Developmental Scalable Hierarchies for Multi-Agent Swarms*

Terence Soule
Department of Computer Science
University of Idaho
Moscow, ID 83844-1010
tsoule@cs.uidaho.edu

Robert B. Heckendorn
Department of Computer Science
University of Idaho
Moscow, ID 83844-1010
heckendo@cs.uidaho.edu

ABSTRACT

Hierarchical control structures for multi-agent systems represent a promising middle ground between fully-distributed systems and centralized control. In this paper we present a developmental approach for evolving hierarchical control structures for large (100-800 member), multi-agent swarms. The results show that this approach can successfully generate control hierarchies that improve the performance of fully distributed swarms and that scale well.

Categories and Subject Descriptors

I.2.2 [**Artificial Intelligence**]: Automatic Programming—*Program synthesis*

General Terms

Algorithms

Keywords

multi-agent swarms, scaling, developmental evolution

1. INTRODUCTION

The use of multi-agent systems, in which complex problems are addressed by large numbers of relatively simple agents, is currently one of the most promising approaches in the field of computational intelligence. Fully distributed systems, in which each agent has limited knowledge of the overall problem and communicates only with its nearest neighbors, have advantages in learning, robustness. However, there is growing empirical and theoretical evidence that for many problems centralized control can significantly improve performance. Unfortunately, centralized control largely nullifies the advantages in learning, robustness, and scalability seen in fully distributed systems. Hierarchical control systems, in which control agents make decisions based on abstracted information supplied by agents "beneath" them, represent a promising intermediate approach.

*This material is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

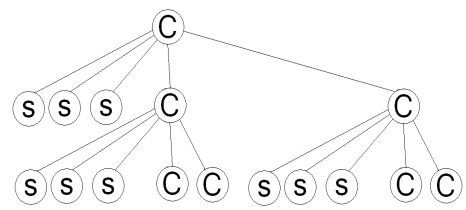


Figure 1: Sample control hierarchy. Nodes marked S represent subswarms of s agents. Nodes marked C are control nodes. The number and size of the subswarms and the number of control nodes at each level are evolved. The hierarchy is defined by an L-system style grammar and can expand to include all available agents.

However, designing a hierarchical control system requires parallel development of both the hierarchy and the agents within the hierarchy in a way that fosters cooperation. This makes automated learning of multi-agent systems with hierarchical control structures very difficult. In this paper we present a developmental evolutionary approach to evolve scalable control hierarchies. The evolved control hierarchies perform well and, more importantly, scale well.

2. EXPERIMENT

Our test problem is the Mixing Problem, a variant of standard swarm dispersion problems [2, 1], in which two or more groups of agents of different types start in different locations and must disperse throughout the entire area (Figure 2). When expanding groups of agents meet they inhibit each others' progress, potentially forming Nash equilibrium in which the movement of any agent decreases the dispersion distance. Simply ignoring the other agent type will lead to collisions, which are unacceptable in the model.

Our hierarchical approach evolves control hierarchies (defined by an evolved L-system grammar, see Figure 1) that apply directive forces to the agents within the hierarchy. The control parameters defining the directive forces are also evolved. However, the behavior of the search agents is not evolved to assure that any observed benefits are strictly due to the use of a control hierarchy.

The evolutionary algorithm uses a standard steady-state, real-valued GA model to evolve the parameters that define both the L-system and the parameters of the control agents (see Table 1). Figure 2 (left) shows a sample result with two agent types using the standard fully distributed rule in

Parameter	Initial Range	Mutation Stepsize	Allowed Range
Controllers/controller	2-5	(-1,1)	$(1,\infty)$
Agents/subswarm	10-25	(-6,6)	$(10,\infty)$
Subswarms/controller	1-10	(-1,1)	$(1,\infty)$
k_i	0-20	(-0.05,0.05)	$(-\infty,\infty)$

Table 1: Evolved parameters, including the initial random ranges and the mutation ranges. k_i represents all eight parameters that influnce the control hierarchy rules.

Population size	80
Trials	15
Crossover rate	100% Uniform Crossover
Mutation rate	0.3
Selection	3 member tournament

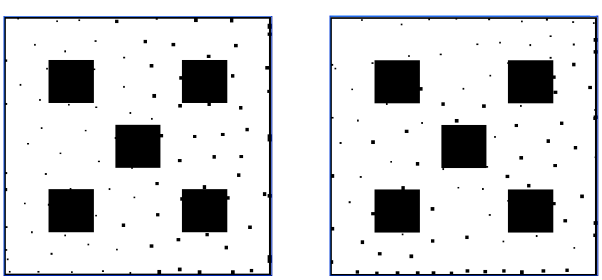
Table 2: GA parameters.

which each agent avoids its three nearest neighbors [1]. The agents disperse rapidly until the types meet, at which point a fairly stable boundary forms.

In these experiments two types of scaling are tested: increasing the *number of types* of agents (Table 3) and increasing the *number of agents* of each type (Table 4). Fitness is the average separation between agents. Two sets of control rules are compared: the fully distributed rule - move away from the three nearest agents; and the evolved control hierarchy. The hierarchy, and associated control parameters, are evolved with two agent types and with 50 agents of each type. The scaled results are generated by testing the best evolved hierarchies.

2.1 Results

Table 3 shows the results of the first experiment - scaling the number of agent types. This increases both the total number of agents and the heterogeneity of the problem, because there are more agent types to 'mix'. The evolved hierarchy produces significantly better results at all scales - showing that the evolved control hierarchy improves mixing and that the improvement is preserved across multiple



Fully Distributed - mostly segregated

Evolved Hierarchy - mixing

Figure 2: Results with two agent types, 100 bots and 500 timesteps. The agent types are represented by small and medium sized squares (the large blocks are fixed obstacles). The fully distributed swarms (left) disperse, but do not mix, the swarms with an evolved control hierarchy (right) do mix.

Agent Types	Fully Distributed	Evolved Hierarchy	Percent Improvement
2	0.212 (0.003)	0.277 (0.0.003)	30.7
4	0.163 (0.003)	0.194 (0.008)	19.0
6	0.139 (0.002)	0.166 (0.009)	19.4
10	0.111 (0.002)	0.131 (0.010)	18.0

Table 3: Results when increasing the number of agent types from 2 to 10. Values in parentheses are standard deviations. Evolved hierarchies perform significantly better than swarms using a fully distributed rule (Student's 2-tailed t-test $p < 0.05$).

Agents per Team	Fully Distributed	Evolved Hierarchy	Percent Improvement
50	0.212 (0.003)	0.277 (0.0.003)	30.7
100	0.141 (0.0017)	0.161 (0.006)	14.2
200	0.092 (0.0007)	0.105 (0.0036)	14.1
400	0.061 (0.0005)	0.067 (0.0016)	9.8

Table 4: Results when increasing the number of agents. Values in parenthesis are standard deviations. The evolved hierarchies perform significantly better than the swarms using a fully distributed rule (Student's 2-tailed t-test $p < 0.05$)

scales. Figure 2 shows sample results with and without an evolved control hierarchy.

Table 4 shows the results of the second experiment - two types of agents, but increasing the number of agents per type. Overall the average separation decreases as the number of agents increases, because there are more agents of each type limiting the maximum possible distance between them. However, at all scales the evolved hierarchies are significantly better at mixing the hierarchies than the fully distributed rule.

3. CONCLUSIONS

This paper presented a developmental approach for evolving scalable control hierarchies for multi-agent swarms, which successfully evolves hierarchical structures that improve the performance over fully distributed rules. Further, the evolved hierarchy naturally extends to larger swarms and scales well along two dimensions: heterogeneity and size of the swarm. Overall these results show that developmental evolution is a promising method for evolving hierarchical control structures that both combine the strengths of fully distributed and centralized swarms and that have the scaling properties necessary for successful 'training in the small, for deployment in the large'.

4. REFERENCES

[1] J. McLurkin and J. Smith. Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots. In *7th International Symposium on Distributed Autonomous Robotic Systems (DARS)*. Springer, 2004.

[2] T. Soule and R.B. Heckendorn. A developmental approach to evolving scalable hierarchies for multi-agent swarms. In *Proceedings of the 12th annual conference comp on Genetic and evolutionary computation*, pages 1769–1776. ACM, 2010.