Exploring Interaction, Diversity and Efficiency of Biologically Inspired Evolutionary Multiagent Systems

Sadat Chowdhury
The Graduate Center
The City University of New York
New York, NY USA
sadatc@gmail.com

Elizabeth I. Sklar
Department of Computer Science
University of Liverpool
Liverpool, UK
e.i.sklar@liverpool.ac.uk

ABSTRACT

This short paper presents work exploring properties of an evolutionary multiagent system assigned to solve sequential task achievement problems in dynamic, real-time, asynchronous environments. Several evolutionary models have been implemented and experiments conducted in a high-performance computing environment in which different interaction mechanisms and population diversity modes are evaluated according to multiple performance metrics. Results are presented that illustrate differences in performance efficiency when different interaction mechanisms, population diversity and evolutionary models are employed.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Intelligence—*multiagent systems*

General Terms

Theory

Keywords

multiagent systems, evolutionary algorithms, genetic programming, speciation, coordination

1. INTRODUCTION

In [9], it was hypothesized that the performance of a multiagent team is affected by both population diversity and level of interaction, such that if interaction mechanisms were employed effectively, a population of heterogenous, single-function agents could generally outperform a homogeneous population of multi-function agents. To test this hypothesis, a series of experiments was run in which the population heterogeneity and interaction mechanism were varied in a simulated multiagent de-mining task environment. Three agent capabilities (*exploration*, *extraction* and *transportation*) and two interaction mechanisms (*broadcast* and *stigmergy*) were implemented. Several sets of experiments were conducted; and although trends appeared to confirm the hypothesis, the results were statistically inconclusive due to large variances in the experimental data.

Here, we continue this line of investigation, within a new simulation framework, called *synthScape*, deployed in a high-performance computing environment that supports extensive experimentation, with the goal of obtaining statistically significant results that will confirm the original hypothesis. In addition, we consider evolutionary methods for varying population composition, in order to evaluate a range of heterogeneous teams.

Our investigation is conducted on an instance of a foraging problem where a population of agents collects *resources* from geospatially dispersed locations and brings them to designated *collection sites*. The problem involves a sequence of *detection*, *extraction* and *transportation* tasks. The same agent need not be responsible for completing all three tasks; indeed, having the tasks completed simultaneously by different coordinating agents may produce more efficient solutions. Coordination may be achieved by agents communicating to indicate that one task in the sequence has been completed at a particular location and the next task can be attempted. The complexity of problems in the sequential task domain may increase with the number of tasks. For example, a foraging problem that requires additional tasks such as refinement and processing of resources is a more complex problem than one that requires only 3 tasks. Such additional tasks would require corresponding traits (also called capabilities or functions), and in general, solving k-task problems in the sequential task domain will require k traits. Such problems can be solved by heterogenous populations of simpler specialist agents or homogenous populations of more complex generalist agents. Stated formally, a k-task problem in the sequential task domain can be solved by (A) a heterogenous population of n-trait agents where $n < k$, or (B) a homogenous population of k-trait agents[1].

We focus here on addressing two research questions:

1. Will a heterogeneous or homogeneous population evolve to be more effective in completing the tasks? For the foraging problem, the number and rate of resources detected, extracted and collected (quantified as *Capture Rate*) are possible ways efficiency may be measured. Such measures of efficiency may depend on population-related factors, such as *species diversity.*

2. Which coordination factors of the evolving popu-

[1]When $n > k$, agents have additional capabilities that are not used in solving problems in the domain.

.

BICT 2014, December 01-03, Boston, United States

DOI 10.4108/icst.bict.2014.257922

lations impact the overall efficiency? When agents coordinate to collect resources, interactions among them may significantly affect their collection efficiency. Communication mechanisms can be characterized by varying levels of interaction. Presumably, to achieve the same collection efficiency, heterogenous populations may require different levels of interaction than homogenous populations.

1.1 Species (Agent Diversity)

Borrowing ecological terms (see [6]), the *species diversity* of a population is defined in terms of the number of unique species, or its *species richness (S)*, and a relative measure of how equally abundant each species is, or its *species evenness (E)*. We define the species of an agent according to its traits: agents having exactly the same traits, $\{Trait_i, Trait_j, \ldots, Trait_k\}$, belong to the same species, $Species_{\{i,j,\ldots,k\}}$. Furthermore, we use the ecological concept of species diversity to mean agent diversity, and will primarily be concerned with the richness (S) component of the diversity.

Several measures, or *indexes*, have been devised to measure species diversity and evenness. The following measure of evenness [5], is based on Simpson's dominance index: $E = \left(1/\sum_{i=1}^{S} p_i^2\right)/S$, where p_i is the proportion of species i, and S is the species richness. E can range from 0, when there is only one dominant species in the population, to 1, when all species are equally abundant in the population. Our measure of species diversity (H) is based on Shannon's entropy [8]: $H = -\sum_{i=1}^{S} p_i ln(p_i)$, where higher values of H are indicative of both greater richness and evenness in the population; if there is only one species, H approaches 0.

The experiments in the research presented here consider two types of populations: (a) heterogenous populations with varied species richness, where $(S, H, E) = (3, 1, 1.098)$; and (b) homogenous population, where $(S, H, E) = (1, 1, 0)$. To understand the impact of the evenness component of diversity, a future set of experiments will consider various species ratios.

1.2 Interaction Mechanisms

Different types of interaction mechanisms can facilitate communication amongst agents. During an interaction, signals are transferred from agents in sender roles to agents in receiver roles. The receiving agents may ignore the signals or decide to act on them. The work presented here considers signals with elementary semantics that indicate the state of detected resources and, in some cases, their locations; the resources can be in one of three states: FOUND-STATE (thus, ready to be extracted), EXTRACTED-STATE (thus, ready to be processed), or PROCESSED-STATE (thus, ready to be transported). The interaction mechanisms are defined as follows:

– **No Interaction.** This indicates an absence of any interaction mechanism. The performance data obtained from experiments with no interaction is used as a baseline to compare with different forms of interaction.

– **Trail.** Senders have the ability to leave "trail signals" on the grid (a form of stigmergy). These signals degrade over time; and before they disintegrate completely, receivers have the ability to detect the signals. There can be delay from the time the signal is sent to the time it is received.

– **Broadcast.** Senders have the ability to broadcast signals to receivers within a certain range. Broadcast signals are received instantly.

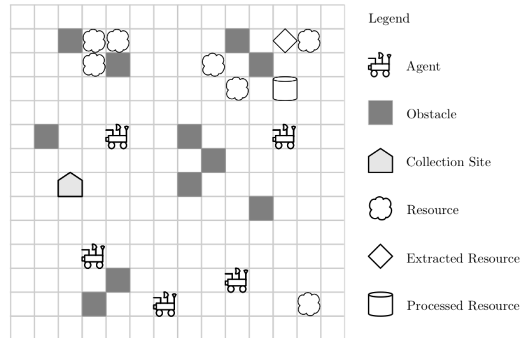

Figure 1: *synthScape* visualization

– **Unicast-Closest-Agent.** Senders have the ability to send signals to the closest receivers; if multiple receivers are equally close, one will be chosen at random. Unicast signals of this type are received instantly.

2. EXPERIMENTAL SETUP

We conducted a series of experiments using *synthScape* (depicted in Figure 1), which we developed in MASON, a discrete event simulation platform [4]. Populations of agents are evolved using Genetic Programming (GP) techniques. The control program (i.e., genotype) of an individual agent is constructed from a custom extension of the Push language [10, 11]. Push programs are series of instructions that are executed by a stack-based virtual machine. Push was designed specifically for evolutionary computation, and thus has two big advantages: (1) any set of Push instructions makes a valid program, so that genetic reproduction operators cannot produce incomplete or invalid programs; and (2) programs can modify themselves by allowing instructions to manipulate the code stack (where instructions reside), which can potentially introduce complex control strategies. In our implementation, each instruction is atomic and either pushes values onto a code stack, executes instruction(s) form the code stack, or executes domain-specific instructions (i.e., sense, move, communicate).

Our current version of *synthScape* implements two population diversity modes (heterogeneous and homogeneous) and four interaction types (detailed above). In addition, any of three different *evolutionary models* (described below) can be used to generate populations. Because of the intense computational requirements necessary to produce statistically significant results, *synthScape* has been designed to be distributed across several processors in a *High Performance Compute Cluster (HPCC)*[2]. Each node in the HPCC executes one experimental scenario at a time. The current implementation distributes 10 batches of each scenario, to be run in parallel on each of 24 experimental conditions (2 population diversity modes $\times 4$ interaction types $\times 3$ evolutionary models)—dedicating 240 processors—each batch repeats the experimental condition several times with different starting conditions (see below).

2.1 Evolutionary Models

There are numerous evolutionary models designed to evolve

[2] `http://wiki.csi.cuny.edu/cunyhpc/`

genotypes, but all follow the same basic algorithm. First, randomly instantiate a population of genotypes and maintain a collection of genotypes (the *gene pool*). Second, *evaluate* members of the gene pool in an environment and rank them using a *fitness function*; then select top-ranked members. In a given gene pool, the agent with the highest fitness is defined to be the *alpha* agent. Third, apply *reproduction* operators, such as mutation and crossover, to the selected genotypes to produce the next *generation* of genotypes. Refresh the gene pool with the new generation. Finally, repeat the evaluation, selection and reproduction steps until a satisfactory level of fitness has been achieved or a maximum number of generations has elapsed.

Specific evolutionary models vary in their methodologies and this may impact the behavior of genotypes and, thus, the performance of the agents. The models also vary in their implementation requirements, and depending on the problem context, some models might be more appropriate and feasible than others. The models in *synthScape* are:

– **Population Islands (PI) Model** [7, 2]. Separate populations of genotypes are co-evolved in separate gene pools, each dedicated to evolving genotypes with particular sets of traits. The concept is analogous to different species evolving in different islands. A global fitness function is used to measure the overall performance of the system and the fitness value is shared across all the gene pools. This fitness value is then used to produce the next generation of genotypes for each gene pool, and the process is repeated until a certain number of generations have been evolved. This model is characterized by a centralized fitness evaluation, a global fitness function, and synchronous evolution of each generation. Some major disadvantages of this model are that it suffers from the credit-assignment problem, it is usable only in simulate-and-transfer methodologies, and its synchronous and centralized nature impose high computational demands and restrict its scalability to small, less complex problems.

– **Embodied Evolution (EE) Model** [12, 1]. The concept of evolution is embodied within the agents. An agent maintains its own gene pool, runs its own evolutionary algorithm and evolves its own genotype with a particular set of traits. When the control code (genotype) has finished its execution, it is replaced by the next generation's control code, as produced by the evolutionary algorithm. Each agent also evaluates its own fitness, based on criteria appropriate to its traits. For instance, an agent with detection and extraction traits will evaluate its performance based on resources it was able to detect and extract. This model is characterized by local fitness functions, decentralized fitness evaluations and asynchronous evolution of each generation. Since each agent maintains its own gene pool, agents belonging to the same species may follow different evolutionary paths and exhibit non-uniform behavior.

– **ALife (AL) Model.** This model is similar to the previous in terms of implicitly embodying an evolutionary process within each agent. In addition to maintaining independent gene pools, agents belonging to the same species, situated in close proximity, are able to copy each others' genotypes to add to their own gene pools. The concept is similar to mating in nature and may allow the transfer of useful genotypes across gene pools and unify the behavior of agents belonging to the same species. Other scale-related benefits can be realized through an extended form of this basic model [3].

3. RESULTS

Our experiments involved an average of 250 runs for each of the 24 experimental conditions. Each run used a different *starting condition*—initial locations of agents, resources and collection sites; a fixed *benchmark* starting condition was repeated every 100 generations. Each experiment used the following parameters:

dimensions of simulated world	=		16×16
population size	=		$\{8, 24\}$
resource capture goal	=	12	(75%)
obstacle density	=	32	(12.5%)
resource density	=	16	(6.25%)
collection site density	=	4	(1.56%)

We examine our results through the lens of our two research questions: (1) Will a heterogeneous or homogeneous population evolve to be more effective in completing the tasks? and (2) Which coordination factors of the evolving populations impact the overall efficiency? The plots below contain data averaged across all the runs executed for experimental condition across all scenarios ($91-476$ runs/scenarios per condition).

Effectiveness. Figure 2 illustrates the best results for the number of resources captured under each experimental condition. The goal was to capture 12 resources, which was achieved in some of the best cases. On average, 4 resources were captured (averaged over all the runs). The left column of plots shows the results for heterogeneous populations, while the right column shows results for homogeneous populations. Each row illustrates the application of a different evolutionary model. The curves within each subplot show the differences in performance for different interaction mechanisms. The best results are obtained with heterogeneous teams interacting via trail signals, evolved using the Alife evolutionary model. While the Island model does not appear to learn at all, the heterogeneous Embodied Evolution and ALife models learn to perform significantly better than their homogeneous counterparts.

Efficiency. Figure 3 illustrates the rate of interaction—number of interaction instructions, i.e., messages sent—across the different experimental conditions. Clearly the heterogeneous populations learn to be much more efficient with their interactions over time. Looking at this figure in tandem with Figure 2, it is clear that there is a significant shift in resource capture rate around generation 100 with Embodied Evolution, versus generation 50 with the ALife model.

4. SUMMARY

We have presented preliminary results from a series of experiments designed to evaluate the performance of a multiagent population in a sequential multi-task domain and the impact of population diversity, interaction mechanism and evolutionary model. Populations of varying diversities (heterogeneous vs homogeneous) were evolved using three evolutionary models (Population Island vs Embodied Evolution vs ALife) and four interaction mechanisms (none, trail, broadcast, unicast). The heterogeneous teams that evolve using the Embodied Evolution and ALife models learn to collect resources more effectively than the homogeneous teams; and the ALife model learns to communicate more efficiently.

5. ACKNOWLEDGEMENTS

This research was supported, in part, by a grant of com-

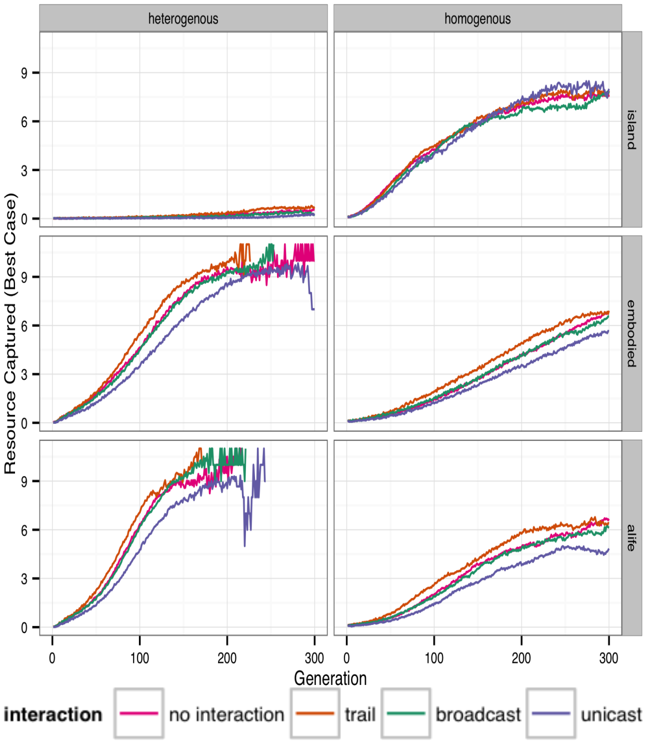

Figure 2: Resource Capture rate (y-axis): for all the evolutionary models, agent diversities and interaction types.

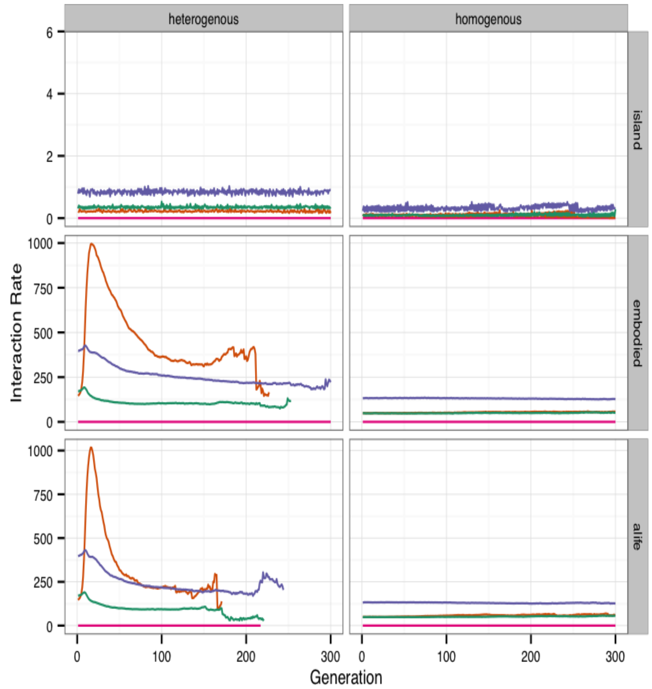

Figure 3: Interaction rates: number of interaction instructions issued during each generation shown for all the evolutionary models, agent diversities and interaction types.

puter time from the City University of New York High Performance Computing Center under NSF Grants CNS-0855217, CNS-0958379 and ACI-1126113; as well as a University of Liverpool Research Fellowship.

6. REFERENCES

[1] R. Bianco and S. Nolfi. Toward open-ended evolutionary robotics: evolving elementary robotic units able to self-assemble and self-reproduce. *Connection Science*, 16(4):227–248, 2004.

[2] S. M. Gustafson. *An analysis of diversity in genetic programming*. PhD thesis, Univ of Nottingham, 2004.

[3] J. L. J. Laredo, D. L. González, F. F. de Vega, M. G. Arenas, and J. J. M. Guervós. A peer-to-peer approach to genetic programming. In *Proc of the European Conference on Genetic Programming (EuroGP)*, volume LNCS 6621, pages 109–118. Springer Verlag, 2011.

[4] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan. Mason: A multiagent simulation environment. *Simulation*, 81(7):517–527, 2005.

[5] C. P. H. Mulder, E. Bazeley-White, P. G. Dimitrakopoulos, A. Hector, M. Scherer-Lorenzen, and B. Schmid. Species evenness and productivity in experimental plant communities. *Oikos*, 107(1):50–63, 2004.

[6] R. K. Peet. The measurement of species diversity. *Annual review of ecology and systematics*, 5:285–307, 1974.

[7] M. A. Potter and K. A. De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000.

[8] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.

[9] E. I. Sklar, M. Schut, K. Diwold, and S. Parsons. Exploring coordination properties within populations of distributed agents. *AAAI Spring Symposium on Distributed Plan and Schedule Management*, 2006.

[10] L. Spector. Autoconstructive evolution: Push, pushgp, and pushpop. In *Proc of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 137–146, 2001.

[11] L. Spector, J. Klein, and M. Keijzer. The push3 execution stack and the evolution of control. In *Proc of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 1689–1696, 2005.

[12] R. A. Watson, S. G. Ficici, and J. B. Pollack. Embodied evolution: Embodying an evolutionary algorithm in a population of robots. In *Proc of the Congress on Evolutionary Computation (CEC)*, volume 1, pages 335–342, 1999.