Evolution of Complex Autonomous Robot Behaviors using Competitive Fitness

A. L. Nelson, E. Grant, G. Barlow, M. White

Center for Robotics and Intelligent Machines
Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695-7911

Abstract–*Evolutionary Robotics (ER) employs population-based artificial evolution to develop behavioral robotics controllers. In this paper we focus on the formulation and application of a fitness selection function for ER that makes use of intra-population competitive selection. In the case of behavioral tasks, such as game playing, intra-population competition can lead to the evolution of complex behaviors. In order for this competition to be realized, the fitness of competing controllers must be based mainly on the aggregate success or failure to complete an overall task. However, because initial controller populations are often sub-minimally competent, and individuals are unable to complete the overall competitive task at all, no selective pressure can be generated at the onset of evolution (the Bootstrap Problem). In order to accommodate these conflicting elements in selection, we formulate a bimodal fitness selection function. This function accommodates sub-minimally competent initial populations in early evolution, but allows for binary success/failure competitive selection of controllers that have evolved to perform at a basic level. Large arbitrarily connected neural network-based robot controllers were evolved to play the competitive team game* **Capture the Flag**. *Results show that neural controllers evolved under a variety of conditions were competitive with a hand-coded knowledge-based controller and could win a modest majority of games in a large tournament.*

1. INTRODUCTION

The goal of evolutionary robotics (ER) is to develop automatic methods of autonomous mobile robot controller synthesis that do not require hand coding or in depth human knowledge of the robot task for which the controller is intended. ER uses population-base artificial evolution to evolve autonomous robot controllers. In most current research, the evolved controller structures are artificial neural networks (ANN). The process of controller evolution consists of a repeating cycle of several steps that are roughly analogous to a generation in a natural evolutionary process. During the cycle, individual neural controllers in a larger population of neural controllers perform a task or engage in an evaluation period. Following this, each neural controller's performance is evaluated based on a fitness selection metric. In the final step of the cycle, a genetic algorithm (GA) is applied. The GA uses information generated by the fitness selection function to select and propagate the fittest individuals in the current population to the next generation population. During propagation, controller networks are altered slightly using stochastic genetic operators such as mutation and crossover to produce offspring that make up the next generation of controllers. This cycle is repeated for many generations to train populations of robot controllers to perform a given task.

Much of the ER research reported on to date has investigated the evolution of controllers for extremely simple behaviors such as phototaxis [1][2] or object avoidance [3][4]. In such cases, fitness selection metrics can be formulated by trial and error. The resulting formulations can include terms that describe sub-behaviors as well as simple sensor-actuator responses. With difficulty, and with sufficient knowledge of the dynamics of a behavior, this can be extended to evolve controllers for somewhat more complex robot tasks. For example, in [5] the authors describe the evolution of a coordinated movement task involving several robots. Other examples of relatively complex behaviors evolved using complex hand-formulated fitness functions include [6] and [7]. Respectively, these report the evolution of an object collection and deposition task (garbage collection) and a task in which a robot must collide with objects (collect) in one zone and avoid them in another.

The main value of the proof-of-concept ER work to date is that it has shown that neural controller structures can be evolved to produce functional behaviors in autonomous robots (self regulating sensory/motor close loop systems). What has not been shown is that ER methods can be extended to generate complex novel behaviors. In particular, no ER work to date has shown that it is possible to evolve complex controllers in the general case or for generalized tasks.

Although issues such as neural architecture and evolutionary dynamics are important, we make the case that the formulation of appropriate fitness selection functions is now the major hurdle confronting the further development of ER. In the early days of ER opinions about the necessity of fully embodied evolution were promulgate [8]. Many of the initial concerns and criticisms of the field of ER that were

related to embodiment and transference from simulation to reality [9] have been addressed. However, concerns related to fitness selection remain largely unresolved. The development of methods for general fitness selection during evolution of controllers is crucial to the future of ER. This view is reflected in some recent ER literature [10] and has been pointed out earlier in [4].

The research presented in this paper applies competitive selection to the evolution robot behavioral controllers. The robots use very large neural networks with large numbers of processed video sensor inputs. We formulate a bimodal fitness selection function. The first mode of this function evaluates the potential of evolving controllers to produce a detectable level of completion of an overall behavior, however poorly, in a finite amount of time. The second mode is used to evaluate competitive performance of controllers that complete the overall behavior to some detectable degree. The first mode requires some human knowledge and bias to formulate whereas the second mode bases fitness solely on the ability of controllers to compete against one another to complete the overall behavior (or task) as a whole. A key feature of this method is that information from the second mode, if present, completely supersedes information from the first mode. This allows robot controllers that have gained a minimum level of competence to evolve in an unrestricted competitive evolutionary search space even though human knowledge and bias was used to achieve that minimal level competence.

We present results generated with an advanced colony of small mobile robots with evolved neural controllers. Using the bimodal fitness selection function, populations of neural controllers were evolved to play a competitive team game, *Capture the Flag*. The evolutionary process was conducted in simulated environments and testing was performed both in simulation and using real robots in a physical environment. Populations of controllers were evolved under several different environmental conditions. The best evolved individuals from several different evolved populations were tested using extensive competitive tournaments against a knowledge-based controller of well-defined abilities.

2. THE ER ENVIRONMENT

2.1 The Robot Task

In this research, populations of robot controllers were evolved to play a robot version of the competitive team game *Capture the Flag*. In this game, there are two teams of mobile robots and two stationary goal objects. All robots on team one and one of the goals are of one color (red). The other team members and their goal are another color (green). In the game, robots of each team must try to approach the other team's goal object while protecting their own goal. The robot which first comes within a range of its

opponent's goal wins the game for its team. The game is played in maze worlds of varying configurations.

2.1 The Robots

The physical robots used in this work are the EvBots [11][12]. These robots use vision base range-finding sensors for detection of their environment [13]. The robots are fully autonomous and are capable of performing all vision processing and control computation on board. Figure 1 shows a photograph of two EvBots. Each robot is fitted with a colored shell. The shells are used in the *Capture the Flag* game behavior and serve to differentiate robots on different teams.
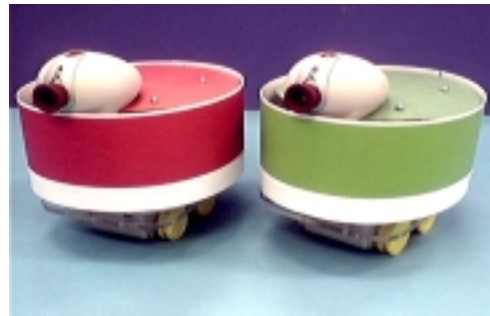
Figure 1. EvBot robots fitted with colored shells.

2.3 The Simulation Environment

Evolution of the neural controllers is performed in a simulated environment. The simulation environment is coupled to a real robot environment that is used for testing and verification.

In the simulation environment, robot agents, sensors, robot-environment and robot-robot interactions are modeled [13]. Simulated sensors extract range data from the environment and format them into the same format reported by real video range sensors used on the physical robots [14].

(a)   (b)
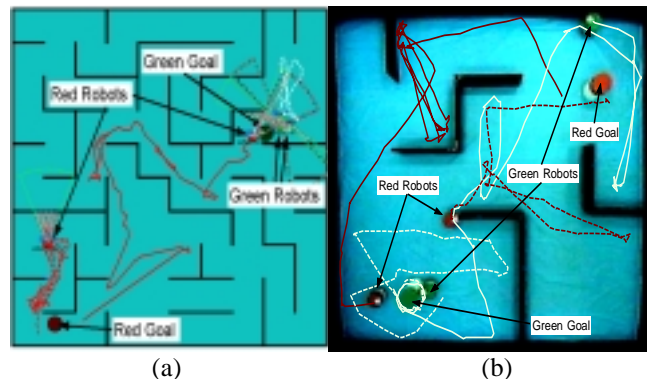
Figure 2. (a) Graphical representation of the simulation environment. The simulated range sensor data received by the robots are displayed as the pie-piece shaped graphics superimposed on the environment. (b) Real robots competing in a physical environment. The light and dark curves indicate the paths taken by the robots during the course of the game. The game was won be the red (dark) team.

Figure 3. An example neural network weight **(W)** and connectivity matrix (a) and associated graphical representation (b). Note that only non-zero weighted connections are shown in the graphical representation.

Figure 2 panel (a) shows a screen-capture form the simulation environment. The figure shows the final state of a game of *Capture the Flag* played between members of an evolving population. In this case, the red team has won the game. The path taken by the robot that won the game for its team is shown by the solid red (dark) line. Figure 2 (b) shows an overhead view of the physical maze environments. The figure shows the final positions of teams of real robots after a game. The light and dark curves superimposed on the image indicate the paths taken by the robots over the course of the game. In this case, the red (dark) team has won the game by locating their opponent's goal.

2.4 The Neural Network Architecture

A generalized class of network structures was used for the research reported on in this paper. The networks can be quite large and may contain 5000 or more weighted connections. These networks include: (1) arbitrary connectivity with feed forward and feedback connections, (2) mixed types of neurons, and (3) variable integer time delays on neuron inputs. Neuron activation function types include sigmoid, linear, step-threshold, and Gaussian radial basis functions. All network connectivity and weighting information is maintained in a Matrix **W**. An additional formatted vector structure **N** is required to store neuron type and time delay information. During evolution, **W** and **N** for each network in a population are operated on directly. No further genetic encoding or genome is used.

Figure 3 shows an example neural network's weight and connectivity matrix **W**, and its associated graphical representation. Note that only non-zero weighted connections are shown in the graphical representation. The network shown is much smaller than the networks actually evolved for control. It is included here as an example. This network formulation is amenable to evolutionary neural computing applications that use variable-size networks because neurons and connections can be added and removed easily without disrupting the connectivity relationships of the other neurons in a given network.

3. THE BIMODAL FITNESS FUNCTION

3.1 Fitness Function Formulation

Fitness for individual controllers was based on their performance in competition in tournaments of games. During each generation, a single tournament of games was played. A bimodal training fitness selection function was used. The selection function has an initial mode that accommodates sub-minimally competent seed populations and a second mode that selects for aggregate fitness based only on overall success or failure (winning or losing games). Additionally, this selection metric was applied in a relative competitive form in which controllers in an evolving population competed against one another to complete their task - to win the competitive game *Capture the Flag*.

Fitness $F(p)$ of an individual p in an evolving population **P** ($p \in \mathbf{P}$) takes the general form:

$$F(p) = F_{\text{mode\_1}}(p) \oplus F_{\text{mode\_2}}(p) \qquad (1)$$

where $F_{\text{mode\_1}}$ is the initial minimal-competence mode and $F_{\text{mode\_2}}$ is the purely success/failure based mode. Here $\oplus$ indicates dependant exclusive-or: if the success/failure based mode's value is non-zero, it is used and any value from $F_{\text{mode\_1}}$ is discarded. Otherwise fitness is based on the output of $F_{\text{mode\_1}}$. $F_{\text{mode\_1}}$ is formulated to return negative values and returns 0 when maximized or if $F_{\text{mode\_2}}$ is active. $F_{\text{mode\_2}}$ in contrast returns positive values based on number of game-wins, if any.

The first mode of the fitness function selects for minimal competence to successfully complete the task (however poorly) in a detectable fraction of the trials, and in a finite amount of time. In essence, the mode selects for the ability to travel a distance D through the competition environment. The general form of mode 1 is as follows:

$$F_{mode\_1} = F_{dist} - s - m \qquad (2)$$

where F_{dist} calculates a penalty proportional to the difference between distance d travel by the best robot on a team, and the minimal competence distance D:

$$F_{dist} = \begin{cases} -\alpha*(D-d) & \text{if } d < D \\ 0 & \text{otherwize} \end{cases} \qquad (3)$$

D is defined as half the length of the training environment's greatest dimension and α is a constant of proportionality. In Equation (2), s and m are penalty constants applied in the case that robots on a team becoming immobilized or stuck (by any means), and, in the case of controllers, producing actuator output commands that exceed the range of the actuators (the wheel motors) respectively.

The second mode of the fitness function F_{mode_2} is classified as aggregate because it produces fitness based only on success or failure of the controllers to complete the task at hand (competitive team game playing). The formulation of the success/failure mode (F_{mode_2}) of the fitness function is determined by the competitive nature of the training algorithm and the behavioral task. Here, competitive games were played, so success or failure was determined by winning or losing games. In each generation, a tournament of games involving all the individuals in the population was conducted. Each individual played two games against one other member of the population (the opponent). Note that the opponent was selected at random from the population at the beginning of each tournament. The possible outcomes of these games incurred different levels of fitness and are summarized in Table. 1 below.

Table 1. Fitness points awarded by the aggregate success/failure mode F_{mode_2}, for pairs of reciprocal games during a generational tournament.

Game Pair Outcomes	Fitness Points Awarded
win-win	3
win-draw	1
win-lose	.5
no win	0 (F_{mode_1} dominates)

Note that in cases where no win occurs F_{mode_1} is used to determine a negative fitness value.

3.2 The GA and neural network mutation

After a tournament of games (one generation), controller population members p are scored relative to each other using the performance metric $F(p)$ defined in Equation (1) before propagation to the next generation.

Offspring are generated using mutation only. During the propagation phase of the GA, the fittest 50% of the population produce offspring that replace the least-fit 50% of the population. An important ramification of this is that in the case that 50% or more of the population receives a positive fitness value, then selection will be based entirely on success/failure information and the minimal competence mode will have no bearing, i.e. all individuals not achieving success will be eliminated.

Network weights, neurons and connectivity may all be mutated during propagation.

4. RESULTS

4.1 Incremental Verses All-in-one Evolution

Populations of robot controllers were evolved under two different environmental setting. In one case, controller populations were evolved in increasingly complex worlds over the course of evolution. This case will be referred to as the incremental case. The population of controllers starts evolution in a very simple world. As a level of competence is evolved, the population graduates to progressively more challenging training worlds. The sequence of training worlds is shown in Figure 4 below.
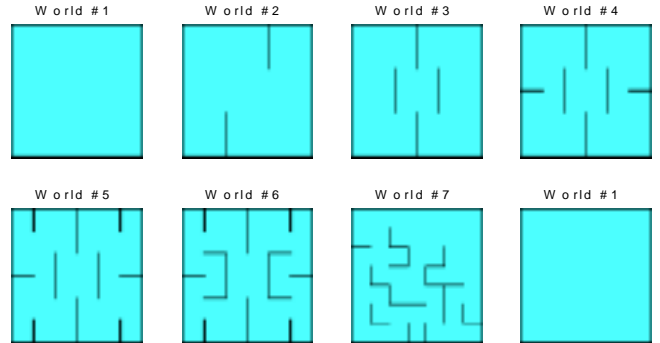

Figure 4. Training worlds used over the course of evolution.

Populations were also evolved completely in a very challenging world configuration (world #7 from Figure 4). This case will be referred to as the all-in-one case. All evolution runs were initialized with populations of randomly configured neural networks. In both cases, the fitness selection function given in Equation (1) from the previous section was used without alteration.

Figure 5 (a) and (b) and show training fitness data from two population evolutions. Figure 5 (a) shows data from an all-in-one evolution case while Figure 5 (b) presents data from an incremental case. In the top panel of (a) and (b), the best controller selection fitness (as measured by the bimodal fitness function) is plotted at each generation over the course of evolution. The population best, mean and poorest fitness levels are all shown. The middle panels of (a) and (b) indicate the number of wins generated by the evolving
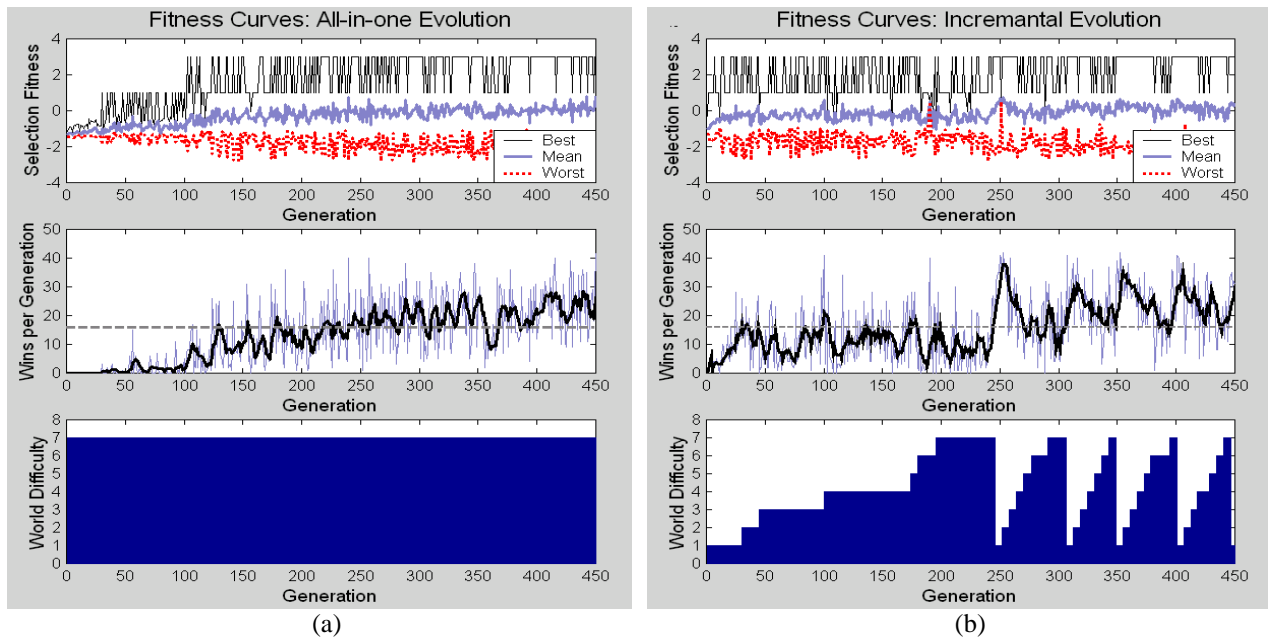
Figure 5. Incremental evolution and all-in-one evolution data curves. (a) shows data training data collected over the course of an all-in-one population evolution. (b) shows data from an incremental evolution.

populations as a whole at each generation. This is a purely passive metric with regard to fitness selection. However, in the incremental case, this passive metric serves to automatically increment the training world difficulty. In Figure 5 (a) and (b) the current training world is indicated at each generation. Note that in (a), this remains at 7 throughout training, whereas in (b) the sequence of environments is stepped through several times over the course of evolution.

In the all-in-one evolution case (Figure 5 (a)), selection fitness was not maximized until after generation 100. This reflects the fact that controllers were evolved from the beginning in the most difficult training world. No controller capable of winning both of its games in a tournament arose before the 100th generation. After the 100th generation the fitness of the best (fittest) controller oscillated between levels corresponding to one and two wins per generation. The total population wins per generation data (middle panel Figure 5 (a)) however, indicate that the overall number of games that a population can win continued to increase over the next 200 generations.

Figure 5 part (b) shows data from an incremental evolution run. In this case the best fitness at each generation was maximized very early in evolution. This indicates that the best individuals in the population evolved relatively quickly to be able to win games in the simpler worlds. Even so, the middle panel of Figure 5 (b) indicates that at the first generation, no controller in the in the initial random seed population was capable of winning a game. By the 25th generation the population as a whole was capable of winning enough games during a tournament to graduate to the next level of training world difficulty. Over the course of training up to the 250th generation, the population became

competent in each of the progressively more challenging environments. The cycle of training worlds repeats after it is completed. The lower panel of Figure 5 (b) indicates that the first cycle through the training world sequence required 250 generations. There after, the population cycled though the sequence completing four full cycles between the 250th 450th generations.

4.2. Evaluation of Evolved Controller Performance

Because a relative competitive fitness selection metric was used to drive the evolutionary process, absolute fitness is not known. To address this, fully evolved controllers were compared to a knowledge-based controller of known abilities.   The fittest controllers from populations evolved under both incremental conditions and in single environments (all-in-one cases) were selected and tested in competition against the knowledge-based controller.
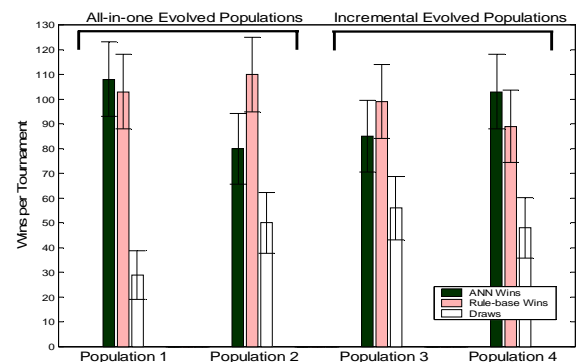

Figure 6.   Results of evolve controller and knowledge-based controller competitions. The dark bars indicate the number of neural controller wins per tournament, the shaded bars indicated the number of knowledge-based wins, whereas the open bars record the number of games played to a draw. The range bars indicate a 95% confidence interval.

Four evolved populations were compared: 2 incrementally evolved populations and 2 all-in-one populations. Using the best controller from each of the evolved populations, an extensive tournament of 240 games was conducted (against the knowledge-based controller). All games were played in a challenging world configuration similar to that shown in world #7 of Figure 4. Each game during the tournaments was initialized with a new randomly generated set of starting positions for robots and goals. Note that the same set of 240 random initializations was used for each of the 4 tournaments.

Figure 6 summarize the results of the four testing tournaments. Controllers from both populations 1 and 4 were able to play competitively against the knowledge-based controller and were able to win modest majorities of games over the course of 240 games. This indicates that incrementation of environmental difficulty produced no clear evolutionary advantage. It is likely that the bootstrap mode of the bimodal fitness selection function compensates for difficult initial training environments. This is further evidenced by comparing the number of wins per generation data from the incremental and all-in-one evolutions. In the all-in-one case, no wins at all occured in early training, while in the incremental case, controllers evolved to produce game wins relatively quickly (early in evolution) in the simpler environments.

5. CONCLUSION

This research investigated the application of a bimodal fitness selection function toward the evolution of team game-playing behaviors in mobile robots.

Although only populations 1 and 4 in Figure 6 produced controllers capable of outperforming the knowledge base, all 4 evolved populations produced relatively competent controllers. This indicates that the bimodal fitness selection function can be used to drive the evolution of game-playing behaviors under the different conditions considered.

Populations of large neural networks using 150 or more processed video sensor inputs were evolved.

Further work will include investigations into a broader range of environmental and algorithmic conditions. For the current research, information from only a single trail (tournament) was used at each generation. Methods for integration of fitness information over the course of evolution will also be investigated.

References

[1] I. Harvey, P. Husbands, D. Cliff, "Seeing the light: artificial evolution, real vision," in D. Cliff, P. Husbands, J.-A. Meyer, S. Wilson Eds, *From Animals to Animates 3, Proc. of 3rd Intl. Conf. on Simulation of Adaptive Behavior, SAB94,* MIT Press/Bradford Books, Boston, MA, 1994, pp. 392-401.

[2] R.A. Watson, S.G. Ficici, J.B. Pollack, "Embodied Evolution: Distributing an Evolutionary Algorithm in a Population of Robots," *Robotics and Autonomous Systems*, vol. 39 no. 1, pp 1-18, Apr. 2002.

[3] J. Kodjabachian, J.-A. Meyer, "Evolution and development of neural networks controlling locomotion, gradient-following, and obstacle avoidance in artificial insects," *IEEE Transaction on Neural Networks*, vol. 9, no. 5, pp. 796-812, Sept. 1998.

[4] D. Floreano, F. Mondada , "Evolution of homing navigation in a real mobile robot," *IEEE Transactions on Systems, Man, Cybernetics Part B: Cybernetics*, vol. 26 no. 3, pp. 396-407, 1996.

[5] M. Quinn, "Evolving communication without dedicated communication channels," in Kelemen, J. and Sosik, P. Eds., *Advances in Artificial Life: Sixth European Conference on Artificial Life (ECAL 2001)*, Prague, Czech Republic, Sept. 2001, Springer, pp. 357-366.

[6] S. Nolfi, "Evolving non-trivial behaviors on real robots," *Robotics and Autonomous Systems*, vol. 22, no. 3-4 pp. 187-198, 1997.

[7] T. Ziemke, "Remembering how to behave: Recurrent neural networks for adaptive robot behavior," in Medsker and Jain Eds., *Recurrent Neural Networks: Design and Applications*, Boca Raton, CRC Press, 1999.

[8] R.A Brooks, "Intelligence Without Representation", *Artificial Intelligence Journal*, vol. 47, pp.139–159, 1991.

[9] G.S. Hornby, S. Takamura, J. Yokono, O. Hanagata, M. Fujita, J. Pollack, "Evolution of Controllers from a High-Level Simulator to a High DOF Robot," *Evolvable Systems: from biology to hardware; proceedings of the third international conference (ICES 2000)*, J. Miller, Ed., *Lecture Notes in Computer Science*, vol. 1801, Springer, 2000, pp. 80-89.

[10] D. Floreano, J. Urzelai, "Evolutionary Robots: The Next Generation," in *The 7th International Symposium on Evolutionary Robotics (ER2000): From Intelligent Robots to Artificial Life*, Gomi T. Ed., pp. 231-266, AAI Books, 2000.

[11] J. Galeotti, The EvBot A Small Autonomous Mobile Robot for the Study of Evolutionary Algorithms in Distributed Robotics, MS Thesis, North Carolina State University, 2002.

[12] J. Galeotti, S. Rhody, A. Nelson, E. Grant, and Gordon Lee, "EvBots – The Design and Construction Of A Mobile Robot Colony for Conducting Evolutionary Robotic Experiments," *Proceedings of the ISCA 15th International Conference: Computer Applications in Industry and Engineering (CAINE-2002)*, pp. 86-91, San Diego Ca, Nov. 7-9, 2002.

[13] A.L. Nelson, E. Grant, T.C. Henderson, "Competitive relative performance evaluation of neural controllers for competitive game playing with teams of real mobile robots," *Measuring the Performance and Intelligence of Systems: Proceedings of the 2002 PerMIS Workshop*, Gaithersburg MD, Aug. 13-15, 2002, pp. 43-50.

[14] A.L. Nelson, "Competitive Relative Performance and Fitness Selection for Evolutionary Robotics", Doctoral Dissertation, North Carolina State University, 2003.