Reducing Collective Behavioural Complexity through Heterogeneity

Josh C. Bongard[1]

[1]Artificial Intelligence Laboratory, University of Zurich, Switzerland
bongard@ifi.unizh.ch

Abstract

In this paper, the correlation between behavioural heterogeneity and behavioural complexity within groups of cooperating agents is investigated. This investigation is accomplished using the Legion system, a type of evolutionary algorithm for evolving group behaviours, and in which behavioural differences among agents in the group is subject to selection pressure. Two collective task domains are studied, and two types of control architecture for the agents are investigated. From the experiments reported here it is concluded that increased behavioural heterogeneity within a group leads to reduced control complexity, and also that limiting the size of control architectures within a group results in increased behavioural complexity within that group. It is argued that this correlation clarifies the relationship between robustness, division of labour and variation within cooperating agent populations, and also that heterogeneity can be a powerful design tool for robot group design.

Introduction

In the field of collective robotics and related simulations, much of the literature deals with groups of homogeneous robots or agents with identical morphologies and sensor/motor mappings. Exceptions include work by Mataric *et al*, who have implemented groups of robots in which heterogeneity is realized through spatial differentiation within the task space in order to minimize physical interference (Fontan & Mataric 1996; Goldberg & Mataric 1997), or by implementing a dominance hierarchy, in which inferior robots can only perform a subset of the basis behaviours available to more dominant robots (Mataric 1995). In (Parker 1994), morphological heterogeneity is studied, in which physical robots have non-overlapping sets of sensors and effectors; in (Balch 1998), physical and simulated robots with distinct motor schemata are referred to as behaviourally heterogeneous populations. Arkin and Hobbs (1992) explicitly delineate the advantages of heterogeneity within robot groups, such as redundancy and divison of labour. Similarly, the focus of this paper is to shed light on the correlation between behavioural heterogeneity and divison of labour by investigating how the average behavioural complexity of agents in the group decreases as heterogeneity increases.

In order to study this relationship, an evolutionary algorithm for evolving group behaviours, called the Legion system, is employed (Bongard 2000). Evolutionary approaches to heterogeneity and collective problem solving include the work by Bull and Fogarty (Bull 1996), who present an island-model genetic algorithm that encodes classifier systems used to control a quadruped robot; in (Potter 1995), cascade neural networks (Fahlman 1990) are evolved for parity computation using an incremental genetic algorithm. In both investigations, however, the behavioural niches of the population are predetermined. The Legion system improves on a genetic programming model for evolving heterogeneity (Luke & Spector 1996). In the Legion system, changes in heterogeneity within a group of agents is dependent on the selective pressure of the given task domain, and evolutionary increases in heterogeneity occur through the biologically-inspired dynamic of division followed by differentiation.

Also, by using an evolutionary approach to intragroup heterogeneity, it is possible to formulate a measure of heterogeneity which is independent of the specific behaviours of the given task domain. This stands in contrast to another, domain-specific measure of heterogeneity (Balch 2000).

Armed with a system for evolving groups of agents to perform some collective task, a measure of heterogeneity and a clear definition of collective behavioural complexity, it is possible to guage correlations between behavioural heterogeneity and specialization. In the next section, a more detailed description of the Legion system is given, along with definitions of the heterogeneity measure used herein and collective behavioural complexity.

In the results section, the Legion system is applied to two task domains. Within both domains, agent pop-

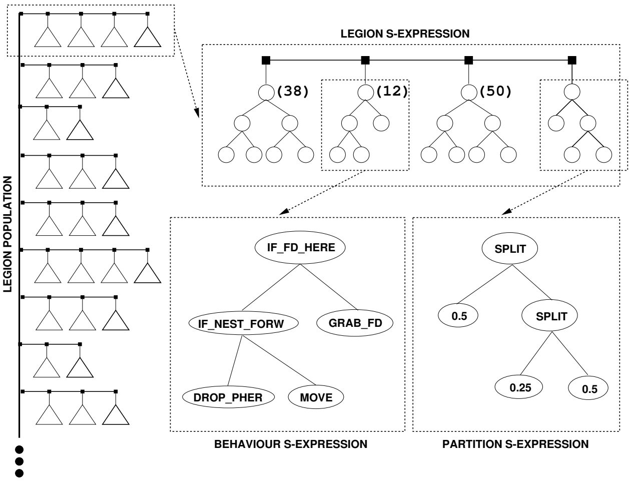

Figure 1: **The genetic programming representation of the Legion system** The bracketed numbers next to the three behaviour s-expressions denote the percentage of agents from a population that would be assigned to that behavioural class. The percentages, and the number of behavioural classes, are determined by the partition s-expression.

ulations are evolved with two different architectures for generating the individual agents' behaviours: a genetic programming architecture, and a neural network architecture. In the discussion session, the correlation between behavioural heterogeneity and complexity in these four classes of simulations is analyzed. We conclude with the implications of this work for robotics research, and avenues for future research.

Methodology

The Legion System

The Legion system builds on the genetic programming system introduced in (Luke & Spector 1996), which in turn extends the concept of automatically defined functions (Koza 1994). The Legion system evolves s-expressions which dictate a suite of behaviours to be used by agents cooperating on a collective task: the s-expression partitions the agent population into separate behavioural classes, and then assigns indentical behaviours to agents within each class. A graphic representation of the Legion system is given in Fig. 1.

Depending on the task domain, agent groups will begin to exhibit increased behavioural heterogeneity over evolutionary time if the task domain favours heterogeneous groups: distinct behavioural classes will form which progressively differentiate over subsequent generations. If the task domain does not require group heterogeneity, behavioural classes will fail to form (Bongard 2000).

Crossover in the Legion system is accomplished by *restricted breeding* (Luke & Spector 1996): given two Legion s-expressions s_1 and s_2 with partition and behaviour s-expressions $\{p_1, b_{1,1}, b_{2,1} \dots b_{i,1}\}$ and $\{p_2, b_{1,2}, b_{2,2} \dots b_{j,2}\}$, the partition s-expressions of the two children are created by sub-tree crossover of p_1 and p_2, and the behaviour s-expressions are created by the pairwise crossings of $\{(b_{1,1}, b_{1,2}), (b_{2,1}, b_{2,2}), \dots (b_{i,1}, b_{i,2})\}$, where $i \leq j$.

From the mechanics of the Legion system, it becomes clear that selection pressure can alter the number of behavioural classes and the number of agents assigned to them (via alterations to the partition s-expression), and can also alter the amount of differentiation between behavioural classes (via alteration of the behavioural s-expressions through restricted breeding). This process occurs by duplication and subsequent differentiation of behaviour s-expressions during a run of the Legion system. This process was modelled on the evolutionary concept of gene families, which are produced by gene duplication and differentiation (Ohta 1988), (Ohno 1970). If it is accepted that the amount of heterogeneity in an agent population 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 can be inferred that the amount of heterogeneity in the agent populations evolved by the Legion system is subject to the selection pressure of the task domain.

Although the Legion system uses s-expressions to evolve the group behaviours, other types of control architectures may be evolved for individual agents. Fig. 2 shows a model for evolving weights for agents controlled by neural networks.

The Travelling Mailman Problem

In this paper the Legion system is applied to two collective tasks. The first task is synthetic, and is referred to as the Travelling Mailman Problem, or the TMP. The TMP was designed in order to test the Legion system on a task domain in which both homogeneous and heterogeneous populations can optimally solve the given task.

Consider a city with s streets that produce $\{l_1, l_2, \dots, l_s\}$ 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

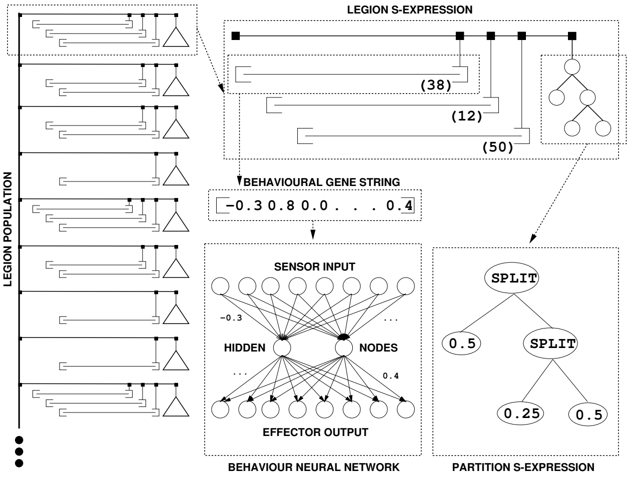

Figure 2: **A hybrid representation of the Legion system** In this model, individual agents generate behaviour using a neural network. All agents within a group have the same neural net architecture, but connection weights may differ between the behavioural classes within a group. The paritioning of the agent group is determined by the partition s-expression; the connection weights for each behavioural class are encoded in a real-valued genetic algorithm.

at the end of the simulation is given by

$$\sum_{i=1}^{n}\sum_{j=1}^{s}\left\{\begin{array}{rcl} u_j - m_j & : & u_j > m_j \\ 0 & : & u_j \leq m_j \end{array}\right. , \qquad (1)$$

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.

In the hybrid Legion system, the action of each mailman is controlled by the neural network specified by the behavioural class to which the mailman belongs.

The architecture of the neural network associated with each mailman is fixed: the number of input nodes equal to s; the number of hidden nodes is specified at run time; and the number of output nodes is set to $\lceil(\log_2(s))$, where s is the number of streets in the city. Now consider a mailman m assigned to a behavioural class b. The first s floating-point values of the behaviour gene string b are used to label the s input nodes: if one of the values is negative, the corresponding input node is disabled; if the value is positive, it is multiplied by 10, and thus indicates a specific street in the city. The remaining $s + 2\lceil(\log_2(s))$ floating-point values are used to label the connections in the network.

The values of the binary input nodes are determined as follows: each input node i with label l_i is checked to see whether i is active. If i is active, i is set to 1 if $u_{l_i} > m_{l_i}$, where u is the amount of uncollected mail on street l_i, and m is the number of mailmen currently servicing street l_i.

Fitness Function	Equivalent to equation 1
Termination Criteria	500 generations completed
Non-terminal Nodes	Description

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

`PLUS` (Arity = 2)	left branch + right branch
Terminal Nodes	Constants `0`, `1`
Population Size	500
Generations	250
Selection Method	Tournament selection Tournament size = 2
Max Tree Depth	7
Mutation Rate	1%

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 values of the hidden and output nodes are computed using the standard sigmoid function $\frac{1}{1+e^{-x}}$, where x is the summed input to the node. The output nodes are converted to binary values. Finally, the binary array of output nodes is treated as a binary value, and converted to a decimal value. This value is treated as the street number to which mailman m is moved.

From the above description it follows that the number of different streets on which a mailman within some behavioural class i bases its action is equivalent to the number of unique positive labellings of the input nodes of the neural network associated with i. This value is denoted by s_i. Thus for some mailman group with c behavioural classes, the average number of streets influencing the action of each mailman in the population is given by

$$es = \frac{mc}{\sum_{i=1}^{c} m_i s_i}, \qquad (2)$$

where m_i is the number of mailmen assigned to behavioural class i, and m is the total number of mailmen. The value es is henceforth referred to as the *environmental specialization* of j: as es increases, the mailmen assigned to each behavioural class rely, on average, on a smaller fraction of the total environmental information available to them. In other words, as es

increases, the average amount of specialization within a given population j increases.

Food Foraging

The second task studied was food foraging in simulated ant colonies (Arkin & Ali 1994; Bennett 1996). Twenty ants must forage within a 32 by 32 toroidal grid for 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 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

$$f + r + \sum_{i=1}^{n} t_i. \tag{3}$$

In the fitness function, f stands for functionality. Given an ant colony (a_1, a_2, $\ldots$, 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 populations to evolve ant colonies with high functionality.

In later generations, 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 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. Table 2 provides the information necessary for applying the Legion system to the food foraging problem.

In applying the hybrid Legion system to the food foraging problem, the actions of each ant are determined by the fully connected feedforward neural network associated with its behavioural class. The neural network of each ant is constructed of eight input and eight output nodes, and a number of hidden nodes specified at run time. Specifically, each input node is associated with one of the conditional non-terminal nodes listed in table 2.

The values of the eight floating-point output nodes are then calculated as for the TMP as explained above, but are not rounded to binary values. The eight output nodes represent the eight actions that are available to the ants, and are listed in table 2. The output node with the maximum value is found, and the action corresponding to this node is performed by the ant.

Fitness Function	See equation 3
Termination Criteria	250 generations elapsed, or all food returned to nest
Non-terminal Nodes	Description
IF_FD_HERE	Ant standing on food pellet
IF_FD_FORW	Food in front of the ant
IF_CARRYING_FD	Ant is carrying food pellet
IF_NEST_HERE	Ant is standing on the nest
IF_FACING_NEST	The ant is facing the nest
IF_SMELL_FOOD	Food pellet next to ant
IF_SMELL_PHER	Pheromone next to ant
IF_PHER_FORW	Pheromone in front of ant
Terminal Nodes	Description
MOVE_FORW	Move one cell forward
TURN_RT	Turn 90° clockwise
TURN_LT	Turn 90° counterclockwise
MOVE_RAND	Choose random direction; Move forward two cells
GRAB_FD	Pick up food pellet if here
DROP_PHER	Drop pheromone
NO_ACT	Do not perform any action
MOVE_DROP	Move forward one cell; drop pheromone
Population Size	400
Generations	250
Selection Method	Tournament selection tournament size = 2
Max Tree Depth	7
Mutation Rate	1%

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 Heterogeneity Measure

Evolution within the Legion system proceeds based on a fitness function, which indicates the relative performance of an agent population on a given task. As pointed in (Balch 1998), the heterogeneity of an agent group is a function of both the number of behavioural classes within the group, and the differences across the behavioural classes. Consider then a population of n agents 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 of this agent population. 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 population to the behavioural classes of p_i, and compute the fitness $f(p_i)$ of the population. Each behavioural class in p_i is assigned $\frac{n}{|p_i|}$ agents. We can now define the heterogeneity measure as

$$H = 1 - \frac{\sum_{p_i \in P}(\sum_{j=1}^{|p_i|} |a_j|)f(p_i)}{(\sum_{p_i \in P}\sum_{j=1}^{|p_i|} |a_j|)f} \tag{4}$$

It follows from this that the heterogeneity measure H for a completely homogeneous agent group—one which contains a single behavioural class—is zero. If agents from a group containing several behavioural classes are constrained to a subset of those classes, and perform the overall task poorly, H will approach unity. In this way, H indicates not only the heterogeneity of an agent group (the number of, and differences between behavioural classes), but also the division of labour within the group: agents within each behavioural class perform some partial task which contributes positively to the overall task. This is formalized as

$$H = \begin{cases} 0.0 & : \quad \text{if} \quad \forall p \in P, f(p) = f \\ > 0.0 & : \quad \text{if} \quad \exists p' \in P, f(p') < f, \\ & : \quad \text{and} \quad \forall \bar{p} \in P - p', f(\bar{p}) = f \\ 1.0 & : \quad \text{if} \quad \forall p \in P, f(p) = 0 \end{cases} \tag{5}$$

Behavioural Complexity

The concept of *collective behavioural complexity* refers to the average amount of computation performed by an individual agent in determining which action to perform, based on its sensory input. In Fig. 1, agents are controlled by s-expressions, the nodes of which are composed of sensor state information, control structure and a series of action primitives. The total number of s-expression nodes encoded in an agent population is $\sum_{i=1}^{n} s_i$, where n is the number of agents within the group, and s_i is the number of nodes encoded in the s-expression controlling the behaviour of agent i. If the agents in a group are partitioned into a set of behavioural classes,

$$t = \sum_{i=1}^{b} n_i s_i, \tag{6}$$

gives the average amount of control structure used to generate behaviour for an agent group, where b is the number of behavioural classes in the group, n_i is the number of agents in class i, and s_i is the total number of nodes encoded in the s-expression generating behaviour for agents in class i. Similarly,

$$v = \sum_{i=1}^{b} n_i v_i, \tag{7}$$

where v_i represents the average number of s-expression nodes evaluated by agents within behavioural class i during the length of the group simulation. Thus, v gives the average amount of control structure actually used by agents within a group: because internal nodes in the s-expression can contain conditional statements, some of the encoded nodes are not evaluated, based on the current state of the agent's environment.

Behavioural complexity in the case of neural network controlled agents is characterized as the number of nodes and connections within the neural network. I this way, the complexity of control architectures is achieved by limiting the number of hidden nodes in the neural network. This has the effect of reducing the total number of connections in the network, and forcing the network to perform dimensionality reduction on the input space (Bishop 1997). For the two problem domains here, this results in limiting the amount of sensory information that can influence an agent's action.

Results

Two sets of 30 runs of the Legion system applied to the TMP were executed. In the first set of runs, the maximum possible behavioural classes was limited to three. Since mailmen groups tend to evolve heterogeneous behaviours over evolutionary time as reported in (Bongard 2000), in the second set of runs, only one behavioural class was allowed, forcing agent groups to evolve completely homogeneous solutions to the TMP. In this way, it was possible to evolve mailmen groups exhibiting a wide range of heterogeneity and fitness. The H, t and v values, given by Eqns. 4, 6 and 7 respectively, were recorded for the most fit agent group after each generation. Figs. 3 and 4 show the correlations between H and t, and H and v, for mailmen groups with similar fitness.

In applying the hybrid Legion system to the TMP, three sets of 30 runs were performed in which individual mailmen contained neural networks with two, four and six hidden nodes, respectively. The most fit mailman group was extracted from the end of each generation of the three sets of 30 runs, and the environmental specialization of the group was computed using Eqn. 2. Fig. 5 plots differences in environmental specialization for these groups: the groups were partitioned into network type and fitness category.

Similarly, three sets of 30 runs of the Legion system applied to the food foraging problem were executed. Since ant groups tend to evolve homogeneous behaviours over evolutionary time as reported in (Bongard 2000), in order to evolve ant groups with both high fitness values and large H values, the second set

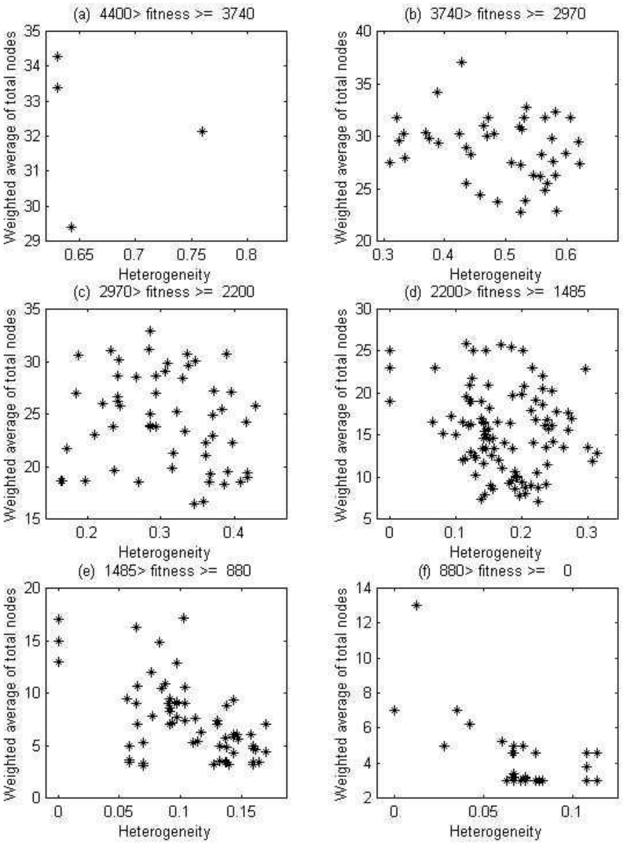

Figure 3: Correlations between heterogeneity and the weighted average of s-expression nodes encoded in the most fit mailmen group after each generation. Agent groups were partitioned into classes of similar fitness.

of 30 runs used the fitness function Hf, where f is defined in Eqn. 3. Figs. 6 and 7 show the correlations between H and t, and H and v, for ant groups with similar fitness.

In applying the hybrid Legion system to the food foraging problem, three sets of 30 runs were executed: each set of runs contained ants controlled by a neural network with two, four and six nodes, respectively. The H value of the most fit ant group after each generation was recorded. For each set of runs the H values of the ant groups were averaged over the 30 runs, and are reported in Fig. 8.

Discussion

The agent groups reported in Figs. 3, 4, 6 and 7 were partitioned into similar fitness categories in order to minimize the positive correlation between solution size and fitness witnessed in genetic programming simulations (Langdon & Poli 1997). It is hypothesized that this positive correlation tends to reduce any negative correlation detected between group heterogeneity and

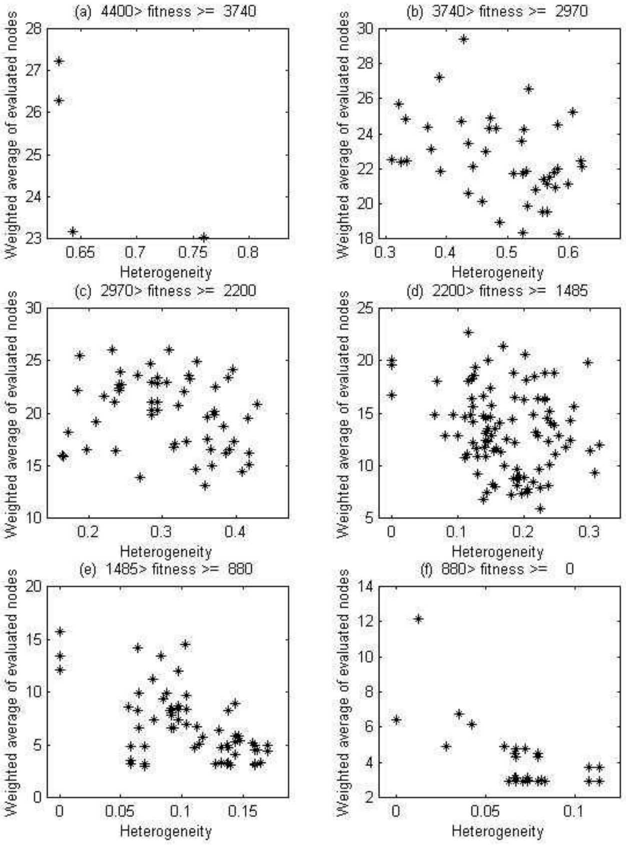

Figure 4: Correlations between heterogeneity and the weighted average of s-expression nodes evaluated by the most fit mailmen group after each generation. Agent groups were partitioned into classes of similar fitness.

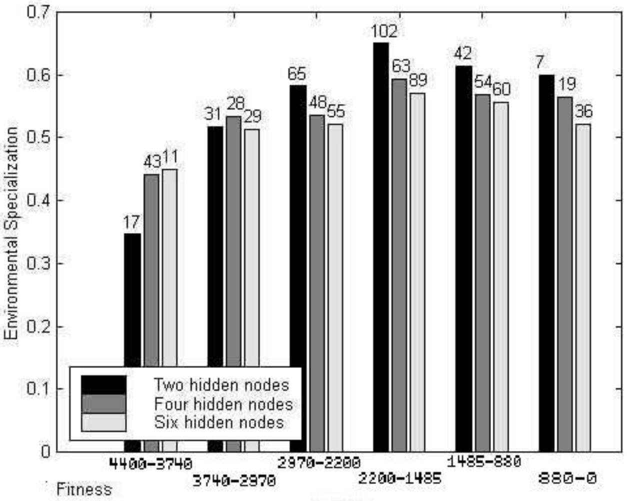

Figure 5: Differences in environmental specialization for mailman groups evolved using the hybrid Legion system.

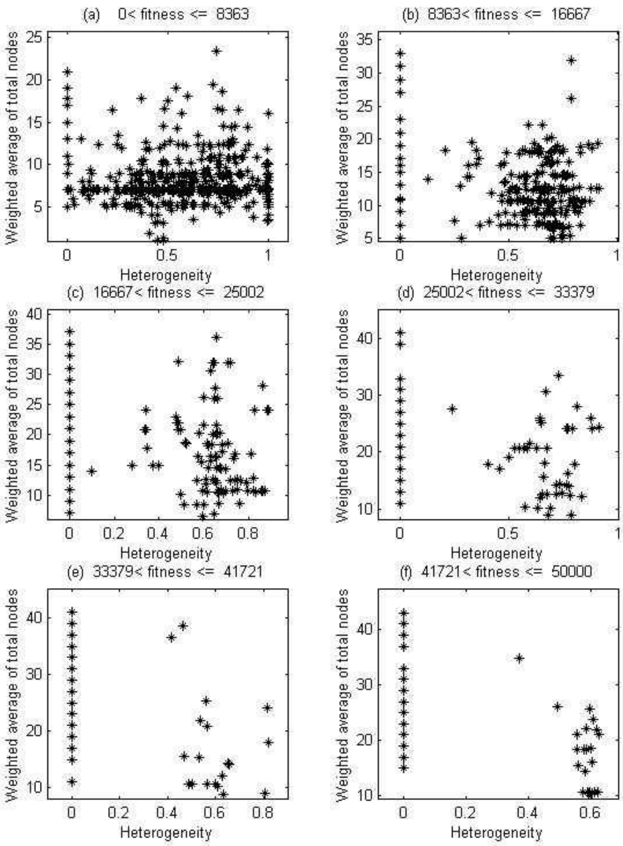

Figure 6: Correlations between heterogeneity and the weighted average of s-expression nodes encoded in the most fit ant group after each generation. Ant groups were partitioned into classes of similar fitness.

the size of the s-expressions controlling agents within that group. One the other hand, by parititioning the groups into finer fitness categories, the average number of groups falling within any one categories drops, and tends to reduces the statistical significance of any anticorrelation detected within a category. It was found that six fitness categories sufficed for both the TMP and the food foraging problems: a sufficient number of agent groups filled each categories, and negative correlation was detected in several categories, despite the mitigating positive correlation between solution size and fitness.

As the heterogeneity, fitness and behavioural complexity distributions of agent groups from both task domains did not derive from any underlying parametric distribution, the non-parametric Spearman rank correlation test (Noether 1991) was employed to test for correlation within the fitness categories.

The results of the tests for correlation for the TMP are given in Tables 3 and 4. Correlation significance is given as the two-sided significance level of the rank

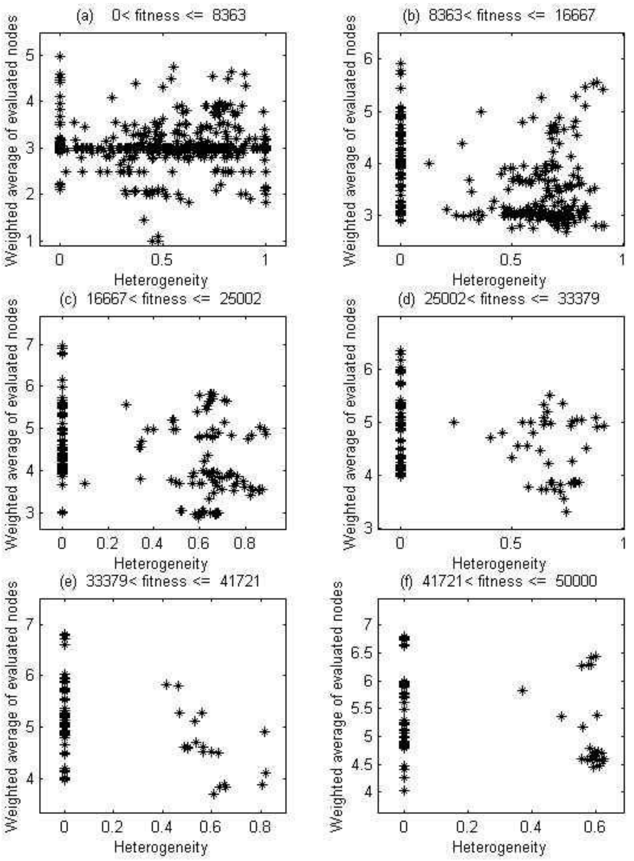

Figure 7: Correlations between heterogeneity and the weighted average of s-expression nodes evaluated by the most fit ant group after each generation. Ant groups were partitioned into classes of similar fitness.

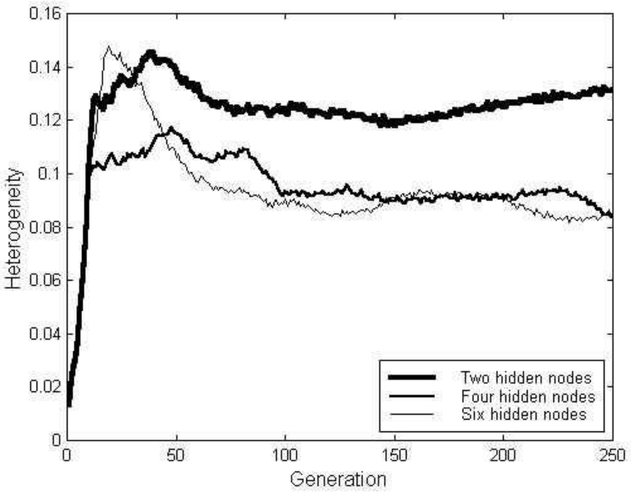

Figure 8: Differences in heterogeneity for ant groups evolved using the hybrid Legion system.

correlation's deviation from the null hypothesis (Press *et al 1992*): the null hypothesis is that there is no correlation between group heterogeneity, measured by

Fitness Range	Data Points	Rank Correlation	Correlation Significance
[4400,3740)	4	-0.737	0.262135
[3740,2970)	43	-0.210	0.175451
[2970,2200)	57	-0.182	0.174957
[2200,1485)	98	-0.160	0.114399
[1485,880)	64	-0.504	0.000021
[880,0]	30	-0.528	0.002707

Table 3: Correlation between heterogeneity and average number of s-expression nodes encoded in mailman groups for the TMP. Refer to Fig. 3.

Fitness Range	Data Points	Rank Correlation	Correlation Significance
[4400,3740)	4	-0.948	0.051316
[3740,2970)	43	-0.388	0.010065
[2970,2200)	57	-0.228	0.087731
[2200,1485)	98	-0.200	0.048173
[1485,880)	64	-0.504	0.000021
[880,0]	30	-0.561	0.001257

Table 4: Correlation between heterogeneity and average number of s-expression nodes evaluated in mailman groups for the TMP. Refer to Fig. 4.

H as defined in Eqn. 4, and that group's behavioural complexity, measured by either t or v as in Eqns. 6 and 7, respectively.

Table 3 indicates that there is only a negative correlation between heterogeneity and the average number of s-expression nodes encoded in the mailman groups for the two last fitness categories: the less fit mailman groups with fitness values between 4400 and 1485 did not show a signficant drop in control architecture size in response to increased heterogeneity.

In the case of the average s-expression nodes evaluated by the group, as shown in Table 4, there is a negative correlation between H and v for all six fitness categories. Again though, the two final categories containing the most fit mailman groups have much better statistical significance that the previous four categories.

Tables 5 and 6 report the correlations between H and t, and H and v for the simulated ant groups shown in Figs. 6 and 7, respectively. In the case of the food foraging problem, significant negative correlation is found for ant groups in the last four fitness categories. In the case of group heterogeneity versus the average number of encoded s-expression nodes in the group shown in Table 5 and Fig. 6, the negative correlation grows for increasingly fit agent groups. For H versus v shown in Table 6 and Fig. 7, the negative correlation stays relatively constant for the last four fitness categories. This implies that the proportion of *unused* control code in the ant groups, $t-v$, decreases with increasingly heterogeneous groups: this suggests that heterogeneity may be useful for counteracting the phenomenon of bloat encountered in many genetic programming models (Langdon & Poli 1997).

Fitness Range	Data Points	Rank Correlation	Correlation Significance
[0,8363)	572	0.015	0.706355
[8363,16667)	376	-0.103	0.045612
[16667,25002)	227	-0.261	0.000064
[25002,33379)	125	-0.348	0.000067
[33379,41721)	93	-0.464	0.000002
[41721,50000]	78	-0.525	$8.1 \text{ x } 10^{-11}$

Table 5: Correlation between heterogeneity and average number of s-expression nodes encoded in ant groups for the food foraging problem. Refer to Fig. 6.

Fitness Range	Data Points	Rank Correlation	Correlation Significance
[0,8363)	572	0.016	0.690497
[8363,16667)	376	-0.159	0.001939
[16667,25002)	227	-0.261	0.000066
[25002,33379)	125	-0.253	0.004399
[33379,41721)	93	-0.314	0.002110
[41721,50000]	78	-0.318	0.004441

Table 6: Correlation between heterogeneity and average number of s-expression nodes evaluated in ant groups for the food foraging problem. Refer to Fig. 7.

These results suggest that for increasingly fit agent groups, increased heterogeneity signals an increase in the division of labour within these groups: on average, relatively less control structure is used by individual agents within heterogeneous groups to perform as well on a collective task as a corresponding homogeneous group.

Fig. 8 supports this argument from another approach: by constraining the control structure of individual agents, heterogeneity within agent groups tends to increase. In this paper, constrained control structure is accomplished by reducing the number of hidden nodes in a feed forward neural network. In Fig. 8 it is shown that groups containing agents controlled by neural networks with four and six hidden nodes tend to evolve more homogeneous behaviours, whereas groups with agents controlled by neural networks with only two hidden nodes tend to evolve more heterogeneous behaviours. It is hypothesized that this dynamic occurs because agents which are forced to rely on a reduced amount of sensory information—the agents controlled by the smaller neural networks—tend to specialize to subtasks which require only a subset of the sensory information available to them. In this way, behavioural classes emerge, composed of agents which use non-overlapping subsets of information in the neural network input space, and thus the heterogeneity measure of the group is non-zero.

The application of the hybrid Legion system to the TMP further supports this claim: the environmental specialization measure given in Eqn. 2 explicitly guages the amount of environmental information accessed by any one agent. Fig. 5 shows that for groups in the last four fitness categories, agent groups with more constrained control architectures tend to access the state of less streets than mailmen from groups with correspondingly more control structure. This relationship demonstrates that constrained control architectures explicitly increase specialization, as least for the TMP.

Conclusions

In this report, the relationship between the size and the amount of heterogeneity in evolved control architectures for agent populations has been studied. A biologically-inspired type of evolutionary algorithm, the Legion system, was used for this purpose.

Specifically, it was determined that for behaviours evolved for agent groups, heterogeneous groups tend to contain less control structure than similarly fit homogeneous groups. This correlation was found in the two collective task domains reported here, and for agents with two types of control structure. The hybrid Legion system experiments show that the correlation between solution size and heterogeneity is not simply an artefact of the genetic programming paradigm, but is rather a result of a deeper relationship between reduced control structure, heterogeneity and specialization. The hybrid Legion system was also used to show that there is the heterogeneity measure used not only indicates the degree of variation within an agent group, but also the amount of specialization that has occurred among different behavioural classes within that group.

The implications of this work suggest that heterogeneity in evolved control structure for robotics research is a useful avenue of study. If we assume that behaviour is dependent on the mapping between sensory information and effector commands in a robot, then using the Legion system to evolve behaviours for groups of robots may lead to a reduction in the average number of sensor/effector pairings required by each robot. It may also lead to a reduction in the intricacy of the transformations from sensory information to effector commands. This could be helpful in the domain of evolved circuit design of robot control architectures, the workings of which are often difficult to analyze (Thompson 1999).

Also, because heterogeneity is dependent on selective pressure in the Legion system, this paradigm may prove to be a useful tool in theoretical evolutionary biology, such as for investigating the possible origins of eusociality in insects (Drogoul 1995; Theraulaz *et al* 1991; Bonabeau 1998).

The concept of robustness is closely allied with that of heterogeneity: it has been pointed out (1992), (Wilson 1990) that the benefits of diversification must be balanced against the reduction of redundancy in heterogeneous agent systems. A promising future area of study would be to introduce individual agent failure into the Legion system, and investigate how this affects heterogeneity in the evolved group behaviours.

It is hoped that the work presented herein serves as sufficient motivation for study into how heterogeneity may be used not only to address some of the technical and engineering challenges faced in the evolutionary design of robot control architectures, but also more general, theoretical implications of evolved heterogeneity for collective problem solving.

Acknowledgements

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

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 the Simulation of Adaptive Behaviour.* MIT Press, Cambridge, MA, pp. 486–493. (1992)

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 the Simulation of Adaptive Behaviour.* MIT Press, Cambridge, MA, pp. (1994)

Balch, T. Behavioral Diversity in Learning Robot Teams. PhD thesis, College of Computing, Georgia Institute of Technology. (1998)

Balch, T. Hierarchic Social Entropy: An Information Theoretic Measure of Robot Group Diversity. *Autonomous Robots*, **8**:3, July, to appear.

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.), *Procs. of the First Annual Conference on Genetic Programming.* MIT Press, Cambridge, MA, pp. 30–38. (1996)

Bishop, C. M. *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, pp. 316–17. (1997)

Bonabeau, E., A. Sobkowski, G. Theraulaz & J.-L. Deneubourg. Adaptive Task Allocation Inspired by a Model of Division of Labour in Social Insections. *Sante Fe Institute Tech. Rep. 98-01-004.* (1998)

Bongard, J. C. The Legion System: A Novel Approach to Evolving Heterogneity for Collective Problem Solving. To appear in *Procs. of the Third European Conf. on Genetic Programming.* Edinburgh, UK, April 14-15. (2000)

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 IV.* Springer-Verlag, Berlin, pp. 12–21. (1996)

MANTA: New Experimental Results on the Emergence of (Artificial) Ant Societies. In Gilbert, N. & R. Conte (eds.), *Artificial Societies: the Computer Simulation of Social Life.* UCL Press, London, UK. (1995)

Fahlman, S. & C. Lebiere. The Cascade-Correlation Learning Architecture. Carnegie Mellon University technical report CMU-CS-90-100. (1990)

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.), *From Animals to Animats 4: Fourth International Conference on Simulation of Adaptive Behavior.* MIT Press, Cambridge, MA. (1996)

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, Cambridge, MA. (1997)

Koza, J. R. *Genetic Programming.* MIT Press, Cambridge, MA. (1992)

Koza, J. R. *Genetic Programming II: Automatic Discovery of Reusable Programs.* MIT Press, Cambridge, MA. (1994)

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)

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.), *Procs. of the First Annual Conference on Genetic Programming.* MIT Press, Cambridge, MA. (1996)

M. J. Mataric. Designing and Understanding Adaptive Group Behavior. *Adaptive Behavior* 4(1):51–80. (1995)

Noether, G. E. *Introduction to Statistics: The Nonparametric Way.* Springer-Verlag, New York, NY, pp. 236–37. (1991)

Ohno, S. *Evolution by Gene Duplication.* Springer-Verlag, New York, NY. (1970)

Ohta, T. Multigene and Supergene Families. *Oxford Surv. Evol. Biol.*, vol. 5, pp. 41–65. (1988)

Parker, L. Heterogeneous Multi-Robot Cooperation. PhD thesis, Massachussets Institute of Technology. (1994)

Potter, M. & K. De Jong. Evolving neural networks with collaborative species. In *Proceedings of the 1995 Summer Computer Simulation Conference* Ottawa, Canada. (1995)

Press, W. H., S. A. Teukolsky, W. T. Vetterling & B. P. Flannery. *Numberical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, Cambridge, UK, pp. 640–42. (1992)

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.), *From Animcals to Animats: Procs. of the First Intl. Conf. on the Simulation of Adaptive Behaviour.* MIT Press, Cambridge, MA, pp. 346–355. (1991)

Thompson, A. Analysis of Unconventional Evolved Electronics. *Communications of the ACM*, **42**:4, pp. 71–79. (1999)

Wilson, E. O. *The Ants.* Belknap Press, Cambridge, MA. (1990)