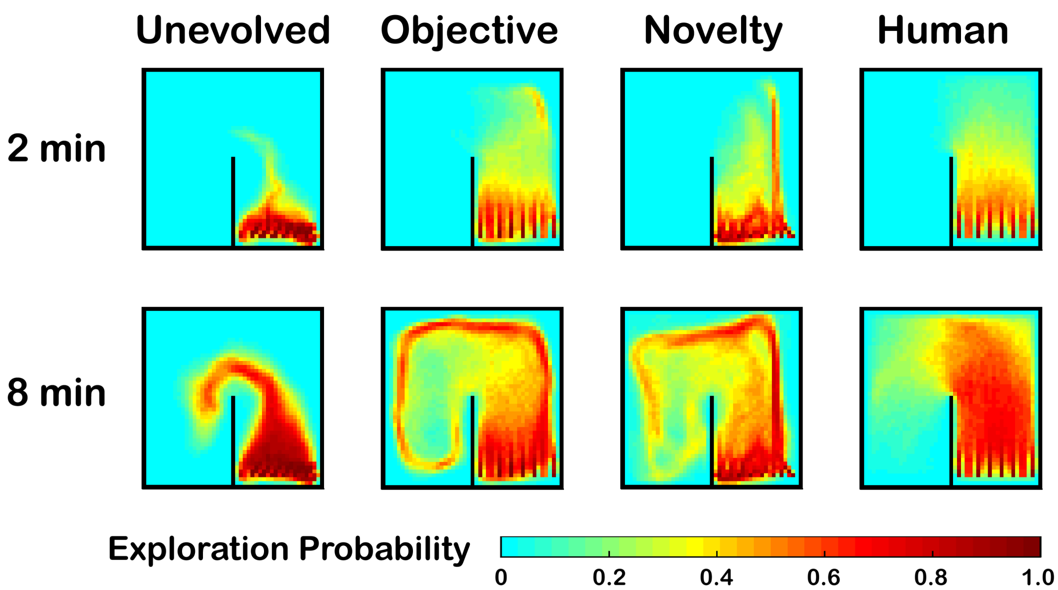

Figure 5: Heat maps of the locations likely to be explored by the various controllers, measured as the empirical probability of robot traversal during 1000 trials. The evolved neurocontrollers (objective & novelty) reliably explore both halves of the environment, while the human-designed controller achieves excellent coverage of the starting half, but rarely makes much progress around the corner.

4.2 Physical Robot Results

The next test was how our evolved controllers would perform in the real/physical robotics MAMTSS task, compared with the software simulation. For this experiment, to reduce noisy performance values based on target placement, we chose just one (fixed) random target layout (the one shown in Figure 2). We transferred the same four neurocontrollers discussed above onto the LEGO EV3 robots, and performed 32 8-minute trials for each controller on this layout, recording the number of targets located after each trial. We also ran the software simulation with this specific layout 100 times. The results of this experiment are shown in the right panel of Figure 4. The physical results correlated strongly with the simulated results for this layout, offering evidence that the software simulator provides sufficient verisimilitude. The physical results also matched the same rank order found in the simulation results across random target layouts. The objective controller outperformed the novelty controller on the physical task (t-test, $p < 0.03$), and the novelty controller appeared to outperform the human-designed controller, but this comparison lacks statistical significance due to the high variance of the number of targets found across trials. The slightly lower performance on the physical task vs. the simulated task may be due in part to the fact that occasionally robots would get jammed or ensnared with other robot chassis, or even knocked over by another robot, causing it to be disabled for the remainder of the trial – contingencies not included in our simulation software.

4.3 Further Observations

We decided to look more closely at the actual neural networks that were evolved, and were surprised to discover that the best-performing evolved neurocontroller from the objective-based search included just one synapse, meaning that the robot was computing an angle to turn based on only *one* of the 12 distance sensor readings. The best-performing controller found by novelty search was also relatively simple, employing just three synapses. Whereas prior research found that novelty search provided better performance and "the ANN controllers from the maze domain and the biped walking task discovered by NEAT with novelty search contain about three times fewer connections than those discovered by objective-based NEAT" (Lehman and Stanley, 2011), we found the opposite. For the MAMTSS task, objective-based search slightly outperformed novelty search, and the best performing ANN from novelty search had three times *more* connections than the objective-based controller. The multiplicative ratio overstates the case here, since 3 synapses versus 1 synapse is a small absolute difference, and may not be significant.

To determine whether we had allowed NEAT long enough to evolve more complex neural structures, we plotted the number of synapses in the best neural nets

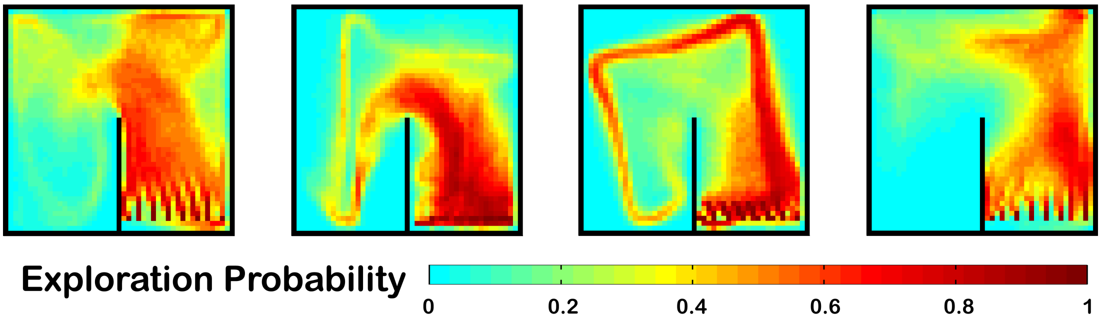

Figure 6: Maps showing empirical probabilities of robot traversal during 1000 trials, for four of the more complex best-performing neurocontrollers, taken from the final generations of the novelty searches.

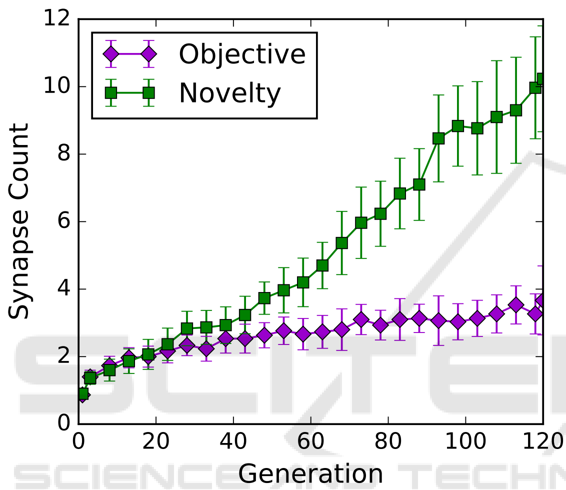

Figure 7: Neural complexity (measured by number of synapses) of the best individuals as the searches progressed. Error bars show 95% confidence intervals for the mean.

from each generation, as shown in Figure 7. This demonstrated that there was time for more complex neural networks to evolve, and that novelty search, in its quest for new behaviors, was evolving them. However, more complex neural nets did not tend to lead to better performance on MAMTSS, which would explain why novelty search's performance began to degrade in later generations, as well as why the objective-based search was avoiding evolving more complex networks. The collective motion behavior for a few of the more complex (15-19 synapses) networks taken from a final generation of novelty search are displayed in Figure 6. Although these heat maps collapse robot positions over time, a variety of qualitatively different swarm movement behaviors are still quite evident, in accordance with the modus operandi for novelty search.

5 CONCLUSIONS AND FUTURE WORK

Upon reflection, novelty search was working quite well – it was successfully evolving a range of complex/interesting swarm motion behaviors. However, high performance solutions to the posed MAMTSS task were possible with simple neurocontrollers. The speed with which objective-based search was able to converge on these suggests that the problem was relatively easy (although we had no *a priori* reason to suspect this would be the case), and that this fitness landscape is mostly non-deceptive. Our finding that objective-based search (slightly) outperformed novelty search on this task is in accord with an earlier project (Gomes et al., 2013) that found objective-based search was superior to novelty search for their simulated swarm aggregation task, which they also attributed to the fitness function not being particularly deceptive.

For future work, it would be interesting to explore more complex boundary geometries, as well other variants of the MAMTSS, with improved robot sensing capabilities or some form of limited communication allowed between robots. Would such variants offer additional challenge and/or deceptive search spaces where novelty search would outperform objective-based search? Furthermore, narrow passageways can form bottleneck difficulties for robot swarms, and it would be interesting to compare evolved solutions against recent approaches such as fear modeling (Konarski et al., 2016). It would also be informative to perform a scaling analysis on the size of the task. Would evolution using a large swarm of robots in a large environment produce qualitatively different behaviors than those we evolved for a small swarm in a small space?

To summarize, in this paper we have:

1. defined a new swarm robotics task (MAMTSS)
2. solved MAMTSS using neuro-evolution, with both novelty and objective-based search yielding better than human-designed performance
3. tested these neurocontrollers and showed verisimilitude between the simulation and the physical robots.
4. characterized the resulting swarm behavior for various neurocontrollers

Even though our formulation of the MAMTSS robot exploration task turned out to be simpler than anticipated, this study still provides one more data point that explores the relative trade-offs between novelty and objective-based search within the domain of neuroevolution for swarm robotics.

ACKNOWLEDGEMENTS

We thank Augustana College for its support of this project through internal research grants.

REFERENCES

Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41.

Choset, H. (2001). Coverage for robotics – a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1):113–126.

Coleman, O. (2012). AHNI (Another HyperNEAT Implementation). [Computer Software].

Davids, A. (2002). Urban search and rescue robots: from tragedy to technology. *IEEE Intelligent Systems*, 17(2):81–83.

Gomes, J., Urbano, P., and Christensen, A. L. (2013). Evolution of swarm robotics systems with novelty search. *Swarm Intelligence*, 7(2-3):115–144.

Ichikawa, S. and Hara, F. (1999). Characteristics of object-searching and object-fetching behaviors of multi-robot system using local communication. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 4, pages 775–781 vol.4.

Konarski, M., Szominski, S., and Turek, W. (2016). Mobile robot coordination using fear modeling algorithm. *International Journal of Mechanical Engineering and Robotics Research*, 5(2):96.

Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223.

Lund, H. H. (2003). Co-evolving control and morphology with LEGO robots. In *Morpho-functional Machines: The New Species*, pages 59–79. Springer.

McLurkin, J. and Smith, J. (2007). *Distributed Autonomous Robotic Systems 6*, chapter Distributed Algorithms for Dispersion in Indoor Environments Using a Swarm of Autonomous Mobile Robots, pages 399–408. Springer Japan, Tokyo.

Parker, G. B. and Georgescu, R. A. (2005). Using cyclic genetic algorithms to evolve multi-loop control programs. In *Mechatronics and Automation, 2005 IEEE International Conference*, volume 1, pages 113–118. IEEE.

Rekleitis, I., Lee-Shue, V., New, A. P., and Choset, H. (2004). Limited communication, multi-robot team based coverage. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, volume 4, pages 3462–3468. IEEE.

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, New York, NY, USA. ACM.

Risi, S., Vanderbleek, S. D., Hughes, C. E., and Stanley, K. O. (2009). How novelty search escapes the deceptive trap of learning to learn. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, GECCO '09, pages 153–160, New York, NY, USA. ACM.

Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127.

Wagner, I. A., Altshuler, Y., Yanovski, V., and Bruckstein, A. M. (2008). Cooperative cleaners: A study in ant robotics. *The International Journal of Robotics Research*, 27(1):127–151.

Wilensky, U. (1999). Netlogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.