Exploring MultiObjective Fitness Functions and Compositions of Different Fitness Functions in rtNEAT

KJ Bredder and Cole Harbeck

Abstract

In machine learning, the fitness function is an incredibly important part of defining and learning how to complete a task. For complex tasks, it is possible to identify a set of sub-goal that can be explicitly rewarded as part of the fitness function. However, there are several different ways of incorporating these sub-goals into evolution. The fitness function can be a weighted sum of the different sub-goals or several separate populations can be evolved as specialists in one sub-goal. These specialists can then be combined or crossed over after evolution is complete to create a composite population or controller designed to complete the complex task. The purpose of this experiment was to explore the effects of using one multi-objective fitness function rather than a composition of individuals evolved using different fitness functions in NeuroEvolving Robotic Operatives (NERO) with real-time NeuroEvolution of Augmenting Topologies (rtNEAT). Two teams of robots were evolved within the NERO game, one evaluated using a multi-objective fitness function and the other a composite of three separate specialist populations each evolved as an expert in a single objective. The multi-objective team was pitted against three separate specialist teams and the composite team. Our hypothesis was that the population evolved using a multi-objective fitness function would be more effective in battle than the composite team, but results showed that the composite team was more effective in battle than the multi-objective team. However, the multi-objective team was more effective than most of the specialist teams on their own. We took this a a promising sign that decomposition of a multi-objective fitness function could be applicable to complex tasks.

1 Introduction

When designing and evolving robot controllers to accomplish difficult tasks, there are often several different sub-tasks. More complicated fitness functions, such as multi-objective functions, allow individuals to be evaluated based on several different parameters simultaneously. This helps negate some of the pitfalls associated with fitness-based search, such as sub-optimal local maxima in the fitness landscape that cause premature stagnation. It also allows for multiple factors to be included in fitness, some potentially more complex than others. However, the calculation of multi-objective fitness evaluations can be computationally costly. The use of ANNs in calculating approximate fitness functions has been a topic of discussion in recent years [1]. Working from the premise that calculating a single-objective fitness function may be less computationally costly than calculating a multi-objective fitness function (due to its parallelizability), the goal of this experiment is to evaluate the performance of a population of individuals generated using a single multi-objective fitness function compared to that of a population formed from a composite of individuals evolved using different fitness functions. We were able to find an example of a study of the decomposition of multi-objective fitness functions. The MOEA/D method separates the fitness function into separate fitness functions, and optimizes each of these separately at each step. The individual optimizations are then combined into one fitness function at each step. This procedure was applied to the knapsack problem, and was shown to be a viable computational alternative to direct computation of the multi-objective fitness [3].

To explore this problem, we will be using the rtNEAT system developed for NeuroEvolving Robotic Operatives (NERO) by Kenneth O Stanley and others [2]. In the NERO game, the task is to evolve a population of individuals and form a team of these individuals that will win a match against another team by killing them off. This task is well suited for the usage of a multi-objective fitness function as several different skills are incorporated in winning battle strategies. The rtNEAT system replaces individuals in the population in real time, allowing for the kind of continuity necessary for evolution in video games. NERO uses rtNEAT as its evolution protocol, replacing individuals in real time. It cycles through the sequence of steps outlined in [2]:

1. Remove worst agent based on adjusted fitness (fitness/species size)
2. Readjust average fitness
3. Choose parents
4. Adjust compatibility threshold and reassign species
5. Replace old brain with new brain

The time between replacements is determined by a parameter I, the ineligible fraction of the population, which is set here to be 50 percent. An individual is considered ineligible for replacement until it reaches a given age, allowing for fair evaluations. Each baby is set down into the world at a fixed "factory" location, again to promote fair evaluation. Each individual in NERO is given five enemy radar sensors that divide the 360° around it into slices which each activate in proportion to the nearest enemy, an on target sensor which is activated if the individual is targeting an enemy, five range sensors, two enemy line of fire sensors which detect where the bullet stream from the nearest enemy is heading, and one bias input. There are three output values for left/right movement, forward/backwards movement, and a fire output (Figure 1).

We expect that the population evolved using a multi-objective fitness function will be more effective in battle than the population evolved as a composite of separate single-objective fitness function teams. This is because by using a multi-objective fitness function, the networks can be evolved to optimize all parameters of the function together, providing the ideal balance between all of the desired behaviors. In contrast, with the single-objective fitness function,the individuals are evolving toward only one parameter in ignorance of the others. We plan to test this hypothesis by forming five different teams of each type of evolution (multi-objective vs. composite of single-objective fitness functions) and having a tournament of sorts where every team will battle every other team (including multi-objective teams fighting other multi-objective teams and single-objective teams fighting other single-objective teams).

2 Experiments

We performed several different experiments. First, we performed an experiment looking at whether a particular specialist team was dominant over the others. Next, we performed a series of experiments pitting a multi-objective team against our composite of specialist teams. We ran this type of battle in two separate environments. We then evolved each of our populations for an additional hour, and pitted our long evolution multi-objective team against our long evolution composite team of specialists. All experiments were designed to test the efficacy of the multi-objective team against the composite team of single-objective specialists. We initially evolved 4 separate populations in NERO, each of 50 individuals. NERO has the capability to evolve for 6 different parameters at once: stick together, hit target, approach enemy, avoid fire, approach flag, and stand ground. We decided to study the problem of multi-objective fitness evaluations versus single-objective evaluations using the three objectives of approach enemy, hit target, and stick together. These parameters were chosen because they seemed to be distinct, non-contradictory goals. Initially, we had attempted to evolve individuals to avoid fire, but the most fit individual still had a fitness value of zero after an hour, so we decided to evolve a team designed to stick together instead. This turned out to be advantageous, as upon further reflection the avoid fire and approach enemy objectives are slightly contradictory. We believe this error was intrinsic to NERO, and not a result of our actions. Each team was allowed to evolve for an hour. The multi-objective population was evolved with the approach enemy, hit target, and stick together all designated as maximally important (i.e. the slider was set to positive 100%). The approach enemy population was evolved with the approach enemy slider set to 100%, the hit target population was evolved with the hit target slider set to 100%, and the stick together population was evolved with the stick together slider set to 100%. Each population was evolved in the sandbox environment with no walls or flags, with the population spawned from one end of the arena and a "better turret", a.k.a. a turret capable of rotating to face the nearest player, at the other end, seen in Figure 2. We chose this type of enemy as it was more complicated than the normal turret, but the troops were still capable of evading fire by shuttling behind the turret.

Out of curiosity, we decided to pit different specialist teams against the multiobjective team to determine whether a particular trait could be more or less important to overall fitness. This is obviously not an absolute measure of performance, as one team may have a strength that is particularly effective at exploiting the multi-objective teams weakness. However, we believed that this could give an indication of why the composite team may be more or less effective than the multi-objective team. If a specific goal was determined to be more or less effective, it may make sense to adjust the reward for each objective to reflect that. The multi-objective team was composed of the 18 most fit individuals from this population at the end of evolution. The composite team was formulated of the 6 most fit individuals of each of the specialist populations (approach enemy, hit target, and stick together). The teams were initially looked at in the standard sandbox environment. The battles here typically lasted less than 5 seconds, and didnt allow the teams to distinguish themselves in terms of performance. Thus the environment was deemed insufficient for analyzing the performance of these different teams, and we attempted battle in a more complicated environment that we designed, with a line of three walls in the middle of the field (Figure 3). As a measure intended to control for the impact of environment on team performance, the teams were then tested in the preset NERO "Andrian Arena" world (Figure 4). The populations were then evolved for an additional hour each, and were re-tested against each other both in the "Andrian Arena" and in the "Sandbox".

3 Results

In this experiment, we evaluated our five different populations of robots in two different ways. First, we examined the average and best individual fitness of each population and second, battled each team against the multi-objective team and evaluated the win percentage and average battle length for each team. The fitness data can be seen in Figure 5. Looking first at the fitness of each individual, the multi-objective team had both the highest average fitness and the most fit individual. This is because the multi-objective fitness function is a construction of all three single objective fitness functions and so can reach higher values more quickly than any of the single-parameter functions. The hit target specialists received the lowest average and individual fitness as this task was the hardest to learn. Not only must an individual learn to find the enemy, but it must also learn to aim. When compared with the stick together specialist population, it is clear that hitting an enemy is much more difficult than sticking together as the robots are all spawned very close to each other initially, but much further from the enemy. The composite team does not have any fitness data as it was formed from taking the six most fit members of each of the specialist populations and placing them all on the same team.

The first set of battles we ran were between the populations that had been evolved for one hour on our custom map (Figure 3). The results from these trials can be seen in Figure 6. The multi-objective team was able to defeat both the approach enemy specialists and the stick together specialists but did not win against the hit target specialists. The multi-objective team fared the best against the stick together specialists, winning 100% of their battles against them. While the average battle length of these battles was the longest, this makes sense because the stick together specialists do not approach the enemy team, but instead stay as a group where they are spawned. This however, makes it very easy for the multi-objective team to defeat this group as the stick together specialists have no training in hitting the opponent. The approach enemy specialists fared a bit better, winning 30% of their battles against the multi-objective team, however their lack of hit training was apparent again as their battles only lasted an average of 15.1 seconds. The hit target specialists were able to defeat the multi-objective team 80% of the time in an average of 17.5 seconds, reinforcing that this was the most important parameter towards the overall goal of winning. When the multi-objective team was battled against the composite team, the composite team won 100% of their battles but had the second longest average battle length only behind the multi-objective team against the stick together specialists.

Our second set of battles were done between the multi-objective team and the composite team only. The results can be seen in Figure 7. The first set of battles took place on the Andrian Arena map (Figure 4), a map with more obstacles. Of the ten battles, only five ended with a clear victor. The composite team won 60% of these battles with an average battle length of 58 seconds. Each team was then evolved for one more hour and battled again on the Andrian Arena map. After running several trials, none had ended in a victor so we decided to pit these teams against each other in the empty Sandbox map, exploring the possibility that longer evolution would lead to more complex battle tactics used by each team and battles would last longer than only several seconds. This proved to be true. Of the battles that concluded, the composite team won 100% with an average battle length of 87.3 seconds.

4 Discussion

From the battle lengths, we can draw several conclusions about the importance of the three objectives we chose, multi-objective versus single-objective speed of evolution, and specific outcomes of NERO. From the fitness graphs, we can see that certain tasks improved in fitness more quickly than others both in terms of average fitness and in terms of highest fitness. This lead us to believe that certain tasks, such as stick together, were easier to learn. We attributed this to the setup of NERO, which spawns all individuals from the same location. The hit target appeared to be the hardest task to learn but proved more effective in battle against the multi-objective team. This indicates that the hit target objective is more closely correlated to success against the multi-objective team. As a player of this video game, performing experiments of this kind could be useful in determining the weighting each objective should be allowed during evolution. Interestingly, during the battles the hit target team was important for the initial exchange of fire, but typically didnt survive the longest, and therefore were not necessarily the most integral to success. The stick together specialists tended to survive a lot longer, due to their propensity for lagging behind and thereby evading the bloodbath. Again, through analysis of the various behaviors a better evolution strategy could be devised for actual players of the game.

The single-objective specialists did have higher fitness values than the multi-objective individuals in their respective tasks when evolved for the same length of time. Additionally, we were able to parallelize the evolution of the specialist populations. This gives credence to the idea that being able to decompose a multi-objective fitness function is a desirable trait. The fact that the composite team handily beat the multi-objective team in combat in our initial world seems to indicate that this decomposition can definitely be useful in complex problems, such as battlefield combat.

As a probe for this problem, NERO is interesting because it allows for a team composed of individuals evolved to do different things, however these individuals cannot be crossed over. For a more direct comparison, the products of single-objective evolution should be combined into a single brain through crossing over, and compared to a team evolved with a multi-objective fitness function. Finding a different platform that allowed us to perform this kind of comparison would be an interesting addition to this analysis.

In conclusion, our results indicate that decomposition of a multi-objective fitness function could be an effective tool for faster evolution. Our composite team of specialists fared sufficiently well against the multi-objective fitness function, disproving our hypothesis. However, while we were able to parallelize evolution of different specialist teams, we were not capable of acheiving the kind of improved evolutionary speed-up that is the primary incentive for investigating this problem.

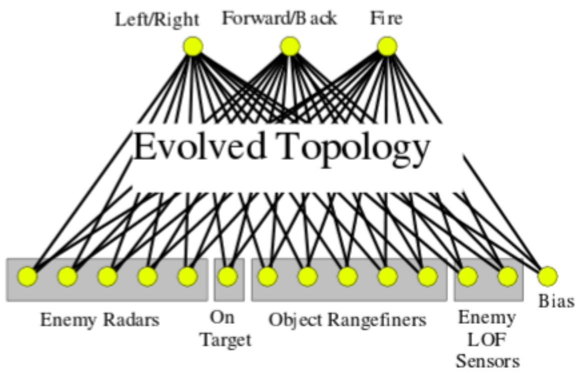

Figure 1: The inputs and outputs of the neural networks used for each individual in NERO.

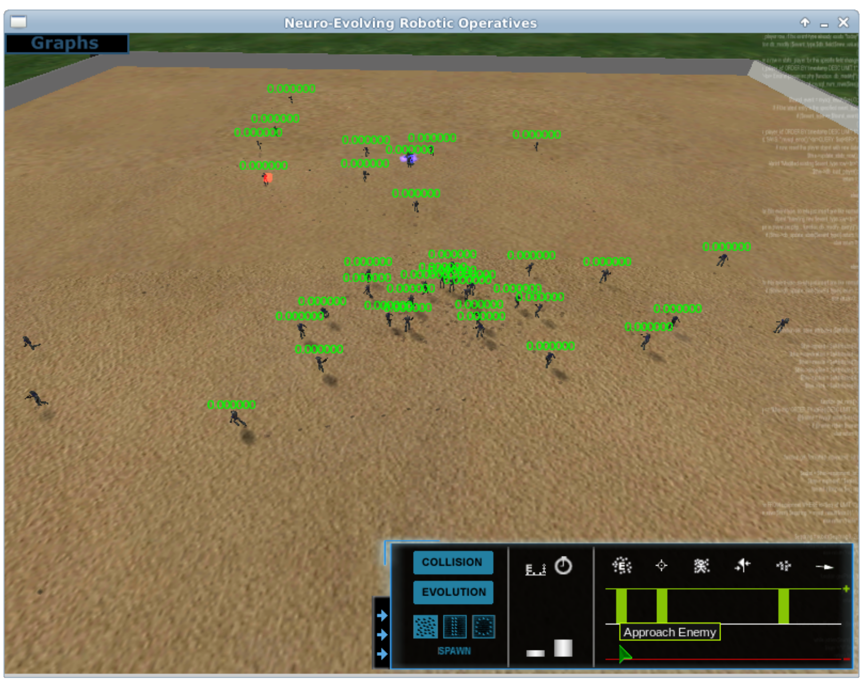

Figure 2: The training environment used for all training. It contains a better turret and no walls.

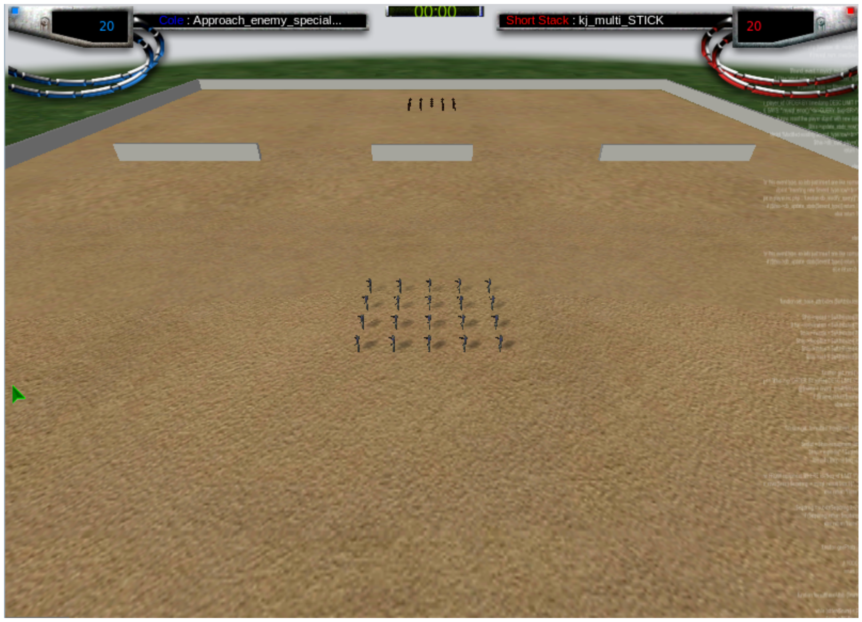

Figure 3: The custom map used for the first set of battles. There are three walls in the center of the map with each team spawning on opposite sides.

Figure 4: An overhead image of the more complex Andrian Arena map.

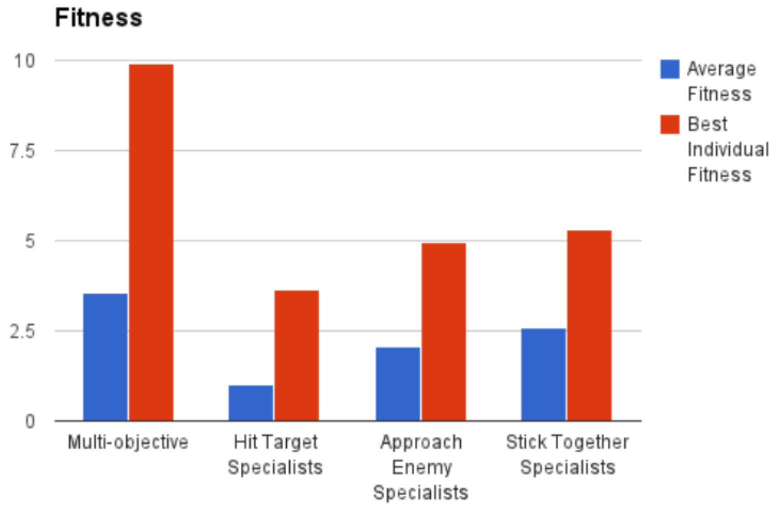

Figure 5: Average fitness and best individual fitness for each population after being evolved for one hour.

Team	Winner	Average Battle Length
Multi-objective vs. Composite	0% Multi 100% Composite	22.4 Seconds
Multi-objective vs. Hit Target Specialists	20% Multi 80% Hit	17.5 Seconds
Multi-objective vs. Approach Enemy Specialists	70% Multi 30% Approach	15.1 Seconds
Multi-objective vs. Stick Together Specialists	100% Multi 0% Together	26.3 Seconds

Figure 6: Win percentages and average battle length for each battle in the first round.

Map	Winner	Average Battle Length
Andrian Arena	40% Multi 60% Composite	58 Seconds
Sandbox Arena	0% Multi 100% Composite	87.3 Seconds

Figure 7: Win percentages and average battle length for each battle in the second round.

References

[1] A. Gaspar-Cunha and A. Vieira. A multi-objective evolutionary algorithm using neural networks to approximate fitness evaluations. *International Journal of Computers, Systems and Signals*, 6(1), 2005.

[2] Bobby Bryant Kenneth Stanley and Risto Miikkulainen. Evolving neural network agents in the nero video game. *Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games (CIG '05)*, 2005.

[3] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions On Evolutionary Computation*, 11(6), 2007.