Generational Neuro-Evolution: Restart and Retry for Improvement

David Shorten
University of Cape Town
Private Bag X3
Rondebosch 7701
dshorten@cs.uct.ac.za

Geoff Nitschke
University of Cape Town
Private Bag X3
Rondebosch 7701
gnitschke@cs.uct.ac.za

ABSTRACT

This paper proposes a new *Neuro-Evolution* (NE) method for automated controller design in agent-based systems. The method is *Generational Neuro-Evolution* (GeNE), and is comparatively evaluated with established NE methods in a multi-agent predator-prey task. This study is part of an ongoing research goal to derive *efficient* (minimising convergence time to optimal solutions) and *scalable* (effective for increasing numbers of agents) controller design methods for adapting agents in neuro-evolutionary multi-agent systems. Dissimilar to comparative NE methods, GeNE employs tiered selection and evaluation as its generational fitness evaluation mechanism and, furthermore, re-initializes the population each generation. Results indicate that GeNE is an appropriate controller design method for achieving efficient and scalable behavior in a multi-agent predator-prey task, where the goal was for multiple predator agents to collectively capture a prey agent. GeNE outperforms comparative NE methods in terms of efficiency (minimising the number of genotype evaluations to attain optimal task performance).

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents

General Terms

Algorithms

Keywords

Neuro-Evolution, Conventional Neuro-Evolution, Multi-Agent Enforced Sub-Populations

1. INTRODUCTION

Neuro-Evolution [29], [9] has been demonstrated as effective for evolving various problem-solving collective behaviors in agent based systems [11], [24]. Collective behaviors (those requiring agent cooperation) that emerge as the result of interactions of cooperative or specialised agent behaviors, have been demonstrated as adaptive, robust and efficient problem solvers in a broad range of tasks [18], [20], [21], [5]. For example, collective behaviors have been evolved to solve tasks such as multi-robot coordination and cooperation [18], [2], multi-agent game playing [27], and multi-agent pursuit-evasion [19], [5], [32].

However, an open research objective is to derive controller design methods that evolve *efficient* and *scalable* collective behaviors that are consistently effective across a broad range of tasks. In this study, *efficient* refers to the number of fitness (genotype) evaluations a method requires to reach an optimal solution, and *scalable* refers to the capability to retain efficient behavior as task complexity increases.

This paper introduces the *Generational Neuro-Evolution* (GeNE) method for automated controller design. GeNE contributes a tiered approach for selecting and evaluating progressively fitter genotypes during each generation. This allows low performance genotypes to quickly be removed and the population to be periodically re-initialized in the region of the fittest genotypes. This allows GeNE to rapidly focus its search in optimal parts of the fitness landscape.

For complex tasks, NE usually requires many fitness (genotype) evaluations before a near optimal or optimal solution is attained, and genotype evaluations are computationally expensive and time consuming [15]. This problem is exacerbated by multi-agent (collective behavior) tasks, where the interactions of multiple agents required to solve such tasks increases the dimensionality of the search space. Also, many collective behavior tasks, such as multi-robot cooperative transport and collective construction, are defined by noisy, discontinuous, multi-modal search spaces [26], [1].

Several methods have been proposed as a means of increasing *Evolutionary Algorithms* (EAs) efficiency. For example, the interpolation between genotypes to build a meta-model of the fitness function [16], the inheritance of fitness from ancestral genotypes [25], and clustering genotypes and assigning fitness to population sub-sets [17]. All of these approaches reduce the number of fitness evaluations required, increasing EA efficiency in their respective tasks.

However, there has been relatively little successful research into NE methods yielding increased efficiency and scalability in collective behavior tasks. To increase efficiency and scalability, GeNE combines techniques used in *Evolutionary Strategies* (ES) [4] and *Hill Climbers* [23]. That is, similar to ES, GeNE employs only *mutation* operators [7], and similar to *Hill Climbers* [23], the genotype population is periodically re-initialized in the neighbourhood of the current fittest solutions in the fitness landscape [3]. An initial experimental validation of GeNE's efficiency and scalability is conducted in controller adaptation in a multi-agent predator-prey (*pursuit-evasion*) task for a range of predator team sizes. *Conventional Neuro-Evolution* (CNE) [12], and *Multi-Agent Enforced Sub-Populations* (MESP) [32] are also tested in the pursuit-evasion task. CNE and MESP were selected as comparative methods since both have been successfully applied to the pursuit-evasion task [19], [32].

Pursuit-evasion is indicative of a broader range of multi-agent tasks that use emergent collective behavior such as cooperation to enable problem solving [6]. Such tasks include multi-robot surveillance [10], [20], collective construction [21], and coordinated movement and search [2]. The pursuit-evasion task in this paper used predator and prey agents moving at equal speeds in a wrap-around world. A consequence of this was that cooperation between predators was mandated for prey capture.

Results support the efficacy of GeNE using efficiency and scalability metrics in the pursuit-evasion task, and yield an initial insight into the type of collective behavior tasks to which GeNE is most suited as a controller design method.

1.1 Research Goal, Hypothesis, and Task

Research goal: Demonstrate that GeNE out-performs (with statistical significance) the CNE and MESP controller design methods in a multi-agent pursuit-evasion task. Efficiency and scalability metrics are used as the performance metrics.

Research hypothesis: GeNE's tiered genotype evaluation mechanism allows GeNE evolved predator teams to achieve a higher task performance (with statistical significance) in the pursuit-evasion task, compared to CNE and MESP evolved predator teams.

Pursuit-evasion task: Requires multiple predators to adapt their collective behavior such that they capture a prey agent [19]. The behavior of the prey is not adapted during simulation, but rather uses a static heuristic predator-evasion behavior from previous experiments [32].

2. METHODS

2.1 CNE: Conventional Neuro-Evolution

CNE [12] uses the NE process illustrated in *Figure 1(a)*. A *Genetic Algorithm* (GA) [7] is applied to evolve a population of *Artificial Neural Network* (ANN) controllers (section 3.1). Each team of predator controllers is represented in the genotype population as a vector of all ANN connection weights. CNE predator teams are heterogeneous in that the genotype encodes N distinct predator ANN controllers. Each generation of CNE, the team is evaluated in multiple task trials (predator team lifetimes), and an average fitness assigned to the genotype (predator team) being evaluated.

2.2 MESP: Multi-Agent Enforced Sub-Populations

MESP [32] uses cooperative co-evolution to adapt behaviors in a team of agent controllers (ANNs). Given N genotype populations, N controllers are evolved, and evaluated according to how well they solve a collective behavior task. MESP teams are heterogenous since each controller is evolved from a separate population. *Figure 1* illustrates an example of MESP applied to evolve three ANN predator controllers.

MESP evolves hidden layer neuron weights and within each population combines the fittest u neurons, selected from u sub-populations (in each population) into complete ANN controllers. Each sub-population evolves a neuron for a specific hidden layer position, where neurons are evaluated based on how well they function in a complete controller (determined by fitness evaluations). Yong and Miikkulainen [32] present a comprehensive description of MESP.

2.3 GeNE: Generational Neuro-Evolution

GeNE's genotype is the same as used by CNE. That is, a vector of all ANN connection weights for a team of N controllers. However, GeNE differs from CNE in the following respects:

1. GeNE uses mutation only. The motivation for this design choice was to investigate (within the context of a collective behavior task) the benefits of mutated solutions being applied, per generation, to progressively fitter regions of the genotype space.
2. GeNE is generational. At the end of each generation the population is re-initialized in a *Cauchy* distribution around the generation's fittest genotype.
3. GeNE's genotype evaluation uses a tiered structure. Initially, all genotypes are evaluated comparatively few times. Only genotypes which perform well on these evaluations are passed on to a subsequent tier of evaluations in which they are evaluated further. The process of passing fewer genotypes on to higher tiers continues until only one genotype remains. The population is re-initialized about this fittest genotype.

The GeNE process is enumerated in the following.

1. *Tiered Evaluation:* The genotypes are passed through T evaluation tiers. Tier t accepts n_t input genotypes, evaluates them in ρ_t task trials, and outputs the n_{t+1} fittest genotypes ($n_{t+1} < n_t$). These n_{t+1} fittest genotypes are then input to tier $t+1$. This process continues until one genotype is output by tier T. A task trial was z iterations of a simulation run, during which a genotype was evaluated.
2. *Re-initialisation:* At the end of each generation, the population is re-initialized around the generation's fittest genotype. Every weight of the fittest genotype has a random value drawn from a *Cauchy* distribution added to it. This process is similar to burst mutation [32].
3. *Stagnation Control:* The fittest genotype is recorded at each generation. Given that fitness values are normalized between 0 and 100, then if the fittest genotype of the current generation is not 20 units or fitter than the fittest genotype 3 generations ago, then stagnation has occurred. If so, GeNE is restarted.
4. *Stop condition:* Re-iterate steps [1, 3] until convergence is achieved (prey-capture in this study), or GeNE has run for Y generations. A generation is the pro-

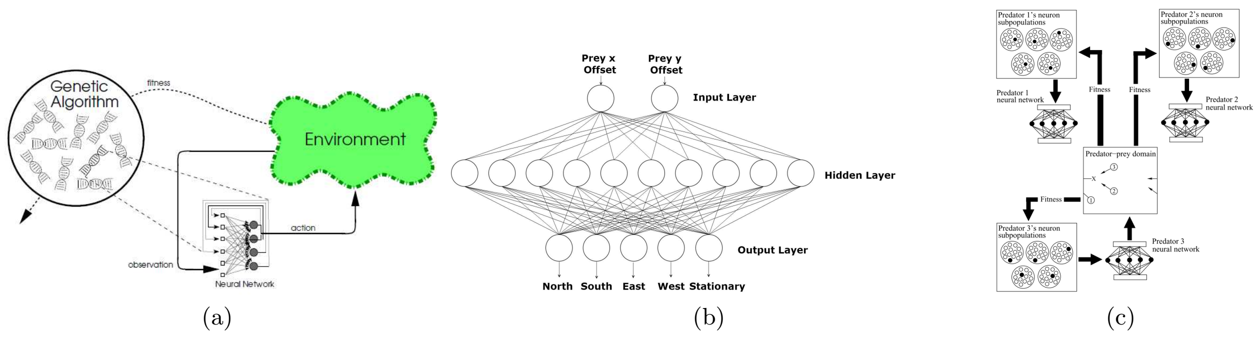

Genetic Algorithm
fitness
Environment
action
observation
Neural Network
(a)
Prey x Offset
Prey y Offset
Input Layer
Hidden Layer
Output Layer
North South East West Stationary
(b)
Predator 1's neuron subpopulations
Predator 2's neuron subpopulations
Predator 1 neural network
Predator 2 neural network
Fitness
Fitness
Predator–prey domain
Fitness
Predator 3's neuron subpopulations
Predator 3 neural network
(c)

Figure 1: ***(a) Conventional Neuro-Evolution (CNE).*** **Complete ANN controllers are evolved from one population (Figure from Gomez et al. [12], used with the permission of the authors).** ***(b) Predator ANN Controller.*** **Connection weights were evolved by CNE, MESP or GeNE in pursuit-evasion experiments.** ***(c) Multi-Agent Enforced Sub-Populations (MESP).*** **Example of MESP applied to evolve three predator controllers in a pursuit-evasion task (Figure from Yong and Miikkulainen [32], used with the permission of the authors and the *IEEE*).**

cess whereby genotypes move up through the evaluation tiers followed by re-initialization around the fittest genotype.

Figure 2 illustrates the GeNE process.

3. PURSUIT-EVASION TASK

Pursuit-evasion requires a *predator* team to collectively capture a *prey* in a multi-agent simulation [32]. The environment was a 100 x 100 wrap-around grid. Predators always started in the top-left corner grid square, and the prey at a random position. Both predators and prey moved at equal speeds and had a global view of the environment (always sensing opponent positions). The task was for the predators to minimise the number of simulation (task trial) iterations to prey-capture. Prey-capture occurs when at least one predator moves onto the same grid-square as the prey. Given equal movement speeds, and a wrap-around environment, at least two predators are required to capture the prey [32]. Predator and prey movement was limited to *north*, *south*, *east* and *west*, and one grid square per iteration. Predators and prey *lived* for 150 iterations (one *task trial* in table 1).

3.1 Predator and Prey Controllers

Predator sensory inputs were mapped to motor outputs using a feed-forward *Artificial Neural Network* (ANN) controller (Figure 1, centre). Two input and five output neurons were fully connected to 10 *Hidden Layer* (HL) neurons. HL and output neuron activation values were computed by a *Sigmoid* function [14]. The output node with the highest activation value was selected as the output (action) of the given simulation iteration. This approach was selected since it was used in the NE controller evolution experiments of Yong and Miikkulainen [32]. Furthermore, the MESP method is re-implemented in this study with the goal of reproducing previous results [32], as well as comparing them with the CNE and GeNE methods.

The prey controller was a heuristic that directed the prey to move in the opposite direction of the closest predator.

3.2 Fitness Function: Predator Evaluation

Predator team prey capture behavior (evolved by CNE, MESP, and GeNE) was evaluated in terms of *efficiency* and *scalability*. That is, the number of fitness evaluations that elapsed before task completion, and the efficiency of evolved prey-capture behaviors as predator team size is increased.

The fitness function shown in equation 1 was used by CNE, MESP and GeNE, where d_e was the average distance between the predators and the prey at the end of a task trial, 100 was the environment length and width, and 200 was used to increase fitness awarded for prey capture.

$$f(d_e) = \begin{cases} 100 - d_e & \text{if the prey was not caught} \\ 200 - d_e & \text{if the prey was caught} \end{cases} \qquad (1)$$

4. EXPERIMENTS

Experiments applied either CNE (section 2.1), MESP (section 2.2), or GeNE (section 2.3) to evolve the connection weights of ANN controllers (figure 1, centre), and thus adapt predator team behavior. One experiment was CNE, MESP or GeNE executed for Y generations, or until a threshold task performance was attained. Each experiment was executed for 200 runs. Table 1 presents the parameters used in these experiments.

The task performance threshold was defined as prey capture for at least 80% of agent starting positions. The prey had 100 possible starting positions in a 10x10 grid in the bottom left corner of the environment. An average fitness of CNE, MESP and GeNE evolved teams was calculated over all runs. CNE, MESP, and GeNE each tested predator team sizes: N = [3, 4, 5, 6]. Each generation of the GeNE, MESP, and CNE methods consists of multiple task trials. A task trial was one predator team executed for 150 simulation iterations (table 1). Each task trial tested different prey starting positions as a means of sampling the many possible starting positions.

To ensure a fair comparison, all methods used the same population size of 500 genotypes per predator, where team sizes of three to six predators were tested. As per previous work [12], the number of genotype evaluations varied, for different methods, as the predator team size increased.

Table 1 presents the simulation and NE method parameter settings. These parameter values were determined experimentally so as to maximize the performance of all the al-

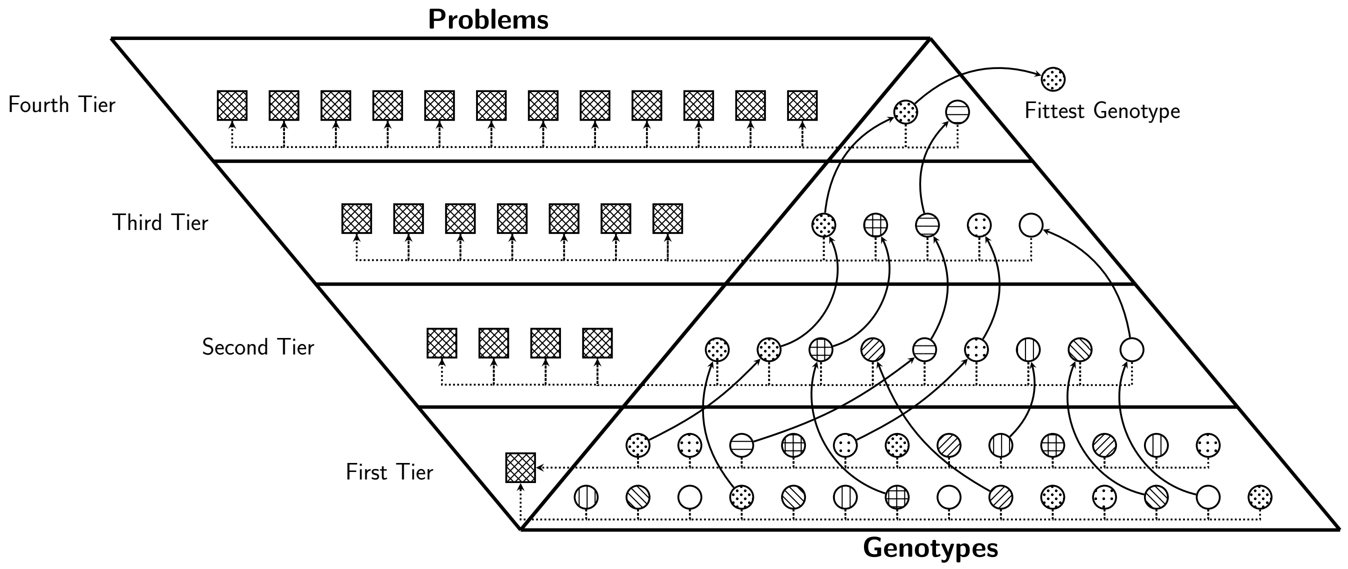


Figure 2: The tiered evaluation process of a generation of a hypothetical run of GeNE. The round nodes represent genotypes and the square nodes represent problems (or problem instances) on which they can be evaluated. A solid edge represents a genotype being promoted from a lower tier to a higher one. A dashed edge represents a genotype being evaluated on a problem. In the pursuit-evasion task multiple problem instances (agent starting positions) are evaluated.

gorithms involved. Minor changes to these values produced similar results for CNE, MESP and GeNE evolved teams. Unless otherwise specified, the parameters listed in table 1 apply to all NE methods.

Yong and Miikkulainen executed similar pursuit-evasion experiments on MESP [31]. They observed that the evolution of three predator teams capable of capturing the prey seven times, for nine different starting positions, required on average 108 000 evaluations (standard deviation of 42000). The experiments presented in this paper found that the evolution of three predator teams capable of capturing the prey eighty times, given one hundred different starting positions required, on average 94 000 evaluations (standard deviation of 112 000, table 3). This demonstrates that the higher task performance of GeNE evolved teams is not a result of parameter tuning issues in either method.

4.1 Cauchy Distribution

Cauchy distribution was used as a mutation operator in our experiments. The reasons for this are:

1. Testing on numerous instances with the Cauchy distribution indicated that it produced better task performance than the normal distribution.
2. In the related field of evolutionary programming, the Cauchy distribution has been found to yield higher performance than the normal [30].
3. There is precedent for using the Cauchy distribution in NE [32].

4.2 CNE: Setup

CNE (section 2.1) predator controllers were evolved from a population of N (predator team size) x 500 genotypes. Each genotype encoded N controllers as a vector of N x 70 real values. That is, two input neurons plus five output neurons fully connected to 10 hidden layer neurons. The CNE process was as follows.

Each gene (weight) in each genotype of the population was *initialized* to a random value from a *Cauchy* distribution ($x_0 = 0$ and $\gamma = 1$). Each genotype was decoded into N ANNs to derive a heterogeneous team of predator controllers. Each team (genotype) was then systematically selected and evaluated in the pursuit-evasion task for 20 task trials. Each team's fitness was calculated as an average over all task trials. At the end of each generation, genotypes in the fittest 25% of the population were randomly paired and *uniform crossover* [7] applied. Enough child genotypes were created to replace the least fit 50%. A child was mutated with a 0.4 degree of probability. If a child was selected for mutation, then N x 10 randomly chosen weights, were modified via adding a randomly drawn value from a *Cauchy* distribution ($x_0 = 0$ and $\gamma = 0.5$).

4.3 MESP: Setup

MESP (section 2.2) evolved N predator ANNs from N populations. Each population contained 10 sub-populations, where each sub-population contained 50 genotypes (hidden layer neurons). That is, the weights of 10 hidden layer neurons were evolved from 10 sub-populations to form complete ANN controllers. A MESP genotype was a vector of seven real values encoding the connection weights of two sensory inputs plus five outputs of a hidden layer neuron. The MESP process was as follows.

Each gene (weight) in each genotype of the population was *initialized* to a random value from a *Cauchy* distribution ($x_0 = 0$ and $\gamma = 1$). In line with the evaluation approach used in previous research [32], evaluation in MESP occurred over *10 task trial sets*. In each task trial set, 50 *groups* (where 50 was the sub-population size) of task trials were performed. A group to be evaluated was composed via

Table 1: Pursuit-Evasion Parameters. Simulation and Neuro-Evolution method settings.

Pursuit-Evasion Parameter Settings	
Simulation runs / Environment size / Predator team size (N)	200 / 100 x 100 / [3, 6]
Predator / Prey controller	Feed-Forward ANN / Heuristic
Predator ANN Input/ Output / Hidden layer neurons	2 / 5 / 10
Predator / Prey sensor range	Environment length
Genotype structure (CNE / MESP / GeNE)	N x ANN weights / Hidden neurons / N x ANN weights
Genotype length (CNE / MESP / GeNE)	N x 70 / 7 / N x 70
Genotype representation	Real valued vector
Generations / Trial iterations (team lifetime)	500 / 150
Mutation probability	0.4
Mutation *Cauchy* distribution γ (CNE / MESP / GeNE)	0.1 / 0.5 / see table 2
Crossover operator (CNE, MESP)	uniform
Population size (CNE, MESP, GeNE)	N x 500
Genotype populations (CNE/MESP/GeNE)	1 / [3, 6] / 1
Sub-populations / Sub-population size (MESP)	10 / 50
Parent selection (MESP, CNE) / (GeNE)	Elitist: top 25% / Tiered selection (section 2.3)
Population replacement	Lower 50% of population

randomly selecting a genotype g_{ij} from each sub-population p_{ij} ($1 \leq j \leq 10$). This was done for each population P_i ($1 \leq i \leq N$, $3 \leq N \leq 6$), where N is the number of predators in a team. Selected genotypes were marked so as they were not evaluated again in a given task trial set. This ensured that each genotype was evaluated the same number of times.

From these selected genotypes, N predator controllers were constructed, and this team was evaluated in six task trials. An average fitness (calculated over the task trials) was assigned to each genotype that participated in the task trial set. A genotype's fitness was calculated as its average fitness across 10 task trial sets.

At the end of each generation, genotypes in the fittest 25% of p_{ij} were randomly paired and *uniform crossover* applied. Enough child genotypes were created to replace the least fit 50%. A child was mutated with a 0.4 degree of probability. If a child was selected for mutation, then N x 10 randomly chosen weights, were modified via adding a randomly drawn value from a *Cauchy* distribution ($x_0 = 0$ and $\gamma = 0.5$).

4.4 GeNE: Setup

GeNE (section 2.3) evolves predator teams from a population of N x 500 genotypes, where each genotype encodes the connection weights of N predator controllers. So as to allow reuse of the MESP architecture, a variation on GeNE's initialisation and re-initialisation procedures (section 2.3) was used for these experiments.

For initialization, N MESP populations were initialized, where each population contained 10 sub-populations and each sub-population consisted of 100 genotypes (section 4.3). For the construction of each GeNE genotype, one MESP genotype was randomly chosen from each of the 10 sub-populations in each of the N populations. These N x 10 genotypes then represented one GeNE genotype.

Re-initialization occurred via mutating the weights of all MESP genotypes that composed the fittest GeNE genotype (for a given generation). This had the effect of re-initializing the corresponding sub-populations (from which the GeNE genotype was composed). Table 2 specifies the γ used for re-initialization and the parameter values used in GeNE's tiered evaluation. Table 2 also presents the number of genotypes (teams) selected for evaluation in each tier, and the number of task trials (ρ) per team evaluation per tier. These values were determined experimentally and found to work well for the pursuit-evasion task.

4.5 Lesion Studies

Lesion studies were done to ascertain the contribution of GeNE's tiered evaluation and re-initialization (section 2.3). For each lesion study, resultant data sets of GeNE evolved teams in non-lesioned studies (section 4.4) were compared with lesioned study results for statistically significant differences ($p < 0.01$) using independent two-sample t-tests [8].

Lesion study 1: This study tested only one tier of evaluation (four were used in the pursuit-evasion task) per GeNE generation. This tier had a size of 123 and ρ was set to 49. The goal was to elucidate the benefit of multiple tiers of evaluation. Supporting this, results indicated a higher (with statistical significance, $p < 0.01$) average task performance (for team sizes N = [3, 4, 5, 6]) of GeNE evolved teams in non-lesioned versus lesioned experiments (figure 3(b)).

Lesion study 2: This study tested four tiers of evaluation to elucidate the impact of GeNE's re-initialization procedure. To test this, this lesion study only mutated each

Table 2: Left: GeNE Tiered Evaluation. N genotypes (teams) are input to each tier. These teams are tested in ρ task trials. A fittest subset of teams is output (input for the next tier). ρ: Number of task trials a team is tested in. Right: γ values used for GeNE population re-initialization given the previous generation's fittest team's capture ratio.

Tier	**Input Teams**	**Output Teams**	ρ	**Capture Ratio (r)**	γ
1	2000	500	1	$0 \leq r < 0.1$	0.9
2	500	50	4	$0.1 \leq r < 0.2$	0.7
3	50	12	16	$0.2 \leq r < 0.3$	0.5
4	12	1	100	$0.3 \leq r < 0.5$	0.4
-	-	-	-	$0.5 \leq r < 0.6$	0.3
-	-	-	-	$0.6 \leq r < 0.8$	0.15

Table 3: Task performance of CNE, MESP, and GeNE evolved predator teams: Average number (in thousands) of fitness (genotype) evaluations until prey capture. Values in parentheses are standard deviations.

Method	**3 Predators**	**4 Predators**	**5 Predators**	**6 Predators**
CNE	3172.35 (863.60)	3981.20 (914.53)	4296.25 (741.06)	4633.80 (365.87)
MESP	93.51 (112.70)	48.70 (10.94)	44.84 (8.62)	44.04 (8.10)
GeNE	37.15 (22.64)	27.99 (16.87)	24.02 (10.57)	24.33 (9.15)
GeNE (Lesion study 1)	109.06 (84.11)	60.42 (43.35)	52.23 (32.19)	39.72 (22.31)
GeNE (Lesion study 2)	69.44 (59.50)	46.17 (35.34)	30.68 (17.46)	32.06 (17.35)

weight in a GeNE genotype with a 0.2 degree of probability. Lesion study 2 results indicated a higher (with statistical significance, $p < 0.01$) average task performance (for team sizes $N = [3, 4, 5, 6]$) of GeNE evolved teams in non-lesioned versus lesioned experiments (figure 3(b)).

5. RESULTS AND DISCUSSION

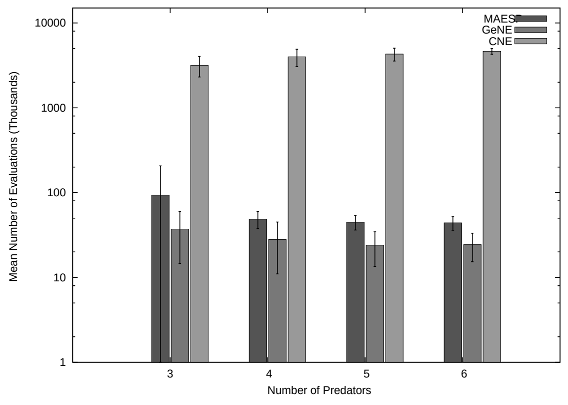
Figure 3(a) presents the average number of *fitness* evaluations (calculated over 200 runs for each method) required by CNE, MESP and GeNE evolved teams to achieve the task performance threshold for method *efficiency* (section 4). These results are also shown in table 3. The *scalability* (maintaining efficiency as team size increased) of CNE, MESP and GeNE evolved prey-capture behaviors was also tested for team sizes of three to six predators.

Statistical comparisons (two-sample independent t-tests, $p < 0.01$) indicated that GeNE evolved teams yielded a significantly higher average efficiency, where this higher efficiency was maintained for all team sizes tested. This partially addresses the research objective (section 1.1) in that results indicate that GeNE out-performs CNE and MESP according to the efficiency metric.

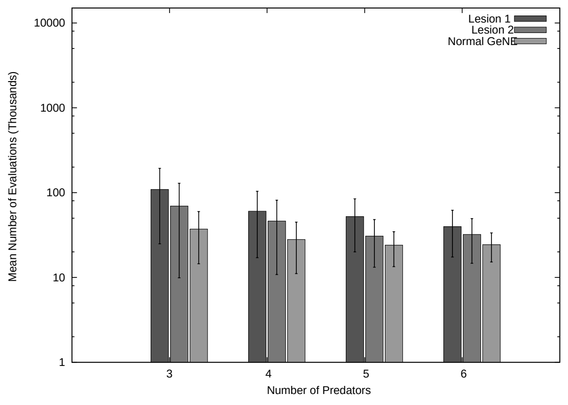
Figure 3(a) indicates that all methods were scalable across the different predator team sizes which were tested. There were no major differences in the number of evaluations required as the predator team size increased. This was theorized to be the result of the interactions of solution space dimensionality and the number predators in the simulation. That is, increasing the number of predators increased the dimensionality of the solution space, however, a larger team size also increased the likelihood that the prey would be captured during the team's lifetime. However, only predator team sizes in the range [3, 6] were tested. In order for scalability to be properly demonstrated, a wider range of predator team sizes would need to be tested. It is thus concluded that the second part of the research goal (demonstrating scalability) was not fully addressed.

Figure 3(b) presents the average number of genotype evaluations taken by GeNE evolved teams versus lesioned GeNE evolved teams (section 4.5), for team sizes of up to six predators. Statistical comparisons ($p < 0.01$) indicated that lesioned GeNE evolved teams yielded a significantly lower average efficiency, that was consistent across all team sizes tested. These lesion study comparisons support the hypothesis (section 1.1) that GeNE's tiered genotype evaluation mechanism allows it to achieve a higher task performance, compared to CNE and MESP evolved predator teams. That is, lesion study 1 results supported the benefits of GeNE's tiered evaluation process, and lesion study 2 results supported the benefits of GeNE's re-initialization process.

The comparatively high efficiency of GeNE is theorized to be a result of its tiered evaluation (section 2.3). Consider that this pursuit-evasion study required the evolution of a prey-capture behavior where prey-capture is exhibited above a given task performance threshold. The evaluation of prey-capture behavior required many task trials that sampled the space of initial states (prey start positions versus predator start positions). The more task trials used to evaluated a



(a)



(b)

Figure 3: *(a)* CNE, MESP and GeNE evolved teams. Plots of average number of fitness (genotype) evaluations for predator team sizes: [3, 4, 5, 6]. Bars show the standard deviation. *(a)* Comparative plots for GeNE versus GeNE (lesioned) for predator team sizes: [3, 4, 5, 6]. Bars show the standard deviation. Note the logarithmic scale.

given genotype (prey-capture behavior), the more accurate that genotype's evaluation will be. Thus, via its tiered evaluation, GeNE was able to focus more evaluations on genotypes that were more likely to yield effective prey-capture behaviors, and thus attain a more accurate evaluation of these genotypes, and propagate these genotypes throughout the evolutionary process.

Furthermore, it is theorized that the re-initialization process enabled GeNE to achieve and maintain a high efficiency for all team sizes tested. That is, per generation of GeNE, the re-initialization process mandated a shift in search focus from exploration to exploitation in a given search space area. That is, GeNE's search was initially in a wide area of the fitness landscape (defined by the γ parameter in the burst mutation operator), around the current fittest genotype. As the fitness of the fittest genotype at each generation improved, γ decreased together with the size of the search space area. This tiered evaluation process is similar to the function of the self-adaptive σ parameter in *Evolutionary Strategies* (ES) [4]. Hence, one supposition is that GeNE will yield a comparative task performance on the same tasks in which ES yield optimal or near optimal performance. As such, future work will comparatively test GeNE and various ES on canonical optimization benchmark tasks [4].

Furthermore, it is theorized that particular features of the multi-agent pursuit-evasion search space, make it amendable to an efficient search by GeNE, where such efficiency is *scalable* (maintained for increasing team sizes). That is, compared to CNE and MESP, GeNE yielded faster convergence to search space optima (prey-capture above a task performance threshold) in ANN connection weight spaces of increasing dimensionality. However, the exact algorithmic and task environment mechanisms responsible for this result is the subject of ongoing research.

Also, most problems upon which NE methods yield a high task performance are presented in either a *general* or a *specific* form. In the pursuit-evasion task, predator agents can be adapted to catch prey starting from a given location (specific) or from a set of locations (general). Similarly, in the pole-balancing task [12], agents can be adapted to balance poles for a specific initial condition or for a range of initial conditions. Future work will also test if GeNE has an advantage where a range of starting conditions must be considered in order to evolve an optimal solution, since GeNE spend more of its evaluations sampling the task performance of genotypes with the potential to be highly fit.

6. CONCLUSIONS AND FUTURE WORK

This paper presented a new NE controller design method: *Generational Neuro-Evolution* (GeNE). In this study GeNE was applied to evolve predator agent controllers in a multi-agent pursuit-evasion task and was tested comparatively with *Conventional Neuro-Evolution* (CNE) and *Multi-Agent Enforced Sub-Populations* (MESP). Behaviors evolved by CNE, MESP and GeNE were measured in terms of *efficiency* and *scalability*. Efficiency was measured as the number of fitness (genotype) evaluations until a pursuit-evasion task performance threshold was attained. Scalability was measured as a method's capability to maintain efficiency as predator team size increased. Results supported part of this research objective, demonstrating that GeNE evolved collective prey-capture behaviors that were consistently more *efficient* (with statistical significance, $p < 0.01$) compared to those evolved by CNE and MESP. However, scalability was not fully demonstrated due to a lack of testing over a large range of team sizes. Results also supported the hypothesis that GeNE's tiered evaluation contributed to the higher efficiency and scalability of GeNE evolved prey-capture behaviors.

Several approaches to future work are currently underway. Future work will attempt to validate the efficacy of GeNE, using metrics of efficiency and scalability, but tested in single and multi-agent tasks. For example, controller adaptation in double-pole balancing [12], and multi-robot controller adaptation for collective behavior tasks such as cooperative transport [13], collective gathering [22], coordinated movement [2] or collective construction [28]. The efficacy of GeNE will also be tested for other controller types and genotype representations. ANNs and NE were selected since both are well established and as such considered appropriate for a preliminary study. Successful application of GeNE to canonical ES and NE benchmark tasks and more complex single and multi-agent tasks would contribute to establishing the efficacy of GeNE as an NE controller design method, and more generally as a broadly applicable evolutionary algorithm.

7. REFERENCES

[1] T. Arai, E. Pagello, and L. Parker. Advances in multi-robot systems. *IEEE Transactions on Robotics and Automation*, 18(5):655–661, 2002.

[2] G. Baldassarre, D. Parisi, and S. Nolfi. Distributed coordination of simulated robots based on self-organisation. *Artificial Life*, 12(3):289–311, 2006.

[3] D. Beasley, D. Bull, and R. Martin. A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1(2):101–125, 1993.

[4] H. Beyer and H. Schwefel. Evolution strategies: A comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.

[5] D. D'Ambrosio and K. Stanley. Scalable multiagent learning through indirect encoding of policy geometry. *Evolutionary Intelligence*, 6(1):1–26, 2013.

[6] A. Eiben, G. Nitschke, and M. Schut. Evolving team behaviors with specialization. *Genetic Programming and Evolvable Machines*, 13(4):493–536, 2012.

[7] A. Eiben and J. Smith. *Introduction to Evolutionary Computing*. Springer-Verlag, Berlin, Germany, 2003.

[8] B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes*. Cambridge University Press, Cambridge, 1986.

[9] D. Floreano, P. Dürr, and C. Mattiussi. Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1(1):47–62, 2008.

[10] E. Folgado, M. Rincon, J. Alvarez, and J. Mira. A multi-robot surveillance system simulated in gazebo. In *Nature Inspired Problem-Solving Methods in Knowledge Engineering*, pages 202–211. Springer Berlin, Heidelberg, Germany, 2007.

[11] R. Goldstone and M. Janssen. Computational models of collective behavior. *TRENDS in Cognitive Sciences*, 9(9):424–429, 2005.

[12] F. Gomez, J. Schmidhuber, and R. Miikkulainen. Accelerated neural evolution through cooperatively coevolved synapses. *Journal of Machine Learning Research*, 9(1):937–965, 2008.

[13] R. Gross and M. Dorigo. Evolution of solitary and group transport behaviors for autonomous robots capable of self-assembling. *Adaptive Behavior*, 16(5):285–305, 2008.

[14] J. Hertz, A. Krogh, and R. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, 1991.

[15] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1):3–12, 2005.

[16] Y. Jin, M. Olhofer, and B. Sendhoff. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation*, 6(5):481–494, 2005.

[17] H. Kim and S. Cho. An efficient genetic algorithms with less fitness evaluation by clustering. In *Proceedings of the Congress on Evolutionary Computation*, pages 887–894, Seoul, Korea, 2001. IEEE.

[18] L. Li, A. Martinoli, and A. Yaser. Learning and measuring specialization in collaborative swarm systems. *Adaptive Behavior.*, 12(3):199–212, 2004.

[19] G. Nitschke. Designing emergent cooperation: a pursuit-evasion game case study. *Artificial Life and Robotics*, 9(4):222–233, 2005.

[20] G. Nitschke, M. Schut, and A. Eiben. Collective neuro-evolution for evolving specialized sensor resolutions in a multi-rover task. *Evolutionary Intelligence*, 3(1):13 29, 2010.

[21] G. Nitschke, M. Schut, and A. Eiben. Evolving behavioral specialization in robot teams to solve a collective construction task. *Swarm and Evolutionary Computation*, 2(1):25–38, 2012.

[22] S. Nouyan, R. Gross, M. Bonani, F. Mondada, and M. Dorigo. Teamwork in self-organized robot colonies. *IEEE Transactions on Evolutionary Computation*, 13(4):695–711, 2009.

[23] S. Russell and P. Norvig. *Artificial Intelligence, A Modern Approach. Second Edition*. University of Michigan Press, Ann Arbor, USA, 2003.

[24] R. Sarker and T. Ray. Agent based evolutionary approach: An introduction. In *Agent-Based Evolutionary Search, Adaptation, Learning, and Optimization*, pages 1–11. Springer, Berlin, Germany, 2010.

[25] K. Sastry, D. Goldberg, and M. Pelikan. Don't evaluate, inherit. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 551–558, San Francisco, USA, 2001. Morgan Kaufmann.

[26] C. Schultz and L. Parker. In *Multi-robot Systems: From Swarms to Intelligent Automata*. Kluwer, Washington DC, USA, 2002.

[27] K. Stanley, B. Bryant, and R. Miikkulainen. Real-time neuro-evolution in the nero video game. *IEEE Transactions Evolutionary Computation*, 9(6):653–668, 2005.

[28] J. Werfel and R. Nagpal. Extended stigmergy in collective construction. *IEEE Intelligent Systems*, 21(2):20–28, 2006.

[29] X. Yao. Evolutionary artificial neural networks. *International Journal of Neural Systems*, 4(3):203–222, 1993.

[30] X. Yao, Y. Liu, and G. Lin. Evolutionary programming made faster. *Evolutionary Computation, IEEE Transactions on*, 3(2):82–102, 1999.

[31] C. Yong and R. Miikkulainen. Cooperative coevolution of multi-agent systems. *University of Texas at Austin, Austin, TX*, 2001.

[32] C. Yong and R. Miikkulainen. Coevolution of role-based cooperation in multi-agent systems. *IEEE Transactions on Autonomous Mental Development*, 1:170–186, 2010.