Fitness, environment and input: Evolved robotic shepherding

M. Henry Linder & Ben Nye

May 14, 2010

Abstract

A simple shepherding task is used as a framework in which to investigate the role played by fitness functions, environmental design and inputs to a neural network in the evolution of robotic behavior. While a cursory approach to this experimental problem might suggest that the only factor relevant to success is the function used to gauge a robotic brain's fitness, experimental results indicate that the structure of the robot's environment, as well as the sensory input provided to the robot about both the physical environment as well as task-specific environmental features, play an equally important role in a robots success (or unsuccess). Furthermore, previous research indicates that the fitness function used must be designed carefully, in order to encourage further development of desired behavior by properly rewarding successful behavior and emphasizing relevant measures of success; it is essential to give constructive rewards to effectively indicate the underlying task, which can be difficult given a human interpretation of the domain under consideration. Furthermore, previous experiments involving robotic shepherding indicate that the evolution of desired behavior is not only possible but nearly trivial; in this experiment, we examine why our previous experiments in this field met with unsuccess, and discuss how our updated experimental method induced success.

1 Introduction

Our research is an examination of the roles played by fitness function, environment and sensory input in evolving a robotic brain that exhibits a desired behavior. The task used as a framework for this investigation is a simple, simulated robotic shepherding task, in which a shepherd robot brain is evolved to herd a sheep towards a target location. This task uses environment-manipulation as the basis for rewarding fitness, using inputs describing the shepherd's location relative to the sheep and target to teach the shepherd to control the sheep's location. Finally, over the course of our experiment, we found ourselves interested in the notion of domain simplification, and whether, in the search for a setup sufficiently simple and clear for the development of the desired behavior (by removing extraneous noise, particularly in environmental design and the sensory input provided to the evolving brains), it is possible to oversimplify the problem in a way that stunts the evolved brains' ability to generalize their behavior beyond the initial scope of their evolutionary environment.

Our research uses an implimentation of NeuroEvolution of Augmenting Technologies (NEAT) to evolve the brains that control the robotic shepherd. NEAT applies evolutionary concepts to neural networks, sets of interconnected nodes that mimic biological neural structures. NEAT initializes a neural network with minimal topological structure, "starting out with a *uniform* population of networks with zero hidden nodes (i.e., all inputs connect directly to outputs)."[4] From this, the NEAT algorithm complexifies the neural network by adjusting the connection weights between nodes and adding intermediary hidden nodes in order to find an effective solution to a problem. A fundamental principle of NEAT is the notion that "evolving structure along with connection weights can significantly enhance the performance [neuroevolution]."[4] Because of the emphasis NEAT places on a ground-up, holistic evolutionary approach to both the neural network's topology as well as the numeric connection weights between nodes, NEAT is an effective tool for evolving robotic shepherd brains:

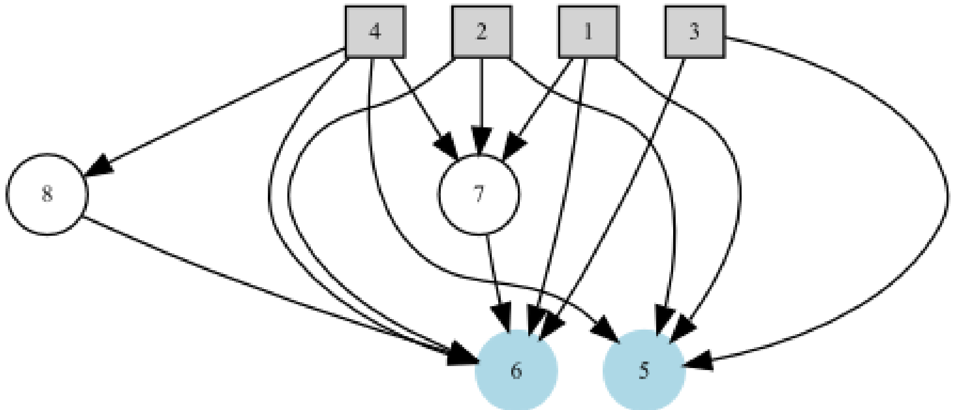

Figure 1: Topological structure of an evolved shepherd brain

an overly-simple brain is evolved into an effective neural control structure based on trial and error through the random weights NEAT assigns upon intialization, and removes any potential human bias towards what might constitute an effective topology. Figure 1 shows the topology of a neural network evolved by NEAT that displays tangible shepherding behavior, in this case after 48 generations. The top four nodes, numbered 1-4, represent the sensory input nodes; the bottom two, 5 and 6, represent the motor outputs; and 7 and 8 are hidden nodes. It is interesting to note that, despite the seemingly complex nature of even this simple task, namely extrapolating an effective strategy to obtain high fitness based on a small set of basic inputs (one that, as will be discussed later, generalizes beyond the initial environment in which this brain was evolved), the neural network's topology remains minimal. This also demonstrates the benefit of NEAT in not over-complexifying, instead finding an effective strategy that may nevertheless be limited in its complexity.

Preceding our research are two studies that examined the evolution of robotic herding behavior: the first in 1996, by Schultz et al.[3], the second in 2001, by Potter et al.[2], the latter of which expanded upon the research of the former.

The former research[3] represents an early forray into evolutionary robotics applied to a complex problem. The task is very similar to ours: a robotic shepherd is provided with sensors indicating the ranges and bearings to the target and the sheep, as well as the sheep's heading with respect to the bearing to the sheep. This research differs from ours in multiple ways: first, it was performed using real robots, necessitating the presence of a sensor indicating the sheep's heading, a sensor we found extraneous due to the simulated and simple nature of our sheep's behavior. Second, the robots did not use neural networks, but instead developed rule-based behavior evolved by a genetic algorithm. Third, we expand upon this experiment by diversifying the types of inputs provided to the shepherd, as well as situating the robots in a variety of experimental environments.

The latter research[2] had as its primary focus the evaluation of the relative success of homogenous and heterogenous brains in cooperative robots, examined within the context of robotic shepherding. However, this research is primarily relevant to that presented in this paper because of its examination and recreation of the research presented in [3]. This research utilized feed-forward neural networks, as with our research, though the topology of the network was designed by hand. Additionally, the sensory inputs consisted of a range and bearing to the sheep, with all angles relative to the sheep's angle to the target. This research went beyond a single, evolving shepherd maniuplating a sheep that "reacts to the presence of nearby objects by moving away from them. Otherwise, the sheep moves in a random walk."[3] Instead, sheep were also given motivation to avoid the target corral and to seek to exit the environment.

These two predecessors found that evolutionary robotics is not only capable of developing robotic shepherds, but was very successful at it. Particularly, [2] found that, while recreating the basic task of [3] involving a single sheep with obstacle-avoidance and a single shepherd, "Although these control systems were evolved for 50 generations...a controller with near optimal behavior on this simple task was easily evolved in as little as 2 generations."[2] The success of these experiments suggests that a NEAT-based approach will be similarly successful; furthermore, the research of [2] suggests that, when using neural networks to evolve behavior, the task may be trivial to solve. We suspect that, given the ground-up complexification of NEAT, shepherding behavior will arise within a simliarly short time-frame in our experiment.

However, our own exposure to this research has precedent as well[1], in which we sought to evolve robotic shepherds using NEAT—the present research is essentially a contination of our initial experiment. However, in spite of the above conjecture that robotic shepherding would evolve easily with NEAT, our initial experiments in our first paper indicated otherwise: across a variety of fitness functions and simplifications of the sheep's behavior, the only success that was observed arose in a very basic setting. Thus the conclusion that "our difficulty in obtaining acceptably successful herding behavior must be attributed to some combination of inputs, fitness function, and the environment."[1] Thus our present focus on these three factors in our expansion and continuation of this initial experiment.

2 Experimental Method

Because we began our research with the benefit of a prior experiment to learn from, our approach has been focused on refining the structure of our experiment and understanding the ways in which the experimental variables impact the evolution of the desired shepherding behavior, primarily with respect to three different components: the environment; the sensory inputs; and the fitness function. Of additional albeit secondary importance are the parameters provided to NEAT in order to evolve the neural network brains, specifically the size of the population under evolution and the number of steps used in each trial. Finally, we discovered in both our prior and present experiments that these three primary factors are neither isolated in their specification nor independent of the other factors; in reality, they depend heavily upon each other.

In both experiments, the robots were Pioneer robots simulated in the Pyrobot robotics simulator. In our initial experiment, the robots were situated in an environment confined by a 10x10 bounding box; the target was located in lower left corner, the sheep in the direct center, and the shepherd's location was randomized in the upper 40% of the environment. The random shepherd location and bearing was intended to prevent location-specific behavior, and to induce generalized strategies. The sheep, which had a hard-coded brain, was equipped with three behaviors; in order of precendence: wall-avoidance; shepherd-avoidance; and a wandering pattern (in later trials, the wandering was disabled, in order to reduce the distractions and make it easier for the shepherd to manipulate the sheep's position; it was with wandering disabled that the only successful shepherding behavior arose, although it is interesting to note that this shepherding behavior was generalized to the extent that the successful shepherd trained without wandering could also herd a wandering sheep). In our present experiment, we used the same hard-coded sheep brain, with wandering disabled. The sheep is programmed to flee in the direction opposite that of the approaching shepherd when the shepherd comes within an unacceptable radius of the sheep; the sheep's heading is situated properly instantaneously away from the shepherd, obviating the need for a sensor indicating which direction the sheep is facing. Also disabled was wall-avoidance, for reasons that are discussed below; thus, the sheep's behavior consists only of shepherd-avoidance.

In our pervious experiment, we hoped to not only evolve shepherding behavior, but along the way, we hoped to evolve wall-avoidance behavior, as well. To this end, the sensors given to NEAT were the ranges and bearings from the shepherd to the sheep and target, as well as the minimum of both the front-right and front-left sonar sensors; outputs consisted of translate and rotate values, which was later changed to direct control of the left and right motor outputs. Trials were ended prematurely if the shepherd stalled against a wall.

We have identified the sensors as a crucial obstacle in development of shepherding behavior. For the most part, successful shepherding does not in any way involve wall-avoidance, as the most efficient strategy is for the shepherd to situate itself behind the sheep such that shepherd, sheep, and target are colinear. However, 33% of the inputs to the neural network consisted of sonar sensors, which did not contribute to the robot's knowledge of how to achieve high fitness. Thus, we realized that the sonar sensors, in all likelihood, were a distraction, as sonar sensors could differ wildly without any impact on fitness. As wall-avoidance is not a primarily-desired behavior in evolution, we made the decision in the present experiment to strip the robots of their sonar sensors. But stalling was still of concern, because robots that explored the environment (that is, did not just spin in place) were inadvertantly punished by shortened trials if they ran into a wall. So, the decision was also made to remove the bounding box in the environment.

Thus, in our present experiment, the environment consists of an infinite plane. Trials are ended early if the sheep strays further than 20 units from the target, or if the shepherd strays further than 20 units from the sheep. Thus, wall-avoidance in the sheep was unnecessary, as were the distracting sonar sensors provided to NEAT. There was no discernable difference between shepherds with outputs of translate/rotate and left/right motor outputs, so the outputs in our experiment were restricted to left/right motor outputs for the sake of uniformity.

However, given the difference in inputs between the research of [3] and [2], we sought to experiment with both types of inputs, yielding two distinct sets of inputs. Input set 1 consists of bearings and ranges from the shepherd to the sheep and target, with all angles relative to the shepherd's heading; input set 2 consists of the bearing and range to the sheep, the sheep's distance from the target, and the angle between the shepherd and the vector between the sheep and the target. This fourth input in set 2 indicates the angle between the shepherd's current heading and the optimal heading for shepherding. That is, when the shepherd is directly behind the sheep such that sheep, shepherd and target are colinear, this input gives an angle of 0. We also adjusted angles to be measured not based directly off the shepherd's heading, because this had the potential for confusion, as well: while straight ahead gave a value of 0, an angle of 364° would give a value near 1, when in fact the difference between the two angles is negligible. Therefore, angles are given by values between -1 and 1, with 0 corresponding to the shepherd's heading.

We further devised three different environments in which to evolve shepherds, with predicted increases in difficulty. The reasoning behind such differences in environment is to examine the way in which environmental complexity impacts evolution. In the “easy” environment, the target is located at a point (0,0), the sheep at (5,5), and the shepherd at (10,10). This environment situates the sheep, shepherd and target colinearly from the beginning, causing the shepherd to only need to move directly towards to target to successfully shepherd. In the “medium” environment, the only change is that the sheep's placement is randomized between either (0,10) or (10,0), requiring the shepherd to maneuver around the sheep before directing it towards the target. Finally, the “hard” environment randomizes the location of both shepherd and sheep, with one at (0,10) and the other at (10,0), a more difficult version of the medium environment, such that the shepherd must make a wide arc around the sheep before it can herd it to the target. In all of these, the shepherd's initial heading is randomized, to promote generalization.

Finally, we have used two fitness functions in measuring shepherding success. In our first experiment, the fitness functions used were intended to strike a balance between continuous fitness—which rewards the behavior of the robot across the entire trial, instead of awarding fitness based solely on final conditions, which may not accurately reflect its overall ability to shepherd—and discrete fitness—which, in a task such as shepherding, is significant because the desired behavior is explicitly to manipulate the sheep's location to a specific end-condition. Strictly continuous fitness—e.g., based on the cumulative distance of the sheep from the target—was unsuccessful because it improperly rewarded individuals that spun in place, while penalizing those that explored the environment (the first step towards herding behavior), because the former accrued fitness despite exhibiting no desired behavior, while the latter often had their trials ended prematurely due to stalling, thus stunting the number of steps for which fitness was awarded. Successful behavior arose with a purely discrete fitness function, based on the sheep's ending distance from the target; because of the bounding box, and the low number of steps in each trial, it was unlikely that a shepherd would accidentally move the sheep away from the target after moving it towards it, so the lack of continuity was not a distinct problem.

With the present experiment, our initial fitness function stored the minimum distance the sheep ever came within the target, and based fitness off the inverse of this; trials were ended early if the sheep came within 1 unit of the target, and full fitness was awarded. This allowed for a continuous component—the sheep's overall closest distance to the target—while at the same time preserving the discrete nature of the task, namely the distance of the sheep from the target. However, we realized that small differences in this minimum distance did not significantly impact the amount of fitness awarded, despite the fact that even small decreases were highly relevant. As a result, the second fitness function we designed was an exponential fitness function, where fitness is calculated by $2^{start-minimum}$ where "start" is equal to the starting distance and "minimum" is equal to the sheep's minimum distance from the target over the course of the trial; as with the first fitness function, full fitness is awarded and the trial is ended prematurely if the sheep comes within 1 unit of the target.

Each neural network brain is tested for 3 trials, and brains are evolved for 50 generations. Initially, the number of steps in each trial was 200, and the size of each population was 20. However, early success was had with input set 1 in the easy environment (shown in figure 1), and this behavior was generalized to the extent that these successful brains were also successful in the medium and hard environments. However, even these fully evolved brains often required more than the default 200 steps to complete the shepherding task (completion defined as manipulating the sheep to within 1 unit of the target), suggesting that if successful brains are to be evolved from scratch in the medium or hard environment, they will also need a larger step-count. Additionally, we suspect that a larger population size could increase the likelihood of successful behavior arising. As a result, further runs were conducted in the medium and hard environments with 400 steps in each trial and population sizes of 35.

Finally, given the successful shepherding behavior that arose quickly in the easy environment, we decided to try to oversimplify the environment to see if it was possible to evolve brains that were successful only in their evolutionary environment. To accomplish this, we also conducted an evolution in the easy environment where the shepherd's heading is always directed towards the target. Theoretically, then, the shepherd will only need to learn to drive straight in order to receive full fitness.

3 Results

As alluded to above, successful shepherding behavior arose with input set 1 in the easy environment, with the fitness function based on the inverse of the sheep's minimum distance from the target. However, this was the only success in the initial set of runs, with no successful shepherding displayed by set 1 in the medium or hard environments, or by any brains equipped with input set 2. The successful brain was further able to scale to the medium and hard environments and successfully shepherd there, as well. Consistent, though rudimentary, shepherding behavior arose after 7 generations, and optimal behavior arose after 11 generations. This was the closest our results came to the trivial results of

[2]. However, though initially mystifying, the reason for this seems to be straightforward: whereas [2] evolved neural networks with fixed and intentionally designed topology, which incorporated hidden nodes from the beginning, NEAT begins with the simplest possible neural network, devoid of any hidden nodes and randomized connection weights between the input and output nodes. It is therefore to be expected that optimal shepherding behavior took more than 2 generations to emerge, because the neural networks needed additional generations to complexify to a point where proper behavior was possible. Thus, it is impressive that successful shepherding occurred after only 7 generations, and optimal behavior after only 11, because the neural networks were initialized with nothing. Additionally, it is interesting to note that even after 48 generations, shown in figure 1, the network only has 2 hidden nodes, where even in the simple recreation of [3] by [2], 3 hidden nodes were used.

Of further interest in comparison with the research of [2] is the potential reason for the uniform failure of input set 2 across all runs—no shepherding behavior arose in any of the 3 environments, nor did the change in fitness function have any impact on the success of the second set of inputs. Upon closer examination of the paper on the reasearch of [2], it was noticed that, in their recreation of the task that mirrors ours, the neural network featured only 3 input nodes, where our input set 2, meant to mimic their inputs, consisted of 4 inputs: the additional input is the distance between the sheep and the target. Also, we measured the angle to the vector between the target and the sheep, while [2] measured the angle to the same vector as it extended behind the sheep; however, this difference of 180° seems insignificant and arbitrary. Similarly, the additional input does not seem to be problematic, at least on principle; at most, it might make the task minorly more complex to learn, though this should not take longer than the 50 generations that we ran, given our own success after only 7 generations with input set 1. Finally, however, it is possible that the explanation for the total lack of success had by input set 2 was a result of insufficient complexity: [2] initialized a neural network with 3 hidden nodes to handle only 3 input nodes, while in our experiment, complexity maximized with at most 2 hidden nodes to account for 4 input nodes. Therefore, it is still plausible, albeit potentially far-fetched, that a significant increase in generations would have yielded more success.

The exponential fitness function did not have a significant impact on performance; in fact, when compared with the success found with the initial fitness function in the easy environment, the exponential fitness function resulted in less successful shepherding. However, it is unclear whether this is a fundamental principle defining the relative success of both fitness functions, or whether this disparity was incidental to the limited number of runs performed.

The brains evolved in the medium and hard environments with larger population sizes and higher step counts, using input set 1 and both fitness functions, produced mixed results: on the one hand, fitness was tangibly higher (on average) than that found in runs with shorter trials, and occasionally, identifiable shepherding behavior was witnessed. However, emphasis must be placed on the occasional nature of this behavior; it was far from consistent, and must be viewed as limited success at best. This suggests that further, additional tweaks to the experimental parameters are necessary in order for shepherding behavior to arise in the medium and hard environments.

Finally, the brains intended to be under-generalized behaved as predicted: under evaluation by both fitness functions, the shepherds developed effective shepherding, but only in the exact environment they were evolved in. Shepherd's under-generalized to the point of lacking generalization at all. Brains were evolved that simply forged straight ahead, and even minor shifts in the location of the sheep from its initial colinear placement caused the shepherd to falter in its ability to complete the task.

4 Conclusion

Based on our experimental results, it is possible to draw conclusions about the roles played by fitness function, sensory input, and environmental design, as well as the possibility for minor changes in these to drastically change the nature of the task at hand. This last component—under-generalization—is perhaps the simplest to summarize: as witnessed with the under-generalized shepherd evolved in the easy environment, a simple environmental variable, namely disabling a random initial heading for the shepherd, results in behavior that is successful only when restricted to the learning environment. That is, by lessening the diversity of situations to which the learning robot is exposed, the brains evolved address a fundamentally different task from their more generalized counterparts. Whereas the latter explore the environment by default and necessity, given the statistical infrequency of a "perfect" heading allowing for strictly straight movement to yield perfect fitness, the former are *only* exposed to this "ideal" circumstance. As a result, the task's nature shifts from one where the shepherd must maneuver around the environment, in the process often causing the sheep to move, to a task in which the shepherd is taught only to move in a straight line. Thus, this domain simplification stunts the exploration and learning of the shepherd. It is particularly interesting and compelling how drastically the learned behavior changes with such a seemingly minor change: while the most successful shepherding brains could adapt to more difficult environments, the under-generalized shepherds could not even account for minor changes in their environment, given the limited scope of their learned behavior.

The nature of the fitness function is quite important to the development of the desired behavior, as was witnessed at the end of our previous research [1], and which was further reinforced in the present experiment. A careful balance must be struck: in our experiment, it was necessary to retain the emphasis on end-conditions in the environment, specifically the proximity of the sheep to the target, in evaluating the fitness, while at the same time accounting for the possibility that mistakes will occur and that taking into account *only* the end conditions may improperly interpret a given brain's behavior. The tentative sub-optimal assessment provided by the exponential fitness function suggests that, while the fitness function based on the inverse of the sheep's minimum distance from the target may undervalue initial exploration and subtle improvements in behavior, it is possible that the exponential fitness function may overvalue such exploration. An individual with the above ideal heading that happens to move directly towards the sheep would be disproportionately rewarded for behavior that is not generalizable, a potential weakness in the function due to the potential to skew results. In addition, it is important, when designing a fitness function, to design a function that rewards the proper behavior: it is possible for a fitness function that seems promising to a human eye to have unintended side-effects in actuality, such as a continuous fitness function that improperly rewards individuals that spin in place. Thus, care must be taken when deciding how to award fitness, and the nature of the task must be accounted for: in the robotic shepherding domain, a discrete fitness function is most appropriate, where another task might benefit instead from a strictly continuous fitness function.

It is also immediately evident that the sensory input to the neural networks controlled by NEAT can have a heavy impact on learning ability. By reducing the number of sensory inputs from 6 to 4 between our first and second experiments, we witnessed an immediate change in evolutionary success. The removal of sonar sensors decreased the dimensionality of the task, making it easier for the robotic brains to develop the desired behavior by extrapolating the relationship between various inputs and the fitness received. However, it is also possible for the sensory input to be misguided; for still-unclear reasons, the second set of inputs to the neural network were not sufficiently lucid to allow for behavioral evolution. The problem may be similar to that confronted in our initial experiment, with sensors that seem logical to a human brain providing inadequate information to the brains in evolution. Therefore, as with the fitness function, care must be taken in understanding the relevant components of an experiment when deciding how to present the environment to the robot. Robotic development is similar to that of children, in that too much information can be problematic for learning; however, while nature limits the biological sensory apparati of children, it is up to the researcher to place boundaries on what the robot knows about his surroundings.

But the surroundings must be designed, as well, and it is here that the third piece of our refined research comes into play. Removing sonar sensors but retaining a bounding box would most likely result in behavior similar to that found in our previous research, at least initially, wherein robots that engage in significant exploration would be discouraged. So when designing the sensory apparati for a robot, it is of equal importance to ensure that the environment and the sensory input are reflective of each other. It would do a robotic shepherd no good, for instance, if it was contained within a hallway from which it first had to extricate itself if it had no sensory input to alert it to the boundary's presence, and when such extra sensors would increase dimensionality and limit learning ability, the environment must be adjusted—and, often, simplified—in order to reflect the necessary simplifications in sensory inputs. Diversity of environments is important, too, in ensuring that robots are exposed to a sufficient variety of scenarios; this can be accomplished through random components that force the robotic brain to make generalizations, preventing environment-specific behavior.

Finally, it must be noted that, due to the success obtained in this experiment, NEAT is an efficient way to evolve all aspects of a neural network when learning a task. The complexity found in the topology in [2], while helpful in the short-term with respect to the speed with which optimal behavior arose, in the long run may be overkill. Our results show that the bottom-up approach of NEAT can lead to highly compact and efficient solutions that a human designer might not anticipate.

References

[1] M. Henry Linder and Ben Nye. Robotic shepherding using neat, 2010.

[2] Mitchell A. Potter, Lisa A. Meeden, and Alan C. Schultz. Heterogeneity in the coevolved behaviors of mobile robots: The emergence of specialists. In Morgan Kauffman, editor, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001.

[3] Alan C. Schultz, John J. Grefenstette, and William Adams. Robo-shepherd: Learning complex robotic behaviors. *Robotics and Manufacturing: Recent Trends in Research and Applications*, 6, 1996.

[4] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10, 2002.