Reducing Collective Behavioural Complexity through Heterogeneity

Josh C. Bongard

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 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 used. From the experiments reported here it is concluded that increased behavioural heterogeneity within a group leads to reduced average control complexity, and also that reducing the maximum size of control architectures results in the evolution of increased behavioural heterogeneity. It is argued that this correlation helps to clarify the relationship between robustness, division of labour and variation within cooperating agent populations, and also that heterogeneity can be a useful tool for robot group design.

Introduction
In the literature on cooperative robotics and agent-based systems, most studies investigate groups of homogeneous robots or agents with identical morphologies and sensor to motor mappings (Dorigo 1999; Reynolds 1987). At most, agents have access to the same behavioural repertoire, but exhibit differing propensities to perform certain tasks (Theraulaz et al 1991; Bonabeau 1998). Agents in (Sims 1994) exhibit large morphological and behavioural variation, but the implications of this variation for collective problem solving was not investigated.

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: inferior robots can only perform a subset of the basis behaviours available to more dominant robots (Mataric 1995). In (Parker 1994), morphological heterogeneity was studied: 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) delineate the advantages of heterogeneity within robot groups, such as redundancy and division of labour. Similarly, the focus of this paper is to shed light on the correlation between behavioural heterogeneity and division of labour by investigating how the average behavioural complexity of agents in a group decreases as heterogeneity increases.

In order to study this relationship, an evolutionary algorithm for evolving group behaviours, the Legion system, is employed (Bongard 2000). Evolutionary approaches to heterogeneity and collective problem solving include work by Bull (1996), who presents an island-model genetic algorithm for encoding 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, evolutionary changes in group heterogeneity are influenced by the selection pressure of the given task domain, and evolutionary increases in heterogeneity occur through the biologically-inspired dynamic of gene duplication and differentiation (Ohno 1970; Ohta 1988).

By using an evolutionary approach to intra-group heterogeneity, a measure of heterogeneity independent of the specific behaviours of the task domain is possible. This stands in contrast to another, domain-specific measure of heterogeneity developed for learning robot groups (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 measure correlations be-
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.

tween 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 and collective behavioural complexity.

In the results section, the Legion system is applied to two task domains. Within both domains, two sets of simulations are performed, in which control of individual agents is accomplished using a genetic programming and a neural network architecture, respectively. In the discussion session, the correlation between behavioural heterogeneity and complexity in these four sets 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 improves on the paradigm introduced in (Luke & Spector 1996), which in turn extends the concept of automatically defined functions in genetic programming (Koza 1994). The Legion system evolves s-expressions that encode 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 distinct behaviours to the agents of 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 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 $\{p_1, b_{11}, b_{12}, \ldots, b_{1i}\}$ and $\{p_2, b_{21}, b_{22}, \ldots, b_{2j}\}$, 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 $\{(b_{11}, b_{12}), (b_{21}, b_{22}), \ldots, (b_{1i}, b_{1j})\}$, 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 behaviour 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 multigene families, which are produced by gene duplication and differentiation (Ohta 1988; Ohno 1970). It has been argued that the amount of heterogeneity in an agent population is dependent on the number, membership and differentiation of the behavioural classes in an agent group (Balch 1998). Because the behaviours of the agents in each behavioural class in the Legion system is under evolutionary control, it follows that the amount of heterogeneity in the agent populations evolved by the Legion system is determined by the selection pressure of the task domain.

Although the Legion system uses s-expressions to evolve the group behaviours, other types of control architectures are evolved for individual agents. Fig. 2 shows the set-up used for evolving the weights of neural networks, which are then used to control agents within a group.

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 (Bongard 2000).

Consider a city with $s$ streets that produces
{\{l_1, l_2, \ldots, l_s\}} letters each day, which are collected by a group of mailmen. Each mailman can collect one letter each day. The goal of the mailman group 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

\[ s \sum_{i=1}^{n} \sum_{j=1}^{s} \left\{ u_j - m_j : u_j > m_j \right\}, \]

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 is set to \( s \), the number of streets in the city; the number of hidden nodes is specified at run time, and is denoted by \( h \); and the number of output nodes is set to \( \lfloor \log_2(s) \rfloor \). Now consider a mailman \( m \) belonging to 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 \( h(s + \lfloor \log_2(s) \rfloor) \) 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 \( i \) if \( u > m \), where \( u \) is the amount of uncollected mail on street \( l_i \), and \( m \) is the number of mailmen currently servicing street \( l_i \).

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 rounded to binary values. Finally, the binary array of output nodes is treated as a binary value, and converted to a decimal value. This value determines 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 in which the action of each mailman in the group is
given by
\[ e_s = \frac{m}{\sum_{i=1}^{c} m_i s_i}, \]  
(2)

where \( m_i \) is the number of mailmen assigned to behavioural class \( i \), and \( m \) is the total number of mailmen. The value \( e_s \) is henceforth referred to as environmental specialization: as \( e_s \) increases, the mailmen assigned to each behavioural class rely, on average, on a smaller fraction of the total environmental information available to them.

### Food Foraging

The second task studied is 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. At each time step of the simulation, each ant performs one action, based on the state of its local environment, and its own state.

The fitness function used to evaluate the performance of an ant colony is given by
\[ f + r + \sum_{i=1}^{n} t_i. \]  
(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 actions grab food, drop pheromone or move; 2 if at least two ants \( a_i \) and \( a_j \) attempt one of these three actions, and the actions 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 actions, and the actions 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.\(^1\)

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 quickly: \( t_i \) 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 feed forward 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, 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 that value is performed.

---

\(^1\)In (Bennett 1996), a similar fitness function to that of equation 3 was employed, but the functionality term \( f \) was not used. Because of this, evolved behaviours reported in (Bennett 1996) 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 400 over 250 generations.

---

<table>
<thead>
<tr>
<th>Fitness Function</th>
<th>See equation 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Termination Criteria</td>
<td>250 generations elapsed, or all food returned to nest</td>
</tr>
<tr>
<td>Non-terminal Nodes</td>
<td>Description</td>
</tr>
<tr>
<td>IF_FOOD_HERE</td>
<td>Ant standing on food pellet</td>
</tr>
<tr>
<td>IF_FDFORW</td>
<td>Food in front of the ant</td>
</tr>
<tr>
<td>IF_CARRYING_FOOD</td>
<td>Ant is carrying food pellet</td>
</tr>
<tr>
<td>IF_NEST_HERE</td>
<td>Ant is standing on the nest</td>
</tr>
<tr>
<td>IF_FACING_NEST</td>
<td>The ant is facing the nest</td>
</tr>
<tr>
<td>IF_SMELL_FOOD</td>
<td>Food pellet next to ant</td>
</tr>
<tr>
<td>IF_SMELL_PHER</td>
<td>Pheromone next to ant</td>
</tr>
<tr>
<td>IF_PHER_FOWR</td>
<td>Pheromone in front of ant</td>
</tr>
<tr>
<td>Terminal Nodes</td>
<td>Description</td>
</tr>
<tr>
<td>MOVE_FORW</td>
<td>Move one cell forward</td>
</tr>
<tr>
<td>TURN_RT</td>
<td>Turn 90° clockwise</td>
</tr>
<tr>
<td>TURN_LT</td>
<td>Turn 90° counterclockwise</td>
</tr>
<tr>
<td>MOVE_RAND</td>
<td>Choose random direction; move forward two cells</td>
</tr>
<tr>
<td>GRAB_FDF</td>
<td>Pick up food pellet if here</td>
</tr>
<tr>
<td>DROP_PHER</td>
<td>Drop pheromone</td>
</tr>
<tr>
<td>NO_ACT</td>
<td>Do not perform any action</td>
</tr>
<tr>
<td>MOVE_DROP</td>
<td>Move forward one cell; drop pheromone</td>
</tr>
</tbody>
</table>

| 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.
to this node is performed by the ant.

The Heterogeneity Measure

Evolution within the Legion system proceeds based on a fitness function, which indicates the relative performance of an agent group on a given task. As pointed out 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 group 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 group. Let \( P = \{ p_1, p_2, \ldots, p_{2^c - 1} \} \) 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

\[
H = 1 - \frac{\sum_{p_i \in P} \left( \sum_{j=1}^{|p_i|} |a_j| \right) f(p_i)}{\left( \sum_{p_i \in P} \sum_{j=1}^{|p_i|} |a_j| \right) f}
\]  

(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.\(^2\) 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

\[
H = \begin{cases} 
0.0 & \text{if } \forall p \in P, f(p) = f \\
> 0.0 & \text{if } \exists p' \in P, f(p') < f, \text{ and } \forall p \in P - p', f(p) = f \\
1.0 & \text{if } \forall p \in P, f(p) = 0
\end{cases}
\]  

(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 indicate sensor information, sensor to action mappings 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,

\[
t = \frac{\sum_{i=1}^c n_i s_i}{n}
\]

(6)

gives the average amount of control structure used to generate behaviour for an agent group, where \( c \) 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,

\[
v = \frac{\sum_{i=1}^c n_i v_i}{n}
\]

(7)

gives the average amount of control structure actually used by agents within this group\(^3\), 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.

Behavioural complexity in the case of neural network controlled agents is characterized as the number of nodes and connections within the neural network. The complexity of control architectures can then be reduced by reducing the number of hidden nodes in fully-connected neural networks. This in turn 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 mailman groups tend to evolve heterogeneous behaviours over evolutionary time (Bongard 2000), in the second set of runs, only one behavioural class was allowed, forcing groups to evolve completely homogeneous solutions. In this way, it was possible to evolve mailman 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 group after each generation. Figs. 3 and 4 show the correlations between \( H \) and \( t \), and \( H \) and \( v \), for mailman groups with similar fitness values.

\(^2\)For task domains in which decreasing fitness values indicate increased fitness, the numerator and denominator in Eqn. 4 are swapped.

\(^3\)Because internal nodes in the s-expression can be conditional statements, some of the encoded nodes are not evaluated, based on the current state of the agent's local sensory input.
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. The groups were partitioned into network type and fitness category: the $e_3$ values of the groups within each partitioned were averaged. Fig. 5 plots the differences between the average $e_3$ values for these partitions.

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 (Bongard 2000), in order to evolve ant groups with both high fitness values and large $H$ values, the second set of 30 runs used the fitness function $H_f$, 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.
Figure 6: Correlations between heterogeneity and the weighted average of s-expression nodes encoded in the most fit ant group after each generation.

The $H$ value of the most fit ant group after each generation was recorded. 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 s-expression size and fitness witnessed in genetic programming simulations (Langdon & Poli 1997). It is hypothesized that this positive correlation tends to weaken any negative correlation detected between group heterogeneity and s-expression size within that group. On the other hand, by partitioning the groups into finer fitness categories, the average number of groups falling within any one category drops, and reduces the statistical significance of any negative correlation 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 category, and negative correlation was detected in several categories, despite the mitigating positive correlation be-

tween 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 cor-

Figure 7: Correlations between heterogeneity and the weighted average of s-expression nodes evaluated by the most fit ant group after each generation.

Figure 8: Differences in heterogeneity for ant groups evolved using the hybrid Legion system.
relationship 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 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 $H$ 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 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 significant 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).

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.

Figs. 5 and 8 support this argument from another approach: by constraining the neural network size of individual agents, heterogeneity within agent groups tends to increase. Fig. 8 shows that groups contain-

<table>
<thead>
<tr>
<th>Fitness Range</th>
<th>Data Points</th>
<th>Rank Correlation</th>
<th>Correlation Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4400,3740)</td>
<td>4</td>
<td>-0.737</td>
<td>0.262155</td>
</tr>
<tr>
<td>(3740,2970)</td>
<td>43</td>
<td>-0.210</td>
<td>0.175451</td>
</tr>
<tr>
<td>(2970,2200)</td>
<td>57</td>
<td>-0.182</td>
<td>0.174957</td>
</tr>
<tr>
<td>(2200,1485)</td>
<td>98</td>
<td>-0.160</td>
<td>0.114399</td>
</tr>
<tr>
<td>(1485,880)</td>
<td>64</td>
<td>-0.504</td>
<td>0.000021</td>
</tr>
<tr>
<td>[880,0]</td>
<td>30</td>
<td>-0.528</td>
<td>0.002707</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Fitness Range</th>
<th>Data Points</th>
<th>Rank Correlation</th>
<th>Correlation Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4400,3740)</td>
<td>4</td>
<td>-0.948</td>
<td>0.051316</td>
</tr>
<tr>
<td>(3740,2970)</td>
<td>43</td>
<td>-0.888</td>
<td>0.010065</td>
</tr>
<tr>
<td>(2970,2200)</td>
<td>57</td>
<td>-0.228</td>
<td>0.087731</td>
</tr>
<tr>
<td>(2200,1485)</td>
<td>98</td>
<td>-0.200</td>
<td>0.048173</td>
</tr>
<tr>
<td>(1485,880)</td>
<td>64</td>
<td>-0.504</td>
<td>0.000021</td>
</tr>
<tr>
<td>[880,0]</td>
<td>30</td>
<td>-0.561</td>
<td>0.001257</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Fitness Range</th>
<th>Data Points</th>
<th>Rank Correlation</th>
<th>Correlation Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0,8363)</td>
<td>572</td>
<td>0.015</td>
<td>0.690497</td>
</tr>
<tr>
<td>(8363,16667)</td>
<td>376</td>
<td>-0.159</td>
<td>0.001939</td>
</tr>
<tr>
<td>(16667,25002)</td>
<td>227</td>
<td>-0.261</td>
<td>0.000066</td>
</tr>
<tr>
<td>(25002,33379)</td>
<td>125</td>
<td>-0.253</td>
<td>0.004399</td>
</tr>
<tr>
<td>(33379,41721)</td>
<td>93</td>
<td>-0.314</td>
<td>0.002110</td>
</tr>
<tr>
<td>(41721,50000)</td>
<td>78</td>
<td>-0.318</td>
<td>0.004441</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Fitness Range</th>
<th>Data Points</th>
<th>Rank Correlation</th>
<th>Correlation Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0,8363)</td>
<td>572</td>
<td>0.016</td>
<td>0.690497</td>
</tr>
<tr>
<td>(8363,16667)</td>
<td>376</td>
<td>-0.159</td>
<td>0.001939</td>
</tr>
<tr>
<td>(16667,25002)</td>
<td>227</td>
<td>-0.261</td>
<td>0.000066</td>
</tr>
<tr>
<td>(25002,33379)</td>
<td>125</td>
<td>-0.253</td>
<td>0.004399</td>
</tr>
<tr>
<td>(33379,41721)</td>
<td>93</td>
<td>-0.314</td>
<td>0.002110</td>
</tr>
<tr>
<td>(41721,50000)</td>
<td>78</td>
<td>-0.318</td>
<td>0.004441</td>
</tr>
</tbody>
</table>

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.
ing 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 forced to rely on a reduced amount of sensory information tend to specialize to subtasks requiring 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 agent groups in the last four fitness categories, 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 increases specialization, at least for the TMP.

Conclusions

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

Specifically, it was determined that for evolved group behaviours, heterogeneous groups tend to contain less control structure than similarly fit homogeneous groups. This correlation was found in the two collective task domains investigated 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 also shows that the heterogeneity measure used here not only measures variation within an agent group, but also the specialization 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 a result of the mapping between sensory information and effector commands in a robot's mind, 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 for 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 depends on selection pressure in the Legion system, this paradigm may prove to be a useful tool in theoretical evolutionary biology, such as for investigating the origins of eusociality in insects (Drogoul 1995).

The concept of robustness is closely allied with that of heterogeneity: it has been pointed out (Arkin & Hobbs 1992; Wilson 1990), that the benefits of diversification must be balanced against reduced 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 biologically inspired 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


