The Legion System: A Novel Approach to Evolving Heterogeneity for Collective Problem Solving

Josh C. Bongard[1]

AI Lab, Computer Science Department
University of Zurich, Winterthurerstrasse 190
CH-8057, Zurich, Switzerland
bongard@ifi.unizh.ch

Abstract. We investigate the dynamics of agent groups evolved to perform a collective task, and in which the behavioural heterogeneity of the group is under evolutionary control. Two task domains are studied: solutions are evolved for the two tasks using an evolutionary algorithm called the Legion system. A new metric of heterogeneity is also introduced, which measures the heterogeneity of any evolved group behaviour. It was found that the amount of heterogeneity evolved in an agent group is dependent of the given problem domain: for the first task, the Legion system evolved heterogeneous groups; for the second task, primarily homogeneous groups evolved. We conclude that the proposed system, in conjunction with the introduced heterogeneity measure, can be used as a tool for investigating various issues concerning redundancy, robustness and division of labour in the context of evolutionary approaches to collective problem solving.

1 Introduction

Investigations into heterogeneous agent groups are only just getting under way. To cite two examples, in [20], morphological heterogeneity is studied, in which physical robots have non-overlapping sets of sensors and effectors; in [3], physical and simulated robots with distinct motor schemata are referred to as behaviourally heterogeneous groups.

These studies stand in contrast to biological models, such as action selection [17] and behaviour thresholds [24, 7], in which the underlying control algorithms of the agents are equivalent, but changes to the parameters of an agent's control algorithm lead to behavioural differentiation. Agents in [22] exhibit large morphological and behavioural variation, but this variation, in the context of collective problem solving, was not addressed.

In biological systems, individual cells in an organism contain (near-) identical genomes; although individual organisms within a species exhibit differing alleles, the actual gene complement across organisms within a species is the same. In contrast, evolutionary algorithms are not limited by this constraint: evolved agent groups can exhibit large behavioural differentiation. To this end, the work presented here is concerned with the dynamics of behaviourally *heterogeneous* groups, in which not only the observed behaviours, but also the underlying control architectures of the agents are differentiated.

In this report, simulated agents are studied. However, there is a growing body of literature dedicated to heterogeneous robot groups. Arkin and Hobbs [1] delineate a number of dimensions along which enlightened design of robot groups should proceed. Mataric *et al* have implemented groups of robots in which heterogeneity is realized through spatial differentiation within the task space in order to minimize physical interference [10, 11], or by implementing a dominance hierarchy, in which inferior robots can only perform a subset of the basic behaviours available to more dominant robots [16].

These studies, however, take a simplistic view of heterogeneity, in that the differences between agents in the group are decided upon by the designers. For example, in the case of territoriality, each agent is assigned its own area prior to execution of the task. It has been pointed out [3] that most of this work is also simplistic in that heterogeneity is treated as a binary property. In a series of studies [3], groups of robots learned to perform a collective task by tuning the heterogeneity of the group to best perform the task. For foraging and cooperative movement tasks, it was shown that groups invariably converge on homogeneous behaviours; in the case of robot soccer, the teams converge on heterogeneous behaviours [3].

Although these studies were concerned with the degree of heterogeneity in a group as a consequence of the task domain, emphasis was placed on exercising a measure for heterogeneity called *social entropy* [5]. Herein it is shown that by using an evolutionary approach to heterogeneous group behaviours, a simplified measure of heterogeneity can be formulated which overcomes some of the drawbacks of social entropy, explained in Sect. 2.

Evolutionary approaches to heterogeneity include the work by Bull and Fogarty [8], who present an island-model genetic algorithm that encodes classifier systems used to control a quadruped robot; in [21], cascade neural networks [9] are evolved for parity computation using an incremental genetic algorithm. In both investigations, however, the behavioural niches of the groups are predetermined.

In [14], a genetic programming approach is introduced in which niche determination is more dynamic: behaviours are evolved for a pride of lions in a predator/prey task domain. Each individual s-expression in the GP population codes for each and all of the behaviours required by members of the pack. The merit of evolving team behaviours, as opposed to evolving individual behaviours which are later combined to form a team, is pointed out in [12]: individual-level evolutionary systems must somehow overcome the credit assignment problem.[1]

[1] The credit assignment problem also appears in learning approaches to group heterogeneity. This problem, as noted in [4], prompted the development of a new (and heavily domain specific) type of reinforcement learning heuristic, *shaped reinforcement learning* [15].

In Luke and Spector's model, the behaviour for each individual lion in a pride is represented as a branch in an s-expression which encodes all of the behaviours for the team. This model successfully avoids the credit assignment problem, and allows for emergent problem decomposition: the amount of divergence (and convergence) between the behaviours of the individual lions is shaped by the selection pressure exerted by the predator/prey task domain. However, this model suffers from two serious drawbacks.

First, more diverse groups are implicitly favoured by the system, because each individual agent possesses its own distinct behaviour: in order to obtain a subset of k agents that perform equivalent behaviours, the system must evolve the same behaviour k times in the same s-expression. Second, the system scales with the number of agents performing the task: for n agents, the s-expression must contain n branches.

2 The Model

We now introduce an augmented genetic programming system, called the Legion system, which shares the advantages of the system described in [14], but overcomes its limitations.

2.1 The Legion System

Each individual s-expression in the Legion population encodes behaviours for an entire agent group, and is composed of two or more branch s-expressions. The first branch s-expression is the *partition s-expression*, and dictates how an agent group is to be partitioned into a set of behavioural classes. The partition s-expression is evaluated in depth-first order, in order to determine how many behavioural classes the agent group will contain, and how many agents will be assigned to each behavioural class. When a `SPLIT` operator is encountered, kf agents are assigned to the next available behavioural class, where k is the number of agents not yet assigned a behavioural class, and f is the floating point value $(0 \leq f \leq 1)$ returned by the `SPLIT` operator's left branch. The remaining $k(1-f)$ agents are further partitioned when the next `SPLIT` operator is encountered. When the final `SPLIT` operator is encountered, the remaining agents are placed into the two next behaviour classes. Any remaining behavioural s-expressions are deleted. If the final `SPLIT` operator is encountered and there remains only one more behaviour s-expression, this last behaviour s-expression is duplicated, and the remaining agents are divided into the two identical behaviour s-expressions. In subsequent generations, mutation and crossover events may differentiate these two branch s-expressions.

It follows from this that, as opposed to the model in [14], the Legion system can dynamically change the number of behavioural classes in an agent group over evolutionary time, as well as modifying the behaviours of members of each class[2]. Moreover, by modifying the number of SPLIT operators in partition s-expressions, selection pressure can increase or decrease the number of behavioural classes—and thus the heterogeneity—of agent groups over evolutionary time.

The remaining branch s-expressions in a Legion s-expression, referred to as *behaviour s-expressions*, are domain-dependent and encode the actions performed by agents assigned to that behavioural class. Fig. 1 presents the architecture of the Legion population in pictorial form.

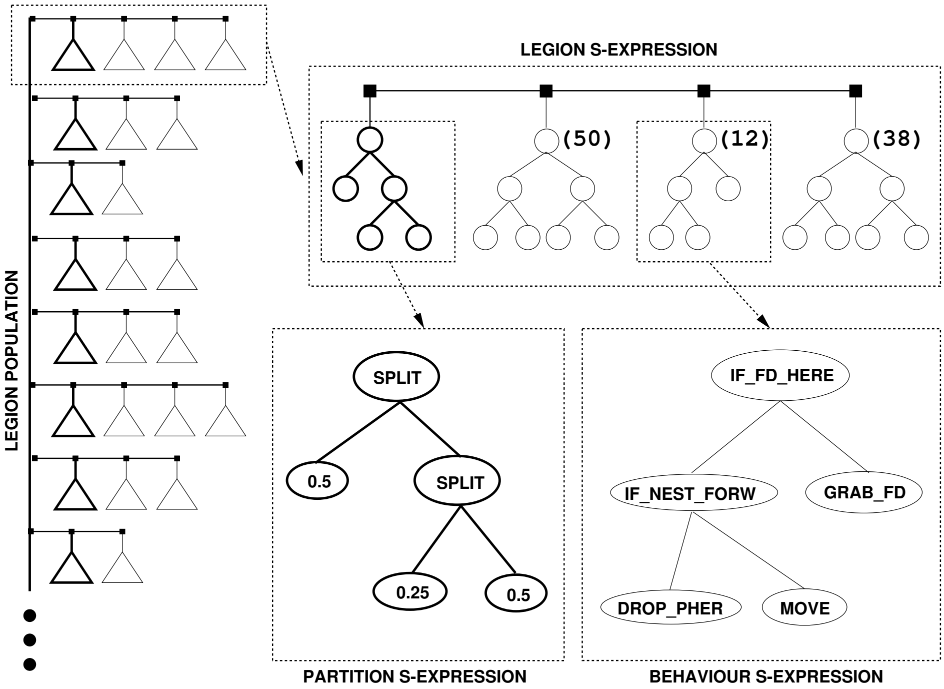

Fig. 1. A pictorial representation of the Legion system The bracketed numbers next to the three behaviour s-expressions denote the percentage of agents from a group that would be assigned to that behavioural class. The percentages, and the number of behavioural classes, are determined by the partition s-expression.

Crossover in the Legion system is accomplished by *restricted breeding*, similar to [14]: given two Legion s-expressions s_1 and s_2 with partition and behaviour s-expressions $\{p_1, b_{1,1}, b_{2,1} \ldots b_{i,1}\}$ and $\{p_2, b_{1,2}, b_{2,2} \ldots b_{j,2}\}$, the partition s-expressions of the two children are created by sub-tree crossover of p_1

[2] This process was modelled on the biological concept of gene families, which are produced by gene duplication and differentiation over evolutionary timescales [19], [18].

and p_2, and the behaviour s-expressions are created by the pairwise crossings of $\{(b_{1,1}, b_{1,2}), (b_{2,1}, b_{2,2}), \ldots (b_{i,1}, b_{i,2})\}$, where $i \leq j$.

If it is accepted that the amount of heterogeneity in an agent group is dependent on the number, membership and differentiation of the behavioural classes in a Legion s-expression, and the constitution of behavioural classes in a Legion system is under evolutionary control, then it follows that the amount of heterogeneity in the agent groups evolved by the Legion system is subject to the selection pressure of the task domain.

2.2 The Heterogeneity Measure

In [3] a measure of heterogeneity, social entropy, is presented and defined as

$$H = -\sum_{i=1}^{M} p_i \mathrm{log}_2(p_i), \tag{1}$$

where M is the number of behavioural classes in an agent group, and p_i is the probability that any given agent is a member of the behavioural class i. Social entropy thus takes into account the number and membership sizes of the behavioural classes in a group, but does not consider the differences between agents in different classes. A more complicated measure of social entropy is given in [5] which takes into account inter-class behavioural differences. However, this measure is domain-specific, and relies on details of the capabilities of agents within the group, such as perceptual or internal state.

When evolving behaviours for agent groups, a fitness function is usually formulated which calculates some quantitative measure of the facility of the group to accomplish its assigned task. The fitness function is dependent on the behaviours of the agents within the group; differences in fitness between any two given agent groups imply behavioural differences between those groups. Thus, in an evolutionary context, a measure of heterogeneity can be formulated based solely on the fitness values of agents within the group, and not directly on the behaviours of the agents themselves.

Consider a group of n agents which has been partitioned by the Legion system (or some other evolutionary algorithm) into a set of behavioural classes $B = \{b_1, b_2, \ldots, b_c\}$. Let f be the fitness value of this agent group. Let $P = \{p_1, p_2, \ldots, p_{2^c-1}\} - \emptyset$ be the power set of B. We can then iteratively assign agents in the group to the behavioural classes of p_i, and compute the fitness $f(p_i)$ of the group. Each behavioural class in p_i is assigned $\frac{n}{|p_i|}$ agents. We can now define the heterogeneity measure as

$$H = 1 - \frac{\sum_{p_i \in P}(\sum_{j=1}^{|p_i|} |a_j|) f(p_i)}{(\sum_{p_i \in P} \sum_{j=1}^{|p_i|} |a_j|) f}. \tag{2}$$

From Eqn. 2 it follows that if the groups assigned to all the subsets of P achieve the same fitness value as that attained by the original, heterogeneous group, the heterogeneity value is zero. This indicates that agents in the different classes, as determined by the original partition, do not exhibit distinct behaviours. If the groups all behave differently than the original heterogeneous group, then the heterogeneity measure will differ from zero. This indicates that members in the different classes perform distinct behaviours. Moreover, if the fitness values $f(p_i)$ are lower[3] than the fitness value for the original partition f, then H will approach unity. This is formalized as

$$H = \begin{cases} 0 & : \quad \text{if} \quad \forall p \in P, f(p) = f \\ > 0 & : \quad \text{if} \quad \exists p' \in P, f(p') < f, \quad \text{and} \quad \forall \bar{p} \in P - p', f(\bar{p}) = f \\ 1 & : \quad \text{if} \quad \forall p \in P, f(p) = 0 \end{cases} \tag{3}$$

The advantage of this domain-independent, fitness-based heterogeneity measure is that it explicitly incorporates the concept of division of labour. When all of the agents in the group are forced to perform only a subset of the behaviours evolved for them (chosen from among the behavioural class combinations in P), and then perform poorly (indicated by a lowered fitness value for the chosen combination), this indicates that a range of behaviours have evolved for this group, all of which must be performed in order to successfully solve the collective task.

3 Results

The first task domain studied is synthetic, and was designed in order to test the Legion system on a task domain in which both homogeneous and heterogeneous groups can optimally solve the given task. This task is named the Travelling Mailman Problem, or the TMP.

Consider a city with s streets that produce $\{l_1, l_2, \ldots, l_s\}$ letters each day, which must be collected by a fleet of mailmen. Each mailman can collect one letter each day. The goal of the mailmen is to arrange themselves across the streets in the city so as to minimize the amount of uncollected mail. At the beginning of each simulation, each mailman indicates the street number which will be his mail route for the duration of the simulation. The total amount of uncollected mail at the end of the simulation is given by

$$\sum_{i=1}^{n} \sum_{j=1}^{s} \begin{cases} u_j - m_j & : \quad u_j > m_j \\ 0 & : \quad u_j \leq m_j \end{cases}, \tag{4}$$

where s is the number of streets, n is the number of iterations in the simulation, u_j is the amount of uncollected mail at street j, and m_j is the number of mailmen servicing street j.

In Table 1, the information necessary for applying the Legion system to the TMP is given.

[3] We here assume that a high fitness value is desirable; for tasks in which low fitness values are desirable, H is computed by flipping the numerator and denominator given in Eqn. 2.

Fitness Function	Equivalent to Eqn. 4		
Termination Criteria	500 generations are completed		
Non-terminal Nodes	Name	Arity	Description

	IF_ST_CAP	2	j = evaluated left branch
			k = evaluated right branch
			if $u_j > m_j$, move to street j
			else move forward k streets

	PLUS	2	left branch + right branch
Terminal Nodes	The two integer constants zero and unity		
Population Size	500		
Number of Generations	250		
Selection Method	Tournament selection; tournament size = 2		
Maximum Tree Depth	7		
Maximum Behavioural Classes	3		
Mutation Rate	1% chance of node undergoing random replacement		

Table 1. Legion System Parameters for the Travelling Mailman Problem The fitness function is a decreasing function; lower fitness values imply a more fit solution.

The second task studied was food foraging in simulated ant colonies [2,6,4]. Twenty ants operating within a 32 by 32 toroidal grid must locate food placed at two food sources, and return as much food as possible to a single nest. Ants may lay and sense pheromones, which can be used by the ant group to increase the rate of food retrieval. At each time step of the simulation, each ant performs one action, based on the state of its local environment.

The fitness function used to evaluate the performance of an ant colony is given by

$$f + r + \sum_{i=1}^{n} t_i. \tag{5}$$

In the fitness function, f stands for functionality. Given an ant colony (a_1, a_2, ..., a_n), f is set to 0 if no ant attempts any behaviour; 1 if at least one ant attempts one of the three behaviours *grab food*, *drop pheromone* or *move*; 2 if at least two ants a_i and a_j attempt one of these three behaviours, and the behaviours of a_i and a_j are distinct; and 3 if at least three ants a_i, a_j and a_k attempt one of the three behaviours, and the behaviours of a_i, a_j and a_k are distinct. The functionality term f is used to motivate initial Legion groups to evolve ant colonies with high functionality.[4]

Ants removing food from the food piles are rewarded by r, the number of food pellets removed by the colony from the food piles. The final term of the fitness

[4] In [6], a similar fitness function to that of Eqn. 5 was employed, but the functionality term f was not used. Because of this, evolved behaviours reported in [6] were produced with a population size of 64000 over 80 generations. These solutions were roughly as fit as the evolved solutions reported in this work, which were generated using a population size of 500 over 250 generations.

function rewards colonies for returning food to the nest as quickly as possible: n is the number of food pellets returned to the nest, and t_i is the number of time steps remaining in the simulation when food pellet i was returned to the nest.

In Table 2, the information necessary for applying the Legion system to the food foraging problem is given.

Fitness Function	See Eqn. 5	
Termination Criteria	250 generations completed, or all food returned to nest	
Non-terminal Nodes	IF_FD_HERE	The ant is standing on a food pellet
	IF_FD_FORW	There is food in front of the ant
	IF_CARRYING_FD	The ant is carrying a food pellet
	IF_NEST_HERE	The ant is standing on the nest
	IF_FACING_NEST	The ant is facing the nest
	IF_SMELL_FOOD	There is a food pellet next to the ant
	IF_SMELL_PHER	There is pheromone next to the ant
	IF_PHER_FORW	There is pheromone in front of the ant
Terminal Nodes	MOVE_FORW	Move one cell forward in current direction
	TURN_RT	Turn 90 degrees clockwise
	TURN_LT	Turn 90 degrees counterclockwise
	MOVE_RAND	Move two cells in a random direction
	GRAB_FD	Pick up a food pellet, if one is here
	DROP_PHER	Drop pheromone at current position
	NO_ACT	Do not perform any action
	MOVE_DROP	Move one cell forward; drop pheromone
Population Size	500	
Number of Generations	250	
Selection Method	Tournament selection; tournament size = 2	
Max Tree Depth	7	
Max Behavioural Classes	3	
Mutation Rate	1% chance of node undergoing random replacement	

Table 2. Legion System Parameters for the Food Foraging Problem The fitness function is an increasing function; higher fitness values indicate a more fit solution.

The thin lines in Fig. 2 report data generated by 30 runs of the Legion system applied to the TMP. Figs. 2 a) and b) plot the heterogeneity (as given in Eqn. 2) and the number of behavioural classes, respectively, of the fittest mailman group at the end of each generation from a typical run of the Legion system. Figs. 2 c) and d) depict the average heterogeneity and number of behavioural classes, respectively, in the Legion population as a whole, averaged over the 30 runs. The parameters for the runs are given in Table 1.

The thick lines in Fig. 2 report data generated by 30 runs of the Legion system applied to the food foraging problem. Figs. 2 a) and b) plot the heterogeneity (as given in Eqn. 2) and the number of behavioural classes of the most fit ant group at the end of each generation in a single run of the Legion system. Figs.

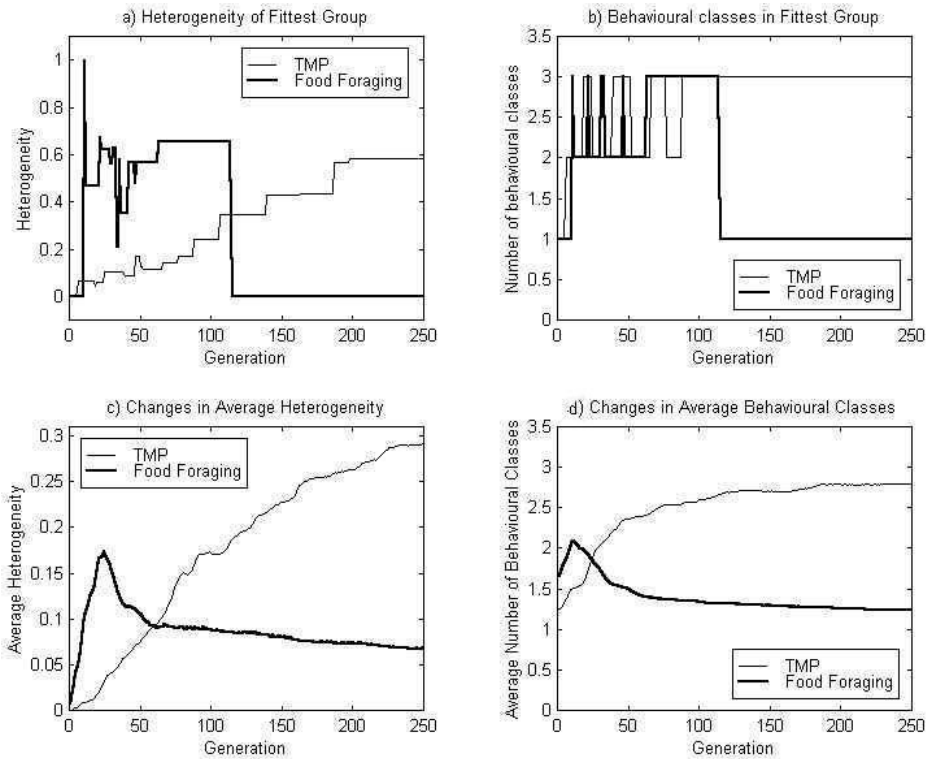

Fig. 2. Changes in heterogeneity for a set of runs of the travelling mailman and the food foraging problems: a) and b) show changes in the heterogeneity and the number of behavioural classes, respectively, of the fittest agent group at each generation during a typical run; c) shows changes in the average heterogeneity of the Legion population; d) shows changes in the average number of behavioural classes for the population. The results in c) and d) are averaged over 30 runs.

2 c) and d) depict the average heterogeneity and number of behavioural classes, respectively, in the Legion population as a whole, averaged over the 30 runs. The parameters for the runs are given in Table 2.

4 Discussion

For the mailman groups evolved for the TMP, Fig. 2 c) shows that the heterogeneity of the groups increases over evolutionary time. Fig. 2 d) shows that mailman groups rapidly approach the asymptote of the maximum possible number of behavioural classes. By comparing the slopes of Figs. 2 c) and d) it becomes clear that even after the Legion population is saturated with agent groups with the maximum number of behavioural classes, new agent groups continue to exhibit increased heterogeneity.

This result is further supported by the data from the sample TMP run shown in Figs. 2 a) and b). In this run, after generation 100, the most fit mailman group always contains three behavioural classes (see Fig. 2 b)). However, subsequent agent groups continue to increase in heterogeneity until the 200th generation (see Fig. 2 a)).

In contrast to these results, the data in Figs. 2 c) and d) show that for the food foraging problem, simulated ant colonies exhibit less heterogeneity over evolutionary time. Our investigations suggest that the initial, rapid increase and subsequent gradual decrease in heterogeneity seen in Figs. 2 c) and d) is due to the generation of a fit behaviour within a single behavioural class of a heterogeneous, ancestral colony. This fit behaviour is then assimilated by a larger fraction of ants in descendant colonies, until eventually all ants in a descendent colony use this behaviour, rendering these descendent colonies completely homogeneous. This hypothesis was supported by studying the lineages of several ant colonies during evolution (data not shown). Note also that the height of the peaks in Figs. 2 c) and d) fall short of the values obtained by corresponding mailman groups in Figs. 2 c) and d).

The tendency of foraging groups to converge on homogeneous solutions, as shown in Fig. 2, supports the findings in [3], in which a set of simulated robots foraging for different coloured pucks converge, via a learning algorithm, on identical members.

In both sets of runs, the maximum number of behavioural classes for any agent group was restricted to three. This was done to minimize computation time: computation of Eqn. 2 increases exponentially with the number of behavioural classes. However, this upper limit was sufficient to demonstrate the convergence to heterogeneous and homogeneous agent groups in the TMP and the food foraging tasks, respectively, and also that group heterogeneity can change even when the number of behavioural classes remains fixed.

These two sets of experiments demonstrate that heterogeneity is neither implicitly nor explicitly affected by the Legion system alone; rather, the amount of heterogeneity is domain-specific. From this it follows that the Legion system serves as a kind of heterogeneity 'divining rod': agent groups that perform better with either differentiated or undifferentiated members are revealed as such by the Legion system.

The Legion system can also be used to artificially exert selection pressure in favour of either homogeneous or heterogeneous groups. For groups that tend to converge on heterogeneous solutions, clamping the maximum number of behavioural classes to one ensures the evolution of only homogeneous groups (this follows from the definition of H in Eqn. 2).

Conversely, by incorporating the heterogeneity measure into fitness functions for problem domains in which agent groups tend to become more homogeneous over time, groups with both a high fitness and high heterogeneity can be generated. This technique was applied to the food foraging problem: the same procedure was used as that summarized in Table 2, but the fitness function used was $h(f + r + \sum_{i=1}^{n} t_i)$, where h is defined in Eqn. 2, and f, r and t_i are explained

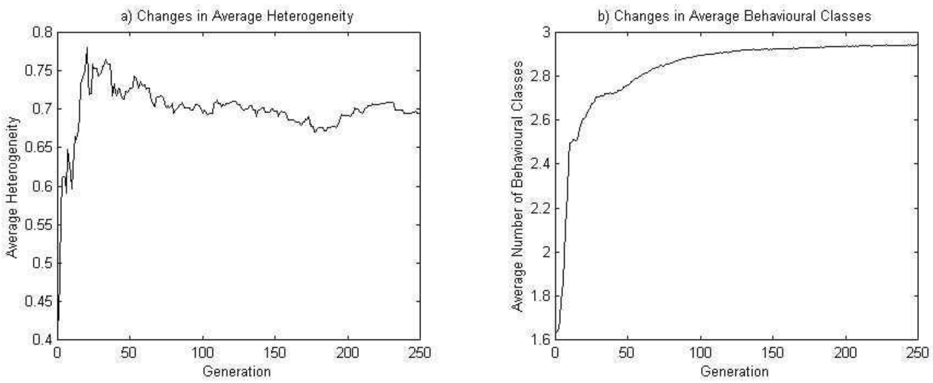

Fig. 3. Artificially evolving heterogeneity for the food foraging task: The Legion system was run for the food foraging task, using the parameters given in Table 2. The fitness function used was $h(f + r + \sum_{i=1}^{n} t_i)$, where h is defined in Eqn. 2. a) reports changes in the average heterogeneity of simulated colonies in the Legion population after each generation. b) reports changes in the number of behavioural classes.

in section 3. Fig. 3 reports data generated by a run of the food foraging task using this fitness function. Note the differences between Figs. 3 a) and 2 c), and between Figs. 3 b) and 2 d).

5 Conclusions

The results documented here support the claim that heterogeneity is a domain-specific property. Using an evolutionary algorithm applied to two task domains, selection pressure consistently evolved heterogeneous agent groups for the one task, and homogeneous groups for the other. Bloat [13] and random diffusion have been cited as two possible alternative explanations for the repeated appearance of multiple behavioural classes in the TMP, but these hypotheses are refuted by the repeated convergence to a single behavioural class in the food foraging problem (see Figs. 2 c) and d)).

The importance of heterogeneity (or the lack thereof) in agent groups is manifold. In the case of physical agents, homogeneous groups may suffer reduced robustness: a group of wheeled robots designed for smooth terrain will fail entirely in a rocky terrain; a mixed group of wheeled and legged robots may perform in both types of terrain. Conversely, morphological and behavioural redundancy may be addressed using the Legion system: for example, by automatically tuning the amount of heterogeneity in a robot group (similar to the technique used for generating the data reported in Fig. 3), one may be able to optimally tune the amount of sensor and effector overlap displayed among members of the group.

In addition to robustness and redundancy, division of labour is another concept intimately linked to heterogeneity. In some initial investigations, we have found that for agent groups with similar fitness values, heterogeneous groups tend to contain less s-expression nodes than homogeneous groups. This may suggest that agents within heterogeneous groups specialize to a specific set of sub-tasks within the main task, and thus exhibit reduced functionality in the form of smaller control architectures. It follows from this that the Legion system may be used to generate not only heterogeneous, but also specialized agent groups. We are currently pursuing this promising avenue of study.

Finally, it follows from the relationship between our heterogeneity measure and division of labour that collective tasks for which heterogeneous agent groups evolve may be *decomposable* tasks. Some tasks may be composed of a number of different subtasks; behavioural classes may then emerge and differentiate in agent groups to solve these subtasks. This was observed in the case of the TMP: behavioural classes emerged, each containing mailmen that serviced a subset of the streets in the city. Conversely, the homogeneity of evolved ant colonies may support the hypothesis that the simulated food foraging task is non-decomposable: all ants must be able to perform all basic behaviours to successfully achieve the collective task. The use of the Legion system for measuring the decomposability of collective tasks may be another interesting topic of future investigation.

In closing, we conclude that the Legion system, in conjunction with the domain-independent heterogeneity measure introduced here, is a powerful tool ideally suited for investigations of heterogeneity in agent-based systems and collective problem solving.

6 Acknowledgments

The author would kindly like to thank faculty and students of the School of Cognitive and Computing Sciences at the University of Sussex, without whom this work would not have been possible. Special thanks to Inman Harvey, for generous contributions of time, thought, criticism and encouragement.

References

1. Arkin, R. C. & J. D. Hobbs. Dimensions of Communication and Social Organization in Multi-agent Robotic Systems. In Meyer, J.-A., H. L. Roitblat & S. W. Wilson (eds.), *Procs. of the Second Intl. Conf. on Simulation of Adaptive Behavior*. MIT Press, pp. 486–493. (1992)
2. Arkin, R. C. & K. S. Ali. Integration of Reactive and Telerobotic Control in Multi-agent Robotic Systems. In Cliff, D., P. Husbands, J.-A. Meyer & S. W. Wilson (eds.), *Procs. of the Third Intl. Conf. on Simulation of Adaptive Behavior*. MIT Press, pp. 473–478. (1994)
3. Balch, T. Behavioral Diversity in Learning Robot Teams. PhD thesis, College of Computing, Georgia Institute of Technology. (1998)
4. T. Balch. Reward and Diversity in Multirobot Foraging. In *IJCAI-99 Workshop on Agents Learning About, From and With other Agents*. Sweden, July 31–August 6. (1999)
5. Balch, T. Hierarchic Social Entropy: An Information Theoretic Measure of Robot Group Diversity. *Autonomous Robots*, **8**:3, July, to appear. (2000)

6. Bennett, F. H. Automatic Creation of an Efficient Multi-Agent Architecture Using Genetic Programming with Architecture-Altering Operations. In Koza, J. R., D. E. Goldberg & D. B. Fogel (eds.), *Genetic Programming 1996 : Proceedings of the First Annual Conference.* MIT Press, pp. 30–38. (1996)
7. Bonabeau, E., A. Sobkowski, G. Theraulaz & J.-L. Deneubourg. Adaptive Task Allocation Inspired by a Model of Division of Labour in Social Insects. *Sante Fe Institute Tech. Rep. 98-01-004.* (1998)
8. Bull, L. & C. Fogarty. Evolutionary Computing in Multi-Agent Environments: Speciation and Symbiogenesis. In Voigt, H.-M., W. Ebeling & I. Rechenberg (eds.), *Parallel Problem Solving from Nature-PPSN IV.* Springer-Verlag, pp. 12–21. (1996)
9. Fahlman, S. & C. Lebiere. The Cascade-Correlation Learning Architecture. *Carnegie Mellon University Tech. Rep. CMU-CS-90-100.* (1990)
10. Fontan, M. S. & M. J. Mataric. A Study of Territoriality: The Role of Critical Mass in Adaptive Task Division. In Maes, P., M. Mataric, J.-A. Meyer, J. Pollack & S. W. Wilson (eds.), *Procs. of the Fourth Intl. Conf. on Simulation of Adaptive Behavior.* MIT Press, pp. 553–561. (1996)
11. Goldberg, D. & M. J. Mataric. Interference as a Tool for Designing and Evaluating Multi-Robot Controllers. In *AAAI-97: Procs. of the Fourteenth Natl. Conf. on Artificial Intelligence.* MIT Press, pp. 637–642. (1997)
12. Haynes, T. & S. Sen. Crossover Operators for Evolving a Team. In Koza, J. R., K. Deb, M. Dorigo, D. B. Fogel, M. Gazon, H. Iba & R. L. Riolo (eds.), *Genetic Programming 1997: Proceedings of the Second Annual Conference.* pp. 162–167, Morgan Kauffman. (1997)
13. Langdon, W. B. & R. Poli. "Fitness Causes Bloat". *Second On-Line World Conference on Soft Computing in Engineering Design and Manufacturing.* Springer-Verlag, London, pp. 13–22. (1997)
14. Luke, S. & L. Spector. Evolving Teamwork and Coordination with Genetic Programming. In Koza, J. R., D. E. Goldberg, D. B. Fogel & R. L. Riolo (eds.), *Genetic Programming 1996: Proceedings of the First Annual Conference.* MIT Press, pp. 141–149. (1996)
15. M. J. Mataric. Reinforcement Learning in the Multi-Robot Domain. In *Autonomous Robots*, 4(1):73–83. (1997)
16. M. J. Mataric. Designing and Understanding Adaptive Group Behavior. *Adaptive Behavior* 4(1):51–80. (1995)
17. McFarland, D. J. Animals as Cost-Based Robots. In Boden, M. (ed.), *The Philosophy of Artificial Life.* Oxford University Press, Oxford. (1996)
18. Ohno, S. *Evolution by Gene Duplication.* Springer-Verlag, New York. (1970)
19. Ohta, T. Multigene and Supergene Families. *Oxford Surv. Evol. Biol.*, 5:41–65. (1988)
20. Parker, L. Heterogeneous Multi-Robot Cooperation. PhD thesis, Massachussets Institute of Technology. (1994)
21. Potter, M. & K. De Jong. Evolving neural networks with collaborative species. In *Procs. of the 1995 Summer Computer Simulation Conference* Ottawa. (1995)
22. Sims, K. Evolving 3D Morphology and Behaviour by Competition. In Brooks, R. and P. Maes (eds.), *Artificial Life VI.* MIT Press, pp. 28–39. (1994)
23. Sneath, P. & R. Sokal. *Numerical Taxonomy* W. H. Freeman and Company, San Francisco. (1973)
24. Theraulaz, G., S. Goss, J. Gervet & J.-L. Deneubourg. Task Differentiation in Polistes Wasp Colonies: a Model for Self-organizing Groups of Robots. In Meyer, J. A. & S. W. Wilson (eds.), *Procs. of the First Intl. Conf. on the Simulation of Adaptive Behaviour. MIT Press, pp. 346–355. (1991)*