HR0011-09-1-0045
2008 DARPA CSSG Final Technical Report
(For Phase II, Ending August 19, 2011):
Scalable Heterogeneous Multiagent Teams Through Learning Policy Geometry

Kenneth O. Stanley (`kstanley@cs.ucf.edu`**)**
Dept. of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816 USA

1 Introduction

This document is the final technical report for Phase II of the DARPA Computer Science Study Group (CSSG) program started by the PI in the year 2008. (Phase II itself began for the PI in 2009.) It follows the reporting requirements specified in the award document for the project.

2 A comparison of actual accomplishments with the goals and objectives established for the grant, the findings of the investigator, or both.

In Phase I of the DARPA CSSG we developed an early version of a new algorithm for training multiple robotic agents to coordinate with each other called *multiagent HyperNEAT* (D'Ambrosio and Stanley 2008). This approach built upon *Hypercube-based NeuroEvolution of Augmenting Topologies* (HyperNEAT), a new algorithm for evolving artificial neural networks that we had introduced shortly before (D'Ambrosio and Stanley 2007; Gauci and Stanley 2007, 2010; Stanley et al. 2009). The HyperNEAT algorithm has the interesting property that it can generate the weights of neural connections based on the locations of the nodes they connect, which means that in effect it generates connectivity based on geometry. This capability led to they key insight behind Multiagent HyperNEAT that it may be possible to generate a *set* of neural networks based on each network's position on a virtual field. For example, the positions are like the positions of soccer players on a soccer team, with forwards exhibiting offensive tactics and fullbacks more defensive. In a similar way, HyperNEAT could be used to generate a whole team of neural networks that share skills yet also exhibit role-specific behaviors. Such controllers, trained (i.e. not programmed directly) through evolution, could in principle be deployed in real robots and UGVs to perform tasks such as room clearing or building patrol autonomously. The aim of Phase II was to turn this idea into reality.

The goals and objectives in the original Phase II proposal were organized into four milestones over two years. These milestones are reviewed next in sequence.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
OCT 2011		**00-00-2009 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Scalable Heterogeneous Multiagent Teams Through Learning Policy Geometry	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Central Florida,Dept. of Electrical Engineering and Computer Science,Orlando,FL,32816**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

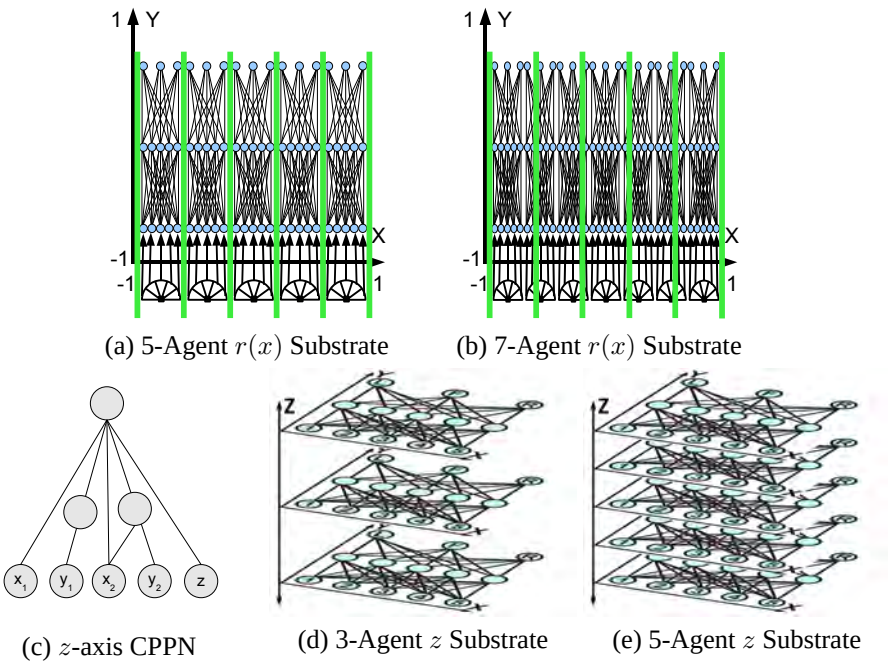
12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**7**	



(a) 5-Agent $r(x)$ Substrate (b) 7-Agent $r(x)$ Substrate

(c) z-axis CPPN (d) 3-Agent z Substrate (e) 5-Agent z Substrate

Figure 1: **Old and New Multiagent HyperNEAT Geometries.** The original formulation (called $r(x)$) shown in (a) placed ANNs on the substrate side-by side. The problem with this approach is that when new networks are added to scale the size of the team up (b), the new arrangements overlap with the old divisions. (Notice that the agent ANNs in (b) cross the original dividing lines between agents.) By introducing a z-axis in the new formulation (c), it becomes possible to scale from e.g. three agents (d) to five agents (e) without any overlap. In this way, no matter how many agents are introduced, they will never overlap along the z-axis.

2.1 Milestone 1: Revising Policy Geometry Encoding

The first milestone was to revise the original policy geometry encoding based on an improved geometric conception of how the artificial neural networks (ANNs) should be arranged in space (which is called the *substrate* in HyperNEAT). Figure 1 gives a sense of how the geometry was revised.

This reorganization, called the *revised policy geometry encoding*, was completed and successful in a variety of tasks. It became the standard encoding that we continued to adopt throughout Phase II and to this day as we enter Phase III. While the original encoding was published in GECCO-2008 (and won a Best Paper Award there) (D'Ambrosio and Stanley 2008), the new encoding was introduced at AAMAS-2010 (D'Ambrosio et al. 2010), a top-tier conference on autonomous agents and multiagent systems. Both publications demonstrated the approach in a coordinated predator-prey task but the AAMAS publication also added room clearing, taking a step towards real-world DoD-related applications. In this way, Milestone 1 was fully satisfied.

2.2 Milestone 2: Key Extensions

This second part of the project focused on extensions to the multiagent HyperNEAT approach. The first proposed extension, seeding, meant starting multiagent evolution from a pre-trained single individual. The idea is that the pre-trained seed would have specific skills (e.g. chasing a prey in predator-prey) from which the entire team might benefit. This idea was successfully demonstrated in various predator-prey variants, first at GECCO-2008 (D'Ambrosio and Stanley 2008) and later in a more sophisticated version of the task that was compared to the SARSA reinforcement learning algorithm. This later demonstration is in submission at present at the Journal of AI Research (JAIR). The main result is that it is easier to train multiagent teams from a pre-trained seed ANN than from scratch, confirming the hypothesis behind seeding.

The second proposed extension was called *multi-dimensional policy geometry* in the original proposal but was changed it to *situational policy geometry* later, which is a similar idea but with more specific meaning. The main

idea is that not only are there ANNs for different agents, but also for different situations in which the agents might find themselves. This conceptual geometry in effect expands the dimensions in the original policy geometry to include not just different positions on a team but different situations for the same position. The result is that the same agent in effect possesses several "brains," one for each situation it might need to confront. This idea is published in IROS-2011, a major conference on robotics (D'Ambrosio et al. 2011). The publication includes a real-world demonstration with Khepera III robots (as was proposed). Videos of this demonstration with real robots are at: `http://eplex.cs.ucf.edu/patrolling.html`

The final proposed extension was called a *hive brain*. The idea was to connect neurons between *different* agents so that they can communicate neural signals among each other through wireless connections. The interesting thing about such communication is that it is at the level of neural signals rather than through any particular language. Preliminary results were promising. However, the hive brain is actually a highly ambitious and complex endeavor, encompassing novel neural configurations, wireless communications, and multiagent coordination; it also opens up new domains. Thus instead of turning it into a publication, we made it part of our Phase III proposal to ARO, which was accepted (as was the DARPA portion, which focuses on simulation to real-world transfer). Thus the hive brain has grown to become a major initiative in Phase III.

It is also important to note that during this time we also began to work with ARL to show that our trained ANN controller can indeed work in their Packbots. While this particular exercise in real-world transfer was not specifically articulated in the original proposal, it became apparent as we approached the Phase III proposal that ARL needed to see that such transfer would work before endorsing further collaboration (which would be a goal for Phase III). Thus during this period we also built a Packbot simulator because ARL's current experimental UGV is the Packbot. We then ran a successful test in which we evolved a hall navigation controller for a Packbot with HyperNEAT in our simulator and transferred it to a real Packbot at ARL, which proceeded to navigate a hallway at the ARL location. This demonstration confirmed that our infrastructure produces controllers that can work in the real world, including at ARL. This work was performed in collaboration with Stuart Young and Dave Baran at ARL.

In summary, two of the three extensions were completed and the other grew to become a basis of the Phase III project. Furthermore, we successfully transferred a controller trained in our simulator to a real Packbot at ARL.

2.3 Milestone 3: Applications and Major Scaling

The focus of this milestone was on scaling and DoD-related applications. One of the key motivations for multiagent HyperNEAT was its scalability. Because an entire team could be described through a single compact encoding, it should be possible to train very large teams. Put another way, multiagent HyperNEAT actually learns a *mapping* between position on a team and ANN policy. Thus in principle it should be possible to sample as many positions as desired within this mapping, e.g. hundreds of them, and still obtain functional teams. Two kinds of scalability are relevant: One is *pre-training scaling*, which means that the size of the team that is trained can be be very large. This kind of scaling is important because most traditional multiagent training techniques struggle with training very large teams (Conitzer and Sandholm 2007; Littman 1994; Singh et al. 2000; Stone and Sutton 2001). The other kind of scalability is *post-training scaling*, which means that new roles can be *interpolated* for agents at positions that were not initially trained. Multiagent HyperNEAT can perform such post-training interpolations based on the policy geometry it learned during training. While role interpolation is a heuristic, in some tasks it may be a useful way to add new agents to a team on the fly. The goal was to train and scale teams up to a size of 1,000 agents.

Both kinds of scaling were tested extensively over the course of the project. Successful results from post-training scaling are published in D'Ambrosio et al. (2010) in two domains. Teams are scaled to sizes of up to 1,000 agents. Videos from these experiments are at: `http://eplex.cs.ucf.edu/mahnaamas2010.html`

However, a larger and more extensive study of scaling with multiagent HyperNEAT in a version of the multiagent predator-prey domain is in our current journal submission to JAIR. In this paper currently under review, multiagent HyperNEAT is compared to multiagent SARSA in both pre-training and post-training scaling. The paper emphasizes the significant advantage for multiagent HyperNEAT in pre-training scaling as team size grows larger. SARSA does not have a mechanism for role interpolation so it naturally does not do as well at post-training scaling, but it also lags multiagent HyperNEAT in pre-training scaling. Although these results are currently under review, videos of the comparisons can be seen at:

http://eplex.cs.ucf.edu/comparison.html

One interesting issue that proved often controversial with reviewers is the generality of post-training scaling. Although many teams trained over the course of our project did scale well post-training, reviewers often point out that in some domains roles may not be possible to interpolate post-training because of complex nonlinearities in the way different roles cooperate with each other. This point is important because it highlights that while post-training scaling is an unusual capability that is useful in particular domains, the more general practical benefit across many domains may be in pre-training scaling, i.e. the ability to train medium or large teams without confronting the curse of dimensionality.

During this period we also began to investigate the new application domain of *patrol and return* (i.e. it is a new domain beyond the original room-clearing domain) in which a team of robots fans out in a building and then individual robots return to the entrance when called back (e.g. to have their batteries recharged). This domain was ultimately demonstrated successfully in the previously-mentioned IROS paper that also examined situational policy geometry (D'Ambrosio et al. 2011). Interestingly, the learned policies generalized to different building maps as well. As noted above, videos of robots in this domain are at: http://eplex.cs.ucf.edu/patrolling.html

Overall then both significant scaling of various types and training in a new domain succeeded.

2.4 Milestone 4: Room Clearing with Real Robots

The original proposal was to culminate with a room clearing experiment with real Khepera robots. The goal is that the Khepera III robots learn to enter a room and spread out along its perimeter on their own. The procedure is to train the team in our custom-designed simulator (created for Phase II) with only five agents and then to transfer it to a version of the domain in the real world. Then the team is scaled to seven agent *after training* (i.e. post-training scaling), forcing HyperNEAT to interpolate roles in the room-clearing team for the additional two agents. This interpolated scaling is first tested in the simulator and then in the real world.

We were able to achieve this entire sequence successfully. A video documenting the team both in the simulator and in the real world is at: http://www.youtube.com/watch?v=2VaDtU5XVC8

Note that the patrol and return domain was also (in addition to room clearing) validated in the real world, as shown (noted again) at:
http://eplex.cs.ucf.edu/patrolling.html

Both of these real-world tests took significant engineering, testing, and design. We learned a great deal about what it takes to move controllers out of simulation and into the real world, knowledge that will serve us well in Phase III.

Thus we were able to demonstrate room clearing, patrol and return, and scaling up team size without further training, all trained in simulation and transferred to real-world robots. These accomplishments show that multiagent HyperNEAT indeed works in the real world. Furthermore, both these tasks are relevant to real DoD-related domains, which in principle thus could be tackled by military robots such as Packbots. Our additional successful transfer of a hallway navigation ANN to a real Packbot at ARL further supports that the work completed here creates significant potential for multiagent learning in DoD-related applications. This enterprise now continues with Phase III of the CSSG, which includes a grant from ARO (Grant No. W911NF-11-1-0489) and the DARPA match (Grant No. N11AP20003).

2.5 Additional Accomplishments

Several other significant technologies were developed and expanded in support of the Phase II work. This section documents this supporting work.

First, the *novelty search* method, which was introduced by my group shortly before Phase II started (Lehman and Stanley 2008, 2011a), has proven an effective alternative to traditional objective-based search in some domains. It also was useful in many of our experiments with multiagent HyperNEAT as an alternative means to exploring the behavior space in various domains. Over the course of Phase II, a number of enhancements and explorations were made: an extension called *minimal criteria novelty search* was introduced (Lehman and Stanley 2010b), novelty search was proven in genetic programming (Lehman and Stanley 2010a), it was shown to help in evolving adaptive ANNs (Risi et al. 2010a, 2009) (the 2009 paper won another Best Paper Award), its creativity was demonstrated through evolving

virtual creatures (Lehman and Stanley 2011b), and it was shown to yield more evolvable genomes (Lehman and Stanley 2011c). Furthermore, we completed a journal article on novelty search (Lehman and Stanley 2011a) (started before Phase II) and shared it with the genetic programming community (Lehman and Stanley 2011d). This approach is a significant new area for evolutionary computation in its own right and magnifies the impact of the Phase II work.

Second, we implemented several enhancements to the underlying HyperNEAT algorithm that can also apply to multiagent HyperNEAT. HyperNEAT is a novel approach to evolving ANNs that opened many new directions for investigation in its own right, some of which were exploited in this supporting work: HyperNEAT was extended to evolve *plastic ANNs*, i.e. ANNs whose synapses change over their lifetime, and HyperNEAT was also extended to decide the placement of density of hidden neurons (which it could not do before) in a new version called evolvable substrate HyperNEAT (ES-HyperNEAT) (Risi et al. 2010b; Risi and Stanley 2011) (the 2010 publication won another Best Paper Award). Both these extensions are built into our multiagent simulator (built for Phase II) and can be run with multiagent HyperNEAT. The simulator is freely available at:
`http://eplex.cs.ucf.edu/software.html`

These enhancements and extensions are important to the progress of the field of neuroevolution and HyperNEAT in general, and are now available to contribute to Phase III.

2.6 Summary

Overall, almost every milestone was fulfilled in its entirety. The only task still ongoing beyond Phase II is the hive brain, but it ultimately formed the basis for the new proposal to ARO (now funded), which means it was a successful stepping stone to further advancement as well. Furthermore, many extensions and enhancements were also completed and published that went beyond what was initially proposed.

3 Reasons why established goals were not met, if appropriate.

No significant goals were unmet.

4 Other Pertinent Information

The research completed in Phase II now has set the stage for Phase III. Thus this research stream is in effect still ongoing. Phase III includes funding from DARPA and ARO. The goals are to enhance simulation to real-world transfer and to build upon the hive brain concept to enable new DoD-relevant applications, which will include collaboration with ARL.

References

Conitzer, V., and Sandholm, T. (2007). AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1):23–43.

D'Ambrosio, D., Lehman, J., Risi, S., and Stanley, K. (2011). Task switching in multirobot learning through indirect encoding. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (IROS 2011)*. Piscataway, NJ: IEEE. To appear.

D'Ambrosio, D., Lehman, J., Risi, S., and Stanley, K. O. (2010). Evolving policy geometry for scalable multiagent learning. In *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2010)*, 731–738. International Foundation for Autonomous Agents and Multiagent System.

D'Ambrosio, D., and Stanley, K. O. (2007). A novel generative encoding for exploiting neural network sensor and output geometry. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007)*. New York, NY: ACM Press.

D'Ambrosio, D. B., and Stanley, K. O. (2008). Generative encoding for multiagent learning. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2008)*. New York, NY: ACM Press.

Gauci, J., and Stanley, K. O. (2007). Generating large-scale neural networks through discovering geometric regularities. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007)*. New York, NY: ACM Press.

Gauci, J., and Stanley, K. O. (2010). Autonomous evolution of topographic regularities in artificial neural networks. *Neural Computation*, 22(7):1860–1898.

Lehman, J., and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In Bullock, S., Noble, J., Watson, R., and Bedau, M., editors, *Proceedings of the Eleventh International Conference on Artificial Life (Alife XI)*. Cambridge, MA: MIT Press.

Lehman, J., and Stanley, K. O. (2010a). Efficiently evolving programs through the search for novelty. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2010)*. ACM.

Lehman, J., and Stanley, K. O. (2010b). Revising the evolutionary computation abstraction: Minimal criteria novelty search. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2010)*. ACM.

Lehman, J., and Stanley, K. O. (2011a). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223.

Lehman, J., and Stanley, K. O. (2011b). Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2011)*. ACM.

Lehman, J., and Stanley, K. O. (2011c). Increasing evolvability through novelty search and self-adaptation. In *Proceedings of the 2011 Congress on Evolutionary Computation (CEC-2011)*. IEEE.

Lehman, J., and Stanley, K. O. (2011d). Novelty search and the problem with objectives. In *Genetic Programming Theory and Practice IX (GPTP-2011)*. Springer.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning: Proceedings of the 11th Annual Conference*, 157–163. San Francisco: Kaufmann.

Risi, S., Hughes, C. E., and Stanley, K. O. (2010a). Evolving plastic neural networks with novelty search. *Adaptive Behavior*, 18(6):470–491.

Risi, S., Lehman, J., and Stanley, K. O. (2010b). Evolving the placement and density of neurons in the hyperneat substrate. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2010)*. New York, NY: ACM Press.

Risi, S., and Stanley, K. O. (2011). Enhancing es-hyperneat to evolve more complex regular neural networks. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO 2011)*, 1539–1546. New York, NY, USA: ACM.

Risi, S., Vanderbleek, S. D., Hughes, C. E., and Stanley, K. O. (2009). How novelty search escapes the deceptive trap of learning to learn. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2009)*. New York, NY, USA: ACM Press.

Singh, S., Kearns, M., and Mansour, Y. (2000). Nash convergence of gradient dynamics in general-sum games. In *In Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*.

Stanley, K. O., D'Ambrosio, D. B., and Gauci, J. (2009). A hypercube-based indirect encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212.

Stone, P., and Sutton, R. S. (2001). Scaling reinforcement learning toward RoboCup soccer. In *Proc. 18th International Conf. on Machine Learning*, 537–544. Morgan Kaufmann, San Francisco, CA.