The Effects of Team Size on the Evolution of Distributed Micro Air Vehicles

Daniel J. Collins
Dept. of Electrical Engineering and Computer Science
The University of Kansas
Lawrence, KS 66049
dcollins@eecs.ukans.edu

Arvin Agah
Dept. of Electrical Engineering and Computer Science
The University of Kansas
Lawrence, KS 66049
agah@ukans.edu
Tel.: (785) 864-7752

Annie S. Wu
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816
aswu@cs.ucf.edu
Tel.: (407) 823-5922

Alan C. Schultz
Navy Center for Applied Research in Artificial Intelligence
Naval Research Laboratory
Washington, DC 20375
schultz@aic.nrl.navy.mil
Tel.: (202) 767-2684

Abstract

This paper focuses on the study of genetic algorithms for evolving controllers for Micro Air Vehicles (small flying robots). Specifically, the effects of team size on the evolution of the robots are studied. The study of team sizes deals with finding an ideal number of robots to collectively perform the surveillance task of a given region. In addition, controllers for evolved robot teams are tested in teams of different sizes to determine if the results can be generalized. In addition, statistical analysis is performed to evaluate the evolved results.

1 INTRODUCTION

Although a single ant can perform many amazing tasks such as carrying a food morsel that weighs several times more than the ant's own body weight, it is often even more interesting to look at the work of a colony of ants. A team of hundreds of ants can efficiently work together on tasks such as building anthills. This distributed approach has been adopted in many aspects of the human world and the robotics world. In general, a team of distributed robots can perform various tasks more efficiently than a single robot. Specifically, using a team of robots adds fault tolerance to the system. If a single robot fails to function, then the remaining robots can possibly take over the duties of the failed robot, thus ensuring the team still meets the goal. There are many applications for distributed robotics such as surveillance, exploration of space, or evacuation of land mines.

Another area that is being explored in robotics is the idea of evolving control of the robots. In robotics, the robots are given rule sets that guide the robot to perform certain actions. The robot has a list of conditions that correspond with different actions. Instead of explicitly writing the rule sets for the robots, scientists have explored evolving the rule sets using Genetic Algorithms (GAs). GAs are search methods based on principles from natural selection and genetic reproduction.

This paper deals with the problem of coordinating a team of distributed Micro Air Vehicles (MAVs) to perform surveillance on a given area. MAVs are lightweight, autonomous air vehicles that can be equipped with sensors and payloads for various missions. In this research, we assume the vehicles can detect certain features of the land below the vehicle, and can detect other MAVs within a certain radius of itself. The MAVs' task, as a group, is to fly over an area and perform reconnaissance. They need to maximize the area covered, concentrating on areas of more importance, and minimizing duplication of effort. In previous work, we successfully used GAs to evolve MAV control rule sets that could accomplish the above surveillance task (Wu *et al.*, 1999).

An important issue in design of multi-robot systems is the size and structure of the deployed team. Therefore, the focus will be on the effects of evolution on team size, and how controllers evolved for a certain team size would perform when utilized in a different team size. It should be noted that the team size refers to the number of MAVs deployed in each simulation experiment, and is different from the population size of Genetic Algorithms.

2 EVOLUTIONARY ROBOTICS

Evolutionary robotics is the result of applying the concepts of genetic algorithms to the realm of robotics, evolving brains (controllers) and possibly the body of the robots. There are numerous studies on the use of GAs to evolve the control of robots in many different environments. One environment that has been used to test the success of GAs is robot soccer (Agah and Tanie, 1997). They used GAs to evolve the Tropism-Based control structure, based on “likes” and “dislikes” of the robots in terms of actions that they like to perform. The fitness function that was used rewarded the robots for scoring goals, assisting teammates in scoring goals, and blocking opponent’s shots on goal. They showed that after evolving the robots for many generations, the robots produced a better team than the initial populations. Evolutionary robotics experiments have been performed using physical robots (Floreano and Mondada, 1994). GAs have also been used to evolve neural networks for robot control (Lund and Miglino, 1996). They developed a system using a combination of the physical and simulated world. They developed a physical world for the Khepera robot, and allowed it to go through many different motions. All the sensory data was read from the robot into the computer during this phase. The simulator could then take real sensory and motor data from the physical world, and model those in the simulated world, using them as inputs for the neural network. They argue that this method significantly reduced the amount of time for transition of a simulated system to a physical system.

Another common application for evolutionary robotics is the area of biped locomotion. Researchers have developed a Steady State Genetic Algorithm to search for the necessary torques for moving each joint of the robot leg in the appropriate path (Rodrigues *et al.*, 1996). They modeled the robot in software as a connection of five rigid bodies and two revolutionary joints for each of the legs. One revolutionary joint was for the knee and the other was for the hip. Each simulation was allowed only three seconds to carry out the desired action. They studied the complexity of the problem in that it took 2000 iterations for the robot to learn how to stand. They repeated the walking experiments with varying velocities, showing the success and failures of the application of genetic algorithms.

There are many other examples of related evolutionary robotics research (Billard et al., 1999) (Bonabeau et al., 1999) (Floreano and Mondada, 1998) (Floreano and Mondada, 1994) (Harvey, 1996) (Nolfi and Parisi, 1995). Utilization of GAs in evolving robot controllers has also been investigated in: (Cliff *et al.*, 1993) (Deneubourg *et al.*, 1991) (Grefenstette *et al.*, 1990) (Shibata and Fukuda, 1993) (Ueyama *et al.*, 1992).

The work in this paper extends previous work in a number ways. First, this work is oriented towards control of distributed unmanned air vehicles, not land-based vehicles. In addition, this study will focus on the dynamics of evolving rule sets for teams of varying size.

3 MICRO AIR VEHICLES

3.1 ROBOTS

The micro air vehicles used in these experiments are set to realistic parameters based on prototypes, as described in (Wu *et. al*, 1999). The robots in the experiment are circular and have a radius of five units. The MAV is equipped with eight sensors, giving it a sensing radius of 30 units. The sizes are parameters that can be changed for each experiment. These sensors can be used to detect other robots and the boundary of the survey area; however, the sensors simply return one bit, 1 or 0, depending on if something is present or not. In other words, the MAVs detect other MAVs and the boundary edges, but the MAVs will not be able to distinguish between another MAV and a boundary edge of the environment.

In addition to sensing other robots and the boundary, each robot has eight sensors that can survey a circular area on the ground below the robot, with a total radius of 15 units. The ratio of the sensing radius compared to the survey radius ensures that it is possible that no two robots will overlap surveying the ground. In other words, a robot will be able to detect another robot, preventing the same area of ground being covered by both robots. However, this depends on the control system of the MAVs and their efficiency in generating the behavior of the MAVs.

3.2 CONTROL SYSTEM

The control of the MAV is governed by rule sets consisting of the typical condition/action rules. The action a robot takes is based on the robot’s sensory information. As mentioned earlier, the robot has eight sensors for sensing other robots, each returning a bit for whether an entity is detected or not. Also, there are eight sensors for surveying the land. The eight survey sensors return two bits corresponding to the level of topology on the ground that is detected. Bit values representing the topology states are: 00 for level 0 (no topology), 01 for level 1 topology, 10 for level 2 topology, and 11 is not used. These topologies represent the level of interest in the location, i.e., places that are of more interest would have a higher topology value.

The 8 bits from sensing and the 16 bits from the surveying make up the condition component of the condition-action pairing. The action portion of the rule is made up of four bits. The first bit tells whether the robot should move forward or hover in place. The other three bits correspond to a turn direction. The directions the robot can turn are the eight compass directions (N, NE, E, SE, S, SW, W, and NW). Table 1 shows all the three bit combinations that make up the degree change in direction.

The condition and action clauses are paired together to make one MAV rule of 28 bits. Table 2 shows an example of a MAV rule broken into the condition and action components. The sensors are numbered from zero to seven as the condition component in Table 2 is read from right to left. The sensors 0, 2, and 4 have detected another entity; thus, they each registered a one for the sensing data and the other sensors return zero. The topology sensors return varying values. Sensors zero through two detect topology level two while the remaining sensors detect topology level one. The bits in the action clause tell the MAV to move forward and turn +45 degrees, for that specific condition.

Table 1: Bit Values for Relative Turn Direction

BIT VALUES	RELATIVE TURN DIRECTION (DEGREES)
000	-135
001	-90
010	-45
100	0
110	+180/-180
101	+45
011	+90
111	+135

Table 2: An example MAV rule.

CONDITION		ACTION	
8 BITS FOR SENSING	16 BITS FOR TOPOLOGY	MOVE	ANGLE
101010000	1010100101010101	1	101

A robot's rule set is made up of one or more of these rules. The sensors are queried at each time step of the simulation and the sensory information is compared with the condition clauses of the rules. The condition clause with the best degree of match to the sensory data is selected. The degree of match is the difference between the condition component of the rule and the current sensor values. In the case that multiple rules qualify for the best match, then one of the best matching rules is chosen at random. If the degree of match is within a certain threshold, then the action clause of the rule is fired. In the simulation, every MAV is able to collect sensory information, determine the action to take, and execute the action. Although in the physical environment some moves may take longer than others may, in the simulation all moves are assumed to take one time step, thus simplifying the simulation task.

3.3 ENVIRONMENT

As previously mentioned, there are three different levels of topology (Wu *et al.*, 1999). The levels of topology correspond to how interesting is the section of land. In other words, if the MAV is used for military surveillance then a section of land that contains a military base or airport is perhaps more interesting or more relevant than a flat, dry plain with farms. Therefore, in this example, the military base could be classified as a level one or level two and the flat plain could be classified as a level zero or no topology.

The topology is normally defined in the following fashion. First, the boundaries of the survey area are defined and everything within is given the level zero topology value. The boundaries are defined so that if a MAV tries to go outside the survey area then the MAV is considered inoperable and can no longer function. Rectangular level one topologies can be defined within the survey boundaries. Smaller, rectangular level two topologies can be defined within the rectangular, level one topologies. Figure 1 describes a topology with three level two topologies within a large level one topology. The level two topologies have one larger topology to the north of two smaller topologies.

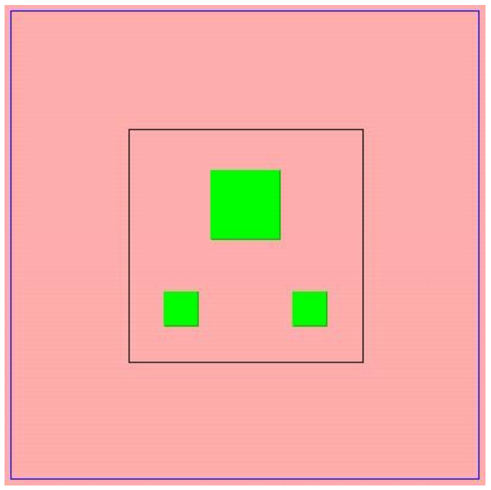

Figure 1: The Defined Topology

3.4 SIMULATION

In the experiments, each MAV is configured with eight sensors, has a radius of five, survey range of 30, and a sensor range of 50 units. In the setup files, the rule set for each MAV is defined. All of the MAVs have the same rule set in the experiments, i.e., a homogenous team of robots. This was done because it is more difficult to evaluate the individual success of a robot in a heterogeneous team of robots.

Using the simulator, each robot checks its sensory data during each time step and compares the data to the list of conditions in the rule set. When the sensory data matches a certain condition, then the corresponding action is fired. Since, there are 24 bits of sensory data then there are 2^{24} or 16,777,216 possible conditions. In the experiments, a maximum and a minimum rule set sizes have been set.

To evaluate the success of the team of MAVs, the percentage of surveillance area (with the added weight value for more significant topology) is calculated for all the MAVs. The value is averaged over the entire experiment time steps and is used as the final metric for evaluating the success of the team, representing the fitness function of the Genetic Algorithm. It should be noted that the credit assignment problem is eliminated using this approach, since the performance of the entire team is considered instead of individual robots (homogeneous members).

The simulator runs without animation to speed up the testing cycle. However, configuring the input files to print a trace can produce a trace file that holds information about where the robots are located and what they are surveying during each time step. If the option to print a trace is enabled, then a file is generated from the simulation program that shows the position of every robot at each time step. This file can then be used to view the animation through a Java applet.

3.5 ANIMATION

A Java applet was developed to view the simulation of the MAVs. The applet displays data from trace file, as generated by the simulation. The applet provides the options of running, pausing, and stepping through the animation. In addition, the animation may be reset at any time or change the speed in which it executes. In the animation, the MAVs are started on the west boundary of the survey area. As mentioned earlier, a thin-lined box denotes the boundary area and the areas of interest are the rectangular shapes within the survey area. The screenshot illustrating the animation using the Java appletviewer is shown in Figure 2. The MAV is the inner of the three circles. The second circle represents the area of the ground that the MAV can survey and the third circle represents the MAV sensor range for detecting the boundaries and other MAVs. The thin line from the center of the MAV to the edge of the MAV indicates the current direction that the robot is heading.

3.6 EVOLUTION

A GA was used to evolve rule sets for controlling teams of MAVs. Each individual in a GA population represents a complete rule set. The fitness of a GA individual (a rule set) is determined by the performance of a team of MAVs using that rule set in the simulation described in section 3.4. As a result, the GA must execute one MAV simulation for every individual in its population. Table 3 gives the parameter settings for the GA that we used to learn the rule sets. We used a population size of 100 individuals and each run evolved for 150 generations. The evolved rule sets could vary in length (i.e. the number of rules varied); however, their size was limited by a maximum and minimum length. For example, the maximum length of an individual in our experiments is 2800 bits. Since each rule is 28 bits long, this translates to 100 rules. The minimum length allowed in these experiments is 10 rules. Initial chromosome length refers to the length of the individuals in the initial population. One point random crossover was used; a single crossover point was randomly and independently selected on each parent, resulting in variable length individuals. Crossover can only occur in between rules; crossover will never occur within a rule. Mutation modifies individual bits and is consequently able change the rules themselves. The parsimony pressure refers to the amount of negative weight in the fitness function that penalizes the MAVs with the longer rule set. This value ranges from zero (no parsimony pressure) up to the maximum chromosome length.

The fitness of a particular rule set is determined by testing its effectiveness in the MAV simulation. For any GA run, the MAV parameters are held constant throughout the entire run. Table 4 gives the parameter values used in our MAV simulations. It is necessary to specify the number of MAVs, the number of sensors, the number of bits for the condition, the number of bits for the action, the size of the MAVs, the sensor range, the survey range of the MAVs, and the initial positions of the robots. An additional specification that is needed is the topology that describes the environment to test the MAVs.

Table 3: GA Parameters

PARAMETER	VALUE
Population size	100
Max number of generations	150
Initial chromosome length	1680
Max chromosome length	2800
Min chromosome length	280
Parsimony pressure	0
Crossover type	one point
Crossover rate	1.0
Mutation rate	0.005

Table 4: MAV Parameters

PARAMETER	VALUE
MAV team size	Variable
Survey range	15
Sensor range	30
Number of sensors	8
Radius	5
Condition length	24
Action length	4
Spacing	5

4 EXPERIMENTS

4.1 EXPERIMENT SETUP

The experiments were run on Sun Ultra 5/333 MHz Workstations. Each experiment was run on different Sun Solaris machines. On average, a Sun Ultra 5/333 can evolve a population of 100 MAVs for 150 generations in just over 24 hours. The experiments were typically divided among six to seven different machines. Tables 3 and 4 show the GA parameters and MAV parameters, respectively. In the experiments, the topology description is fixed. The topology used has a 400x400 region with a level one 300x300 topology centered in the middle of the region. Three level two topologies were placed within the level one area. A larger area is to the north of two smaller areas. Figure 1 shows the topology.

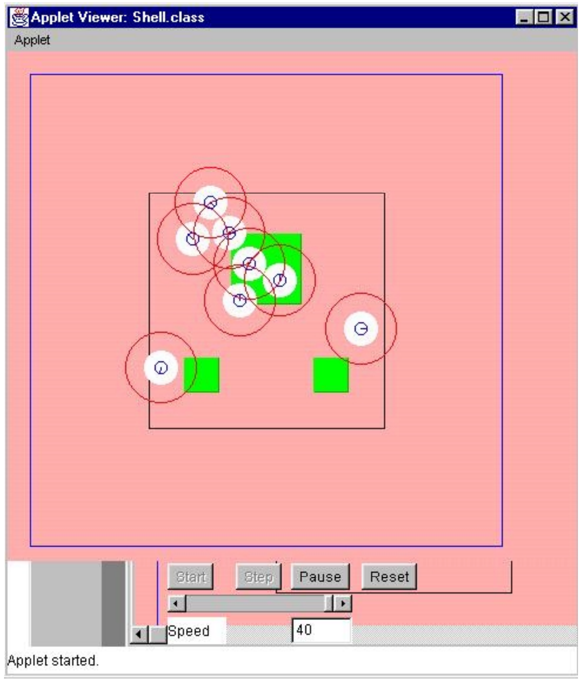

Figure 2: The MAV Animation

In all the experiments, the data presented concerning the evolution program is based on three runs, i.e., the average of three repetitions of the experiments. Furthermore, all data that is gathered solely from the simulation program are based on 10 runs. The experiments were repeated to increase the reliability of the results.

4.2 TEAM SIZE EXPERIMENTS

A series of experiments focused on investigating the effect of the MAV team size on the on the GAs ability to evolve effective and robust rule sets. The term *team size* refers to the number of MAVs used in a particular MAV simulation. For instance, a MAV team size of eight means that eight MAVs will be used to survey the land in that simulation. It should be noted that, during a simulation, the MAVs could collide or leave the surveillance area, rendering them inoperable, reducing the number of MAVs operating for the rest of the simulation. Since a single GA run uses the same MAV simulation parameters throughout its entire run, the MAV team size is fixed for individual GA runs.

Using the GA and MAV parameters given in tables 3 and 4, we evolved rule sets for MAV team sizes ranging from 2 to 20 MAVs per team. Each experiment used a different MAV team size. Each experiment averaged the performance of three GA runs. The goal of these experiments was to see if these tests would yield an ideal team size for the simulated task. Figure 3 shows the resulting fitnesses of the rule sets evolved in our experiments. The fitness of the evolved rule sets decreases as the number of MAVs reached 20. The descent in the fitness values appears to start when the GA is evolving rule sets for teams of 10 or more MAVs. We hypothesize that this decrease may be due to the environment becoming too crowded. The optimal number of MAVs for the environment size that we tested appears to be around six. With a small number of MAVs (e.g. two) the fitness value seemed relatively high; however, the deviation between the best fitness and the worst fitness is significantly larger.

4.3 SCALABILITY OF EVOVLED RULE SETS

The previous set of experiments looked at evolving rule sets for a fixed, known team size. A more interesting problem involves evolving rule sets that will work well for unknown or a range of team sizes. That is, we would like to evolve an effective rule set for controlling a team of MAVs in a particular task; however, the actual number of MAVs that will be available may not be known at the time that we are learning the rule set. Is it possible to evolve rule sets that will work well for team sizes other than the one used during its evolution?

In this second set of experiments, rule sets were evolved using team sizes of 2, 5, 10, 15, and 20 MAVs. The best rule set from generation 150 of these runs was then selected to be tested in simulations using teams of 2, 5, 10, 15, and 20 MAVs. Each selected rule set was tested10 times with each team size. Figure 4 shows the fitness of robots using the various rule sets and the various testing population sizes. These results suggest that it is better to evolve rule sets using larger MAV team sizes if the actual deployed team is unknown in advance. For each rule set selected from a GA with a fixed team size, Table 5 shows the fitness or performance of that rule set in simulations using other team sizes.

In addition, the average fitness/performance of each rule set over all team sizes is calculated, excluding the diagonal elements, where the number of MAVs were the same for evolved and simulated cases. As shown in Table 5, the average performance of a rule set that was evolved using a team size of 20 is much better than the average performance of a rule set evolved using a team size of two. Performance gradually increases as the evolving team size increases. We speculate that lack of interaction between MAVs in small team sizes may play a large role in these results. The statistical analysis in the next will be used to support this hypothesis.

4.4 STATISTICAL ANALYSIS

Statistical analysis was performed to help support or deny the hypothesis that it is better to evolve larger population of MAVs. The Student's t test is an acceptable method for comparing the two groups of data (Caprette, 1998). The t test determines if the two populations are the same based on the variable data that is collected.

The t test was applied to the values from the evolved rule sets of two MAVs and five MAVs. Table 6 shows the fitness data points collected and the mean and standard deviation of the two samples. Calculating the *A* value for the two samples yields 8/16 or 0.5. Calculating the *B* value for the two samples yields 88.35. Therefore, the value of *t* is 3.37. With six degrees of freedom, the critical value is 2.45 for a probability of 0.05. The *t* value 3.37 is greater than 2.45 so we can confidently reject the null hypothesis. In other words, we can confidently say that the two groups are different. Applying the t test to the other comparisons yields similar results. The *t* values from comparing the various samples are listed in the Table 7. Looking up the confidence probability on a table of critical values yields Table 8. From Table 8 it is important to realize that the t tests that allow the rejection of the null hypothesis concern the rules evolved for two MAVs and any other rule set. The cells with the star indicate the null hypothesis cannot be rejected. The preliminary result is that t tests determined there is a significant probability that the rule sets for 10 MAVs and 20 MAVs are different. The remainder of the evolved rule sets for 5, 10, 15, and 20 MAVs provide too similar results to reject the null hypothesis. It can be argued that the rule sets evolved for five MAVs and up include similar interactions with other robots when compared with the larger rule sets.

5 CONCLUSION

In this paper, we investigate the use of a GA to evolve rule sets for controlling teams of distributed micro air vehicles (MAVs). We looked at two main aspects of this topic: (1) does the size of the team affect the effectiveness of the rule sets that can be evolved, and (2) can we evolve rule sets that will work in simulations using team sizes different from the team size that was used during the evolution of the rule set. Our results suggest that it is more difficult to evolve effective rule sets for larger team sizes than smaller team sizes. We speculate that this difficulty arises from the fact that rule sets for larger team sizes must be able to deal with more interactions than rule sets for smaller team sizes. When the actual size of the team is unknown in advance, however, it appears to be better to evolve rule sets using larger team sizes. The larger sized teams appear to provide the learning process with more instances of interactions, allowing the GA to evolve a more complete rule set.

Table 5: Fitness of Evolved Rule Sets in Different Population Sizes

# MAVS TESTED IN SIMUL.	# MAVS EVOLVED IN GA				
	2 MAVs	**5 MAVs**	**10 MAVs**	**15 MAVs**	**20 MAVs**
2	51.899	49.242	33.493	49.481	46.835
5	27.294	56.871	48.835	57.608	56.268
10	17.724	52.456	55.056	52.901	55.074
15	19.143	36.032	44.694	43.008	47.906
20	11.489	27.587	32.795	33.120	37.314
AVERAGE	18.91	41.32	39.95	48.278	51.521

Since the experiments were carried out in a simulated world, many limitations were not accounted for in these tests. Many of the variables of the physical world did not come into play in the simulated world. For instance, the robots were limited to two-dimensional motion. In the physical world, the robot would need to be able to ascend and descend vertically in the environment along with moving horizontally. In addition, other constraints on the robot were not taken into consideration such as battery levels. In the real world, the robots would base their control system not only on the sensing and survey data, but also on the internal sensors such as battery usage. Finally, the surveying of the robot was greatly simplified in the simulation. In the real world, the robots would have to perform image recognition or be given satellite maps and corresponding topology levels.

There are many new directions that this project's future work can take. First, the dynamics of the physical world and the constraints of the robots can be added to the current software package. Three-dimensional maps can be added to the environment along with additional sensors for the robots to handle the three-dimensional world. A new environment can be developed, complete with buildings and recharging stations for the robots. Additionally, the information from these experiments can be tested in the physical world. Many other areas would need to be explored and implemented such as image recognition. Implementing these experiments in the real world would convincingly show the feasibility of using the genetic algorithm to evolve the rule sets for teams of flying robots.

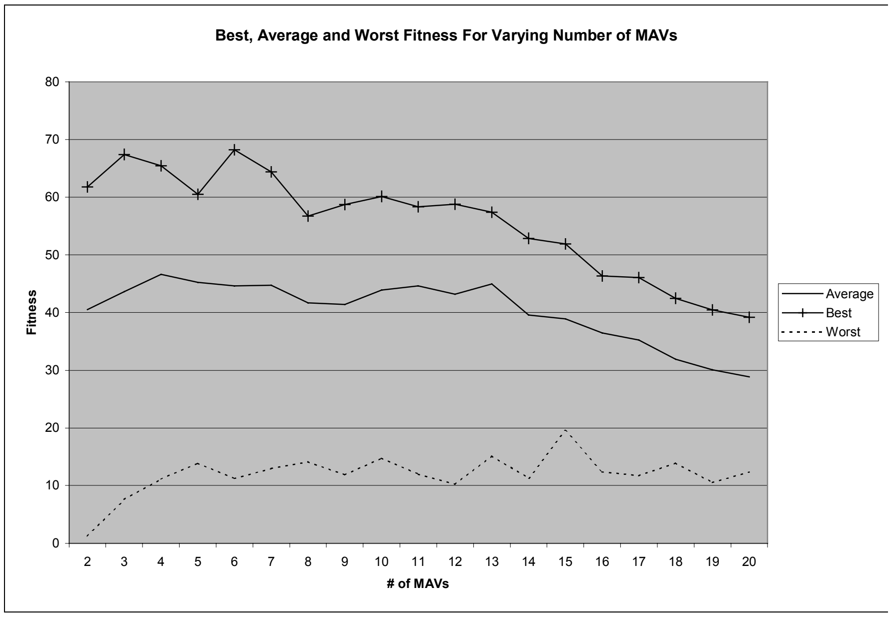

Figure 3: Fitness Function for Varying Number of MAVs

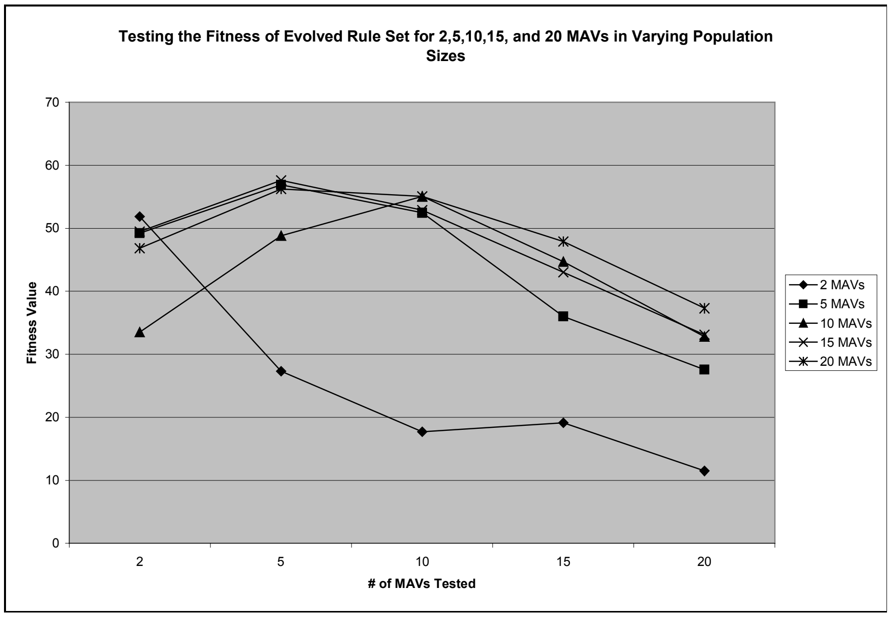

Figure 4: Fitness of Evolved Rule Sets in Different Population Sizes

References

Agah, A., and Tanie, K. (1997). Robots Playing to Win: Evolutionary Soccer Strategies. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 632-637.

Billard, A., Ijspeert, A.J., and Martinoli, A. (1999). Adaptive exploration of a dynamic environment by a group of communicating robots. In *Proceedings of the 7th European Conference on Artificial Life*, Lausanne.

Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York.

Caprette, D.R. (1998). Student's t test for independent samples. http://www.ruf.rice.edu/~bioslabs/tools/stats/ttest.html.

Cliff, D., Harvey, I., and Husbands, P. (1993). Explorations in evolutionary robotics. *Adaptive Behavior*, Vol. 2, 73-110.

Deneubourg, J.-L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chretien, L. (1991). The dynamics of collective sorting: robot-like ants and ant-like robots. In Meyer, J.-A. and Wilson, S.W. (Eds.) *From Animals to Animats*. MIT Press, Cambridge, Massachusetts, 356-363.

Floreano, D., and Mondada, F. (1998). Evolutionary Neurocontrollers for Autonomous Mobile Robots. *Neural Networks*, Vol. 11, 1461-1478.

Floreano, D. and Mondada, F. (1994). Automatic creation of an autonomous agent: genetic evolution of a neural network driven robot. In Cliff, D., Husbands, P., Meyer, J.-A., and Wilson S.W. (Eds.) *From Animals to Animats III*, MIT Press, Cambridge, MA.

Grefenstette, J.J., Ramsey, C.L., and Schultz, A.C. (1990). Learning sequential decision rules using simulation models and competition. *Machine Learning*, Vol. 5, 355-381.

Harvey, I. (1996) Artificial evolution and real robots. In *Proceedings of International Symposium on Artificial Life and Robotics*, Beppu, Japan, 138-141.

Lund, H.H., and Miglino, O. (1996). From simulated to real robots. In *Proceedings of the IEEE Conference on Evolutionary Computing*, 362-365.

Nolfi, S., Parisi, D. (1995). Evolving non-trivial behaviors on real robots: An autonomous robot that picks up objects. In Gori, M. and Soda, G. (Eds.) *Topics in Artificial Intelligence, Proceedings of the 4th Congress of the Italian Association of Artificial Intelligence*, Springer-Verlag, Berlin, 243-254.

Rodrigues, L., Prado, M., Tavares, P., da Silva, K., and Rosa, A. (1996). Simulation and control of biped locomotion-GA optimization. In *Proceedings of the IEEE Conference on Evolutionary Computing*, 390-395.

Schultz, A.C. (1994). Learning robot behaviors using genetic algorithms. In *Proceedings of the First World Automation Congress*, 607-612.

Shibata, T. and Fukuda, T. (1993). Coordinative behavior in evolutionary multi-agent robot system. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 448-453.

Ueyama, T., Fukuda, T., and Arai, F. (1992). Structure configuration using genetic algorithm for cellular robotic system. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1542-1549.

Wu, A.S., Schultz, A., and Agah, A. (1999). Evolving control for distributed micro air vehicles. *In Proceedings of the 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '99)*, Monterey, California, November 1999.

Table 7: Fitness Data for Two MAVs and Five MAVs

# OF MAVS TESTED	**2 MAVS**	**5 MAVS**
2	-----	49.242
5	27.294	-----
10	17.724	52.456
15	19.143	36.032
20	11.489	27.587
Mean	18.91	41.32
Standard Deviation	6.5	11.59

Table 8: T Values for Varying Numbers of MAVs

# OF MAVs	**2**	**5**	**10**	**15**	**20**
2	-----	3.373	4.067	4.710	8.048
5	3.373	-----	0.195	0.883	1.622
10	4.067	0.195	-----	1.248	2.463
15	4.710	0.883	1.248	-----	0.555
20	8.048	1.622	2.463	0.555	-----

Table 9: Probabilities of Two Similar Rule Sets

# OF MAVs	**2**	**5**	**10**	**15**	**20**
2	-----	0.025	0.010	0.005	0.005
5	0.025	-----	*	*	*
10	0.010	*	-----	*	0.050
15	0.005	*	*	-----	*
20	0.005	*	0.050	*	-----