2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Evolving a Scalable Multirobot Controller Using an Artificial Neural Tissue Paradigm

Jekanthan Thangavelautham, Alexander D. S. Smith,
Dale Boucher, Jim Richard, Gabriele M. T. D'Eleuterio

Abstract— **We present an "Artificial Neural Tissue" (ANT) architecture as a control system for autonomous multirobot tasks. This architecture combines a typical neural-network structure with a coarse-coding strategy that permits specialized areas to develop in the tissue which in turn allows such emergent capabilities as task decomposition. Only a single global fitness function and a set of allowable basis behaviors need be specified. An evolutionary (Darwinian) selection process is used to derive controllers for the task in simulation. This process results in the emergence of novel functionality through the task decomposition of mission goals. ANT-based controllers are shown to exhibit *self-organization*, employ *stigmergy* and make use of *templates* (unlabeled environmental cues). These controllers have been tested on a multirobot resource-collection task in which teams of robots with no explicit supervision can successfully avoid obstacles, explore terrain, locate resource material and collect it in a designated area by using a light beacon for reference and interpreting unlabeled perimeter markings. The issues of scalability and antagonism are addressed.**

I. INTRODUCTION

One must marvel at the collective behavior of a colony of ants excavating a network of tunnels or a swarm of termites building towering cathedral mounds with internal heating and cooling shafts [1]. Such behavior in the natural world can both awe and inspire the roboticist [2] when considering the control of multiagent systems.

Developing a control strategy for an assembly of robots presented with even a moderately intricate task is a daunting problem. Model-based control approaches have their limitations in the face of such complexity. In this work, we address the problem and seek an approach that draws heavily from nature. We particularly focus on two aspects: artificial neural structures and evolutionary algorithms to derive them. The approach presented here involves a machine-learning paradigm we call the "Artificial Neural Tissue" framework [3] and we demonstrate its capability in a task of multirobot resource collection and berm (mound/wall) formation. Machine-learning systems of the type examined here can perform task decomposition through "emergent" (self-organized) behavior.

The artificial neural tissue (ANT) superimposes on a typical feedforward neural-network structure a coarse-coding mechanism inspired by the work of Albus [4]. This coarse coding allows areas of specialization to develop in the tissue which in turn facilitate task decomposition. The mechanism has a biological analogy as neurons can communicate not only electrically by exchanging signals along axons but also through chemical diffusion which can selectively activate neurons. The "genome" of the tissue is evolved in an artificial Darwinian fashion.

With minimal task-specific assumptions and limited supervision, an ANT controller can exploit *templates* (unlabeled environmental cues), *stigmergy* (indirect communication mediated through the environment), and *self-organization*. Because little preprogrammed knowledge is given, ANT may discover novel solutions that might otherwise be overlooked by a human supervisor.

ANT is particularly advantageous for multirobot tasks in which some global behavior must be achieved without a centralized controller. Even though each individual has no "blueprint" of the overall task and must operate with only local sensor data, an ensemble of robots is able to complete a mission requiring a global consensus. The desired global behavior emerges from the local interactions of individual robots. Designing controllers of this sort by hand can be very difficult because the process by which local behaviors work to form a global consensus can be difficult to understand and even counterintuitive. Previous work in the field such as [5], [6], [7], [8] rely on task-specific human knowledge to develop simple "if-then" rules or equivalent coordination behaviors to solve multirobot tasks. In contrast, the approach outlined here provides for a generic framework that allows, by evolutionary computation, decomposition of the task to emerge naturally. It has already been demonstrated that ANT can produce emergent controller solutions for a multirobot tiling pattern formation task, a single-robot phototaxis task and an unlabeled sign-following task [3]. In this paper, we look at the resource-collection task, which is motivated by plans to collect and process raw material on the lunar surface. Furthermore, we address the issue of *scalability*; that is, how does a controller evolved on a single robot or a small group of robots but intended for use in a larger collective of agents scale? We also investigate the associated problem of *antagonism*.

In the following section, we present background to the problem at hand. Section III details the artificial neural tissue model. This is followed by a description of the simulation experiments conducted for the resource-collection task. The results are discussed in Section V. Finally, we offer some preliminary conclusions in Section VI.

J. Thangavelautham, A. Smith, and G. M. T. D'Eleuterio are with the Institute for Aerospace Studies, University of Toronto, 4925 Dufferin St., Toronto, ON M3H 5T6, Canada

D. Boucher is with the Northern Centre for Advanced Technology, 1400 Barrydowne Rd., Sudbury, ON P3A 3V8, Canada

J. Richard is with Electric Vehicle Controllers Ltd, 2200 Valley View Rd., Val Caron, ON P3N 1L1, Canada

1-4244-0602-1/07/$20.00 ©2007 IEEE.

II. BACKGROUND

Collective robotic tasks can benefit from some of the same mechanisms that are used by social insects. These include the use of templates, stigmergy, and self-organization. *Templates* are environmental features perceptible to the individuals within the collective [9]. *Stigmergy* is a form of indirect communication mediated through the environment [10]. *Self-organization* describes how local or microscopic behaviors give rise to a macroscopic structure in systems which are not in equilibrium [11]. These are positive traits of natural systems which would be advantageous to implement in robotic systems. However, many existing approaches suffer from another emergent feature called *antagonism* [12]. This describes the effect that arises when multiple agents trying to perform the same task interfere with one another and reduce the overall efficiency of the group. Because our approach is evolutionary in nature, it is able to “learn” how to take advantage of the techniques identified above. As we will show, the systems also learns how to mitigate the effects of antagonism, which is something that is difficult to do in hand-crafted systems.

In insect colonies, templates may be a natural phenomenon or they may be created by the colonies themselves. They may include temperature, humidity, chemical, or light gradients. In robotic applications, template-based approaches include the use of light fields to direct the creation of circular [13] and linear walls [14] and planar annulus structures [7]. Spatiotemporally varying templates (*e.g.*, adjusting or moving a light gradient over time) allow the creation of more complex structures [15].

Stigmergy describes the use of changes in the environment as a means of indirect communication between agents. In the natural world, one way in which ants and termites do this is through the use of pheromone trails. Stigmergy has been used extensively in collective-robotic construction tasks, including blind bull dozing [6], box pushing [5], heap formation [16] and tiling pattern formation [17].

Most of the works cited above rely on either user-defined deterministic “if-then” rules or on stochastic behaviors. The goal is to design a simple controller for agents that have access only to local information, but that are nevertheless able to work together to achieve an overall objective. This is difficult to do by hand, since the global effect of these local interactions is often hard to determine. Existing approaches can be categorized in one of three groups. The simplest method is to design a controller that will allow a single agent to complete the task and have it treat any other agent as an “obstacle” to be avoided. A more sophisticated solution will incorporate an extra set of rules to handle the interaction with other agents gracefully. This usually involves adding some kind of explicit communication between the agents. Even in this case, the rules handling interactions between agents are added as a second step in order to “fix” problems that would occur if they were omitted. It is rarer to find applications that fall into a third category, wherein the controllers are designed from the start with cooperation and interaction in mind.

In [16], a group of robots make use of stigmergy in performing a clustering task. In [6], robots perform the opposite task, area clearing, and [13] uses templates to direct the construction of a wall. In all three cases, a single robot is capable of performing the whole task. The controllers are “if-then” rules which treat other robots as obstacles to be avoided. The authors of [16] note that for more than three robots, the efficiency begins to decrease. In [6], although increasing the number of robots increases the efficiency, there is a reduction in the ability to control the final size and shape of the area cleared.

The wall-building task described in [14] was originally designed with a single robot in mind. Because only one robot can add a block to the end of the wall at a time, additional arbitration rules had to be added to handle the situation in which multiple robots arrive with a block to add at the same time. Here again, performance begins to decrease after a certain point as more agents are added. The end of the wall becomes a bottleneck, with multiple agents waiting for their turn to access the end of the wall. One could argue that a more efficient solution would have some robots fetching blocks and depositing them near the end of the wall while fewer robots stayed near the wall and moved those blocks into their final positions. In Section V, we describe a task in which an ANT-based controller is able to evolve a similar solution.

For the box-pushing task described in [5], a controller for a two-robot system was designed from the start with cooperation in mind. This controller is shown to exhibit superior performance to two non-cooperating robots trying to perform the same task. However, in this case, the controllers make use of explicit communication to share complete state information with one another at each step. This does not scale well to larger groups of agents.

As we can see, most existing controllers are designed first with a single agent in mind, and then are enhanced (based on human knowledge) with arbitration rules when multiple agents must interact. The result is that the interactions between agents are more often antagonistic than cooperative. It is more difficult to design controllers by hand with cooperation in mind, because it is difficult to predict or control the global behaviors that will result from local interactions. Designing successful controllers by hand can devolve into a process of trial and error, especially in the case of the first two categories described above.

A means of reducing the amount of effort required in designing controllers by hand is to encode controllers as behavioral look-up tables and allow a genetic algorithm to evolve the table entries. This approach is used to solve a heap formation task in [18] and a 2×2 tiling formation task in [17].

A limitation with look-up tables is that they have poor sensor scalability, as the size of the look-up table is exponential in the number of inputs. Look-up tables also have poor generalization. Neural network controllers perform better generalization since they effectively encode a compressed representation of the table. Neural controllers have been used to solve 3×3 tiling tasks [19] and to build walls, corridors, and briar patches [20], and have been used for multirobot communication and coordination [21]. When using fixed-topology networks, the size of the network must be specified ahead of time. Choosing the wrong size may lead to a network that is either unable to solve the problem or difficult to train [3].

The ANT framework presented here simultaneously addresses both the problems in designing rule-based systems by hand and the limitations inherent in fixed-topology evolutionary systems. Since it is a variable-length neurocontroller model, it provides good scalability and generalization of sensory input [3]. In addition, ANT does not rely on detailed task-specific knowledge. It evolves controllers to optimize a user-specified global fitness function. The evolutionary selection process is able to discover for itself how to make use of templates and stigmergy and to mitigate the effects of antagonism.

III. ARTIFICIAL NEURAL TISSUE MODEL

The ANT architecture Fig. 1(a) presented in this paper consists of a developmental program, encoded in the "genome," that constructs a three-dimensional neural tissue and associated regulatory functionality. The tissue consists of two types of neural units, decision neurons and motor-control neurons, or simply motor neurons. Regulation is performed by the decision neurons that dynamically exhibit or inhibit motor-control neurons within the tissue based on a coarse-coding framework. Let us discuss the computational mechanism of the tissue first and then outline the process by which the tissue is created.

A. Computation

We imagine the motor neurons of our network to be spheres arranged in a regular rectangular lattice in which the neuron $N_{\boldsymbol{\lambda}}$ occupies the position $\boldsymbol{\lambda} = (l, m, n) \in \mathbb{I}^3$ (sphere centered within cube). The state $s_{\boldsymbol{\lambda}}$ of the neuron is binary, i.e., $s_{\boldsymbol{\lambda}} \in S = \{0, 1\}$. Each neuron $N_{\boldsymbol{\lambda}}$ nominally receives inputs from neurons $N_{\boldsymbol{\kappa}}$ where $\boldsymbol{\kappa} \in \Uparrow(\boldsymbol{\lambda})$, the nominal input set. Here we shall assume that these nominal inputs are the 3×3 neurons centered one layer below $N_{\boldsymbol{\lambda}}$; in other terms, $\Uparrow(\boldsymbol{\lambda}) = \{(i, j, k) \,|\, i = l-1, l, l+1;\ j = m-1, m, m+1;\ k = n-1\}$. (As will be explained presently, however, we shall not assume that all the neurons are active all the time.) The activation function of each neuron is taken from among four possible threshold functions of the weighted input σ:

$$
\begin{aligned}
\psi_{\text{down}}(\sigma) &= \begin{cases} 0, & \text{if} \sigma \geq \theta_1 \\ 1, & \text{otherwise} \end{cases} \\
\psi_{\text{up}}(\sigma) &= \begin{cases} 0, & \text{if} \sigma \leq \theta_2 \\ 1, & \text{otherwise} \end{cases} \\
\psi_{\text{ditch}}(\sigma) &= \begin{cases} 0, & \min(\theta_1, \theta_2) \leq \sigma < \max(\theta_1, \theta_2) \\ 1, & \text{otherwise} \end{cases} \\
\psi_{\text{mound}}(\sigma) &= \begin{cases} 0, & \sigma \leq \min(\theta_1, \theta_2) \text{ or } \sigma > \max(\theta_1, \theta_2) \\ 1, & \text{otherwise} \end{cases}
\end{aligned}
\qquad (1)
$$

The weighted input $\sigma_{\boldsymbol{\lambda}}$ for neuron $N_{\boldsymbol{\lambda}}$ is nominally taken as

$$
\sigma_{\lambda} = \frac{\sum_{\boldsymbol{\kappa} \in \Uparrow(\boldsymbol{\lambda})} w_{\boldsymbol{\lambda}}^{\boldsymbol{\kappa}} s_{\boldsymbol{\kappa}}}{\sum_{\boldsymbol{\kappa} \in \Uparrow(\boldsymbol{\lambda})} s_{\boldsymbol{\kappa}}} \qquad (2)
$$

with the proviso that $\sigma = 0$ if the numerator and denominator are zero. Also, $w_{\boldsymbol{\lambda}}^{\boldsymbol{\kappa}} \in \mathbb{R}$ is the weight connecting $N_{\boldsymbol{\kappa}}$ to $N_{\boldsymbol{\lambda}}$. We may summarize these threshold functions in a single analytical expression as

$$
\begin{aligned}
\psi = (1 - k_1)[(1 - k_2)\psi_{\text{down}} + k_2 \psi_{\text{up}}] \\
+ k_1[(1 - k_2)\psi_{\text{ditch}} + k_2 \psi_{\text{mound}}]
\end{aligned}
\qquad (3)
$$

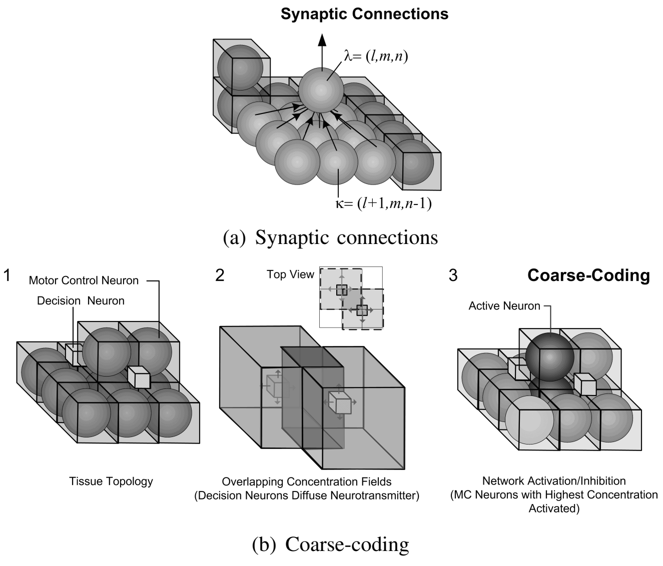
where k_1 and k_2 can take on the value 0 or 1. The activation function is thus encoded in the genome by k_1, k_2 and the threshold parameters $\theta_1, \theta_2 \in \mathbb{R}$.

It may appear that ψ_{down} and ψ_{up} are mutually redundant as one type can be obtained from the other by reversing the signs on all the weights. However, retaining both increases diversity in the evolution because a single 2-bit "gene" is required to encode the threshold function and only one mutation suffices to convert ψ_{down} into ψ_{up} or *vice versa* as opposed to changing the sign of every weight.

The sensor data are represented by the activation of the sensor input neurons $N_{\boldsymbol{\alpha}_i}, i = 1 \ldots m$, summarized as $A = \{s_{\boldsymbol{\alpha}_1}, s_{\boldsymbol{\alpha}_2} \ldots s_{\boldsymbol{\alpha}_m}\}$. Similarly, the output of the network is represented by the activation of the output neurons $N_{\boldsymbol{\omega}_j}, j = 1 \ldots n$, summarized as $\Omega = \{s_{\boldsymbol{\omega}_1^1}, s_{\boldsymbol{\omega}_2^2} \ldots s_{\boldsymbol{\omega}_n^b}\}$, where $k = 1 \ldots b$ specifies the output behavior. Each output neuron commands one behavior of the agent. (In the case of a robot, a typical behavior may be to move forward a given distance. This may involve the coordinated actin action of several actuators. Alternatively, the behavior may be more primitive such as augmenting the current of a given actuator.) If $s_{\boldsymbol{\omega}_j^k} = 1$, output neuron $\boldsymbol{\omega}_j$ votes to activate behavior k; if $s_{\boldsymbol{\omega}_j^k} = 0$, it does not. Since multiple neurons can have access to a behavior pathway, an arbitration scheme is imposed to ensure the controller is deterministic where $p(k) = \sum_{s_\diamond, j=1}^{n_k} s_{\boldsymbol{\omega}_j^k} / n_k$ and n_k is the number of output neurons connected to output behavior k resulting in behavior k being activated if $p(k) \geq 0.5$.

As implied by the set notation of Ω, the outputs are not ordered. In this embodiment, the order of activation is selected randomly. We are primarily interested here in the statistical characteristics of relatively large populations but such an approach would likely not be desirable in a practical robotic application. However this can be remedied by simply assigning a sequence *a priori* to the activations (as shown in Table II for the resource collection task).

We moreover note that the output neurons can be redundant; that is, more than one neuron can command the same behavior, in which case for a given time step one behavior may be "emphasized" by being voted multiple times. Neurons may also cancel out each other such as when one output commands a forward step while another a backward step. Finally, not all behaviors need be encoded in the neural tissue. This is left to the evolutionary process.


Synaptic Connections
λ= (l,m,n)
κ= (l+1,m,n-1)
1
Motor Control Neuron
Decision Neuron
2
Top View
3
Coarse-Coding
Active Neuron
Tissue Topology
Overlapping Concentration Fields
(Decision Neurons Diffuse Neurotransmitter)
Network Activation/Inhibition
(MC Neurons with Highest Concentration
Activated)

(a) Synaptic connections

(b) Coarse-coding

Fig. 1. Synaptic connections between motor neurons and operation of neurotransmitter field.

B. The Decision Neuron

The coarse-coding nature of the artificial neural tissue is provided by the decision neurons. Decision neurons can be thought of as rectangular structures occupying nodes in the lattice as established by the evolutionary process (Fig. 1). The effect of these neurons is to excite into operation or inhibit (disable) the motor control neurons (shown as spheres). Once a motor control neuron is excited into operation, the computation outlined in (2) is performed. Motivated as we are to seek biological support for ANT, we may look to the phenomenon of chemical communication among neurons. In addition to communicating electrically along axons, some neurons release chemicals that are read by other neurons, in essence serving as a "wireless" communication system to complement the "wired" one.

In ANT, the state of a decision neuron $T_{\boldsymbol{\mu}}$, $\boldsymbol{\mu}$ is binary and determined by one of the same activation functions 1 that also embedded within the motor control neurons. The inputs to $T_{\boldsymbol{\mu}}$ are all the input sensor neurons $N_{\boldsymbol{\alpha}}$; *i.e.*, $s_{\boldsymbol{\mu}} = \psi_{\boldsymbol{\mu}}(s_{\boldsymbol{\alpha}_1} \dots s_{\boldsymbol{\alpha}_m})$ where $\sigma_{\boldsymbol{\mu}} = \sum_{\boldsymbol{\alpha}} v_{\boldsymbol{\alpha}}^{\boldsymbol{\mu}} s_{\boldsymbol{\alpha}} / \sum_{\boldsymbol{\alpha}} s_{\boldsymbol{\alpha}}$ and $v_{\boldsymbol{\alpha}}^{\boldsymbol{\mu}}$ are the weights. The decision neuron is dormant if $s_{\boldsymbol{\mu}} = 0$ and releases a virtual neurotransmitter chemical of uniform concentration $c_{\boldsymbol{\mu}}$ over a prescribed field of influence if $s_{\boldsymbol{\mu}} = 1$. Motor control neurons within the highest chemical concentration field are excited into operation. Only those neurons that are so activated will establish the functioning network for the given set of input sensor data. Owing to the coarse-coding effect, the sums used in the weighted input of 1 are over only the set $\overline{\Uparrow}(\boldsymbol{\lambda}) \subseteq \Uparrow(\boldsymbol{\lambda})$ of active inputs to $N_{\boldsymbol{\lambda}}$. Likewise the output of ANT is in general $\overline{\Omega} \subseteq \Omega$. The decision neuron's field of influence is taken to be a rectangular box extending $\pm d_{\boldsymbol{\mu}}^{r}$, where $r = 1, 2, 3$, from $\boldsymbol{\mu}$ in the three perpendicular directions. These three dimensions along with $\boldsymbol{\mu}$ and $c_{\boldsymbol{\mu}}$, the concentration level of the virtual chemical emitted by $T_{\boldsymbol{\mu}}$, are encoded in the genome.

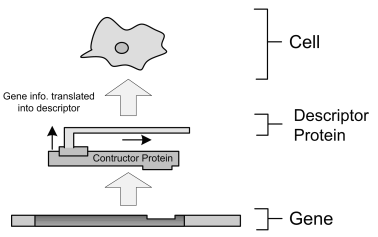


Fig. 3. Genes are "read" by constructor proteins that transcribe the information into a descriptor protein which is used to construct a cell. When a gene is repressed, the constructor protein is prevented from reading the gene contents.

C. Evolution and Development

A population of ANT controllers is evolved in an artificial Darwinian manner [22]. The "genome" for a controller contains a "gene" for each cell with a specifier D that is used to distinguish the functionality (between motor control, decision and tissue). A constructor protein (an autonomous program) interprets the information encoded in the gene and translates this into a cell descriptor protein (see Fig. 2). The gene "activation" parameter is a binary flag resident in all the cell genes and is used to either express or repress the contents of gene. When repressed, a descriptor protein of the gene content is not created. Otherwise, the constructor protein "grows" the tissue in which each cell is located relative to a specified seed-parent address. A cell death flag determines whether the cell commits suicide after being grown. Once again, this feature in the genome helps in the evolutionary process for a cell, by committing suicide, still occupies a volume in the lattice although it is dormant. In otherwise retaining its characteristics, evolution can decide to reinstate the cell by merely toggling a bit.

In turn mutation (manipulation of gene parameters with uniform random distribution) to the growth program results in new cells being formed through cell division. The rate at which mutation occurs to a growth program is also specified for each tissue and is dependent on the neuron replication probability parameter. Cell division requires a parent cell (selected with highest replication probability relative to the rest of the cells within the tissue) and involves copying $m\%$ of the original cell contents to a daughter cell (where m is determined based on uniform random distribution), with the remaining cell contents initialized with a uniform random distribution. The cell type of each new cell is determined based on the ratio of motor control to decision neurons specified in the tissue gene. The new cell can be located in one of six neighboring locations (top, bottom, north, south, east, west) sharing a common side with the parent and is not occupied by another cell.

IV. SIMULATION EXPERIMENTS

The effectiveness of the ANT controller is demonstrated in simulation on the resource-collection task. A team of robots collects resource material distributed throughout its work space and deposits it in a designated dumping area.

Motor Control Neuron Gene

Specifier	Reference Address	Position			Activation Function Parameters								Gene Activate	Cell Death	Replication Prob.	Output Behaviour	Reference Pointer
D	A	x	y	z	w_1	w_2	...	w_v	θ_1	θ_2	k_1	k_2	G	C	R	k	P
Integer [0,2]	Integer	Integers			Real [0,1]						Integers [0,1]		Binary	Binary	Real [0,1]	Integer [0,b]	Integer

Decision Neuron Gene

Specifier	Reference Address	Position			Diffusion Param.			Diffusion Concentration	Activation Function Parameters								Gene Activate
D	A	x	y	z	d_x	d_y	d_z	c	w_1	w_2	...	w_v	θ_1	θ_2	k_1	k_2	G
Integer [0,2]	Integer	Integers			Integers [1,3]			Integers [0,1]	Real [0,1]						Integers [0,1]		Binary

Tissue Gene

Specifier	Neuron Replication Prob.	Neuron Replication Ratios		Seed Address
D	T_r	n_d	n_m	T_s
Integer [0,2]	Real [0.001,0.1]	Integers [1,10]		Integer

Fig. 2. Gene map.

TABLE I
SENSOR INPUTS

Sensor Variables	Function	Description
$V_1 \dots V_4$	Resource Detection	Resource, No Resource
$C_1 \dots C_4$	Template Detection	Blue, Red, Orange, Floor
S_1, S_2	Obstacle Detection	Obstacle, No Obstacle
LP_1	Light Position	Left, Right, Center
LD_1	Light Range	0-10 (distance to light)

TABLE II
BASIS BEHAVIORS

Order	Behavior	Description
1	Dump Resource	Move one grid square back; turn left
2	Move Forward	Move one grid square forward
3	Turn Right	Turn 90° right
4	Turn Left	Turn 90° left
5, 7, 9, 11	Bit Set	Set memory bit i to 1, $i = 1 \dots 4$
6, 8, 10, 12	Bit Clear	Set memory bit i to 0, $i = 1 \dots 4$

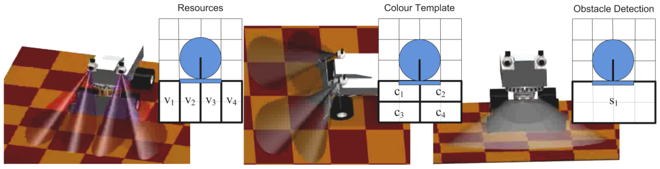


Fig. 4. Input sensor mapping, with simulation model inset.

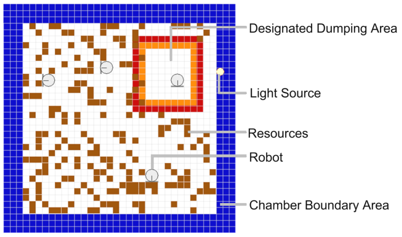


Fig. 5. 2D grid world model of experiment chamber.

The workspace is modeled as a two-dimensional grid environment with one robot occupying four grid squares. For this task, the controller must possess a number of capabilities including gathering resource material, avoiding the workspace perimeter, avoiding collisions with other robots, and forming resources into a berm at the designated location. (In the present experiment, a berm is simply a mound of the resource material.) The berm location has perimeter markings on the floor and a light beacon mounted nearby. The two colors on the border are intended to allow the controller to determine whether the robot is inside or outside the berm location. Though solutions can be found without the light beacon, its presence improves the efficiency of the solutions found, as it allows the robots to track the target location from a distance instead of randomly searching the workspace for the perimeter. The global fitness function for the task measures the amount of resource material accumulated in the designated location within a finite number of time steps, in this case $T = 300$. Darwinian selection is performed based on the fitness value of each controller averaged over 100 different initial conditions.

For this task, inputs to the ANT controller are shown in Table I. The robots are modeled on a fleet of rovers designed and built at the University of Toronto Institute for Aerospace Studies. They have access to a pair of webcams and a set of sonar sensors. All raw input data are discretized. The sonar sensors are used to determine the values of S_1 and S_2. One of the cameras is used to detect resource material and colored floor templates (see Fig. 4). The other camera is used to track the light beacon. In order to identify resources and colored floor templates, a naïve Bayes classifier is used to perform color recognition [23]. Simple feature-detection heuristics are used to determine the values of $V_1 \dots V_4$ and $C_1 \dots C_4$ based on the grid locations shown. For detection of the light beacon, the electronic shutter speed and gain are adjusted to ensure that the light source is visible while other background features are underexposed. The position of the light LP_1 is determined based on the pan angle of the camera. The distance to the light source LD_1 is estimated based on its size in the image. The robots also have access to four memory bits, which can be manipulated using some of the basis behaviors. Table II lists the basis behaviors the robot can perform. These behaviors are activated based on the output of the ANT controller, and all occur within a single time-step. The evolutionary algorithm population size for the experiments is $P = 100$, with crossover probability $p_c = 0.7$, mutation probability $p_m = 0.025$ and a tournament size of $0.06P$. The tissue is initialized as a “seed culture” , with 3×6 motor control neurons in one layer. After this, the tissue is grown to include 70–110 neurons (selected from a uniform random distribution) before starting the evolutionary process.

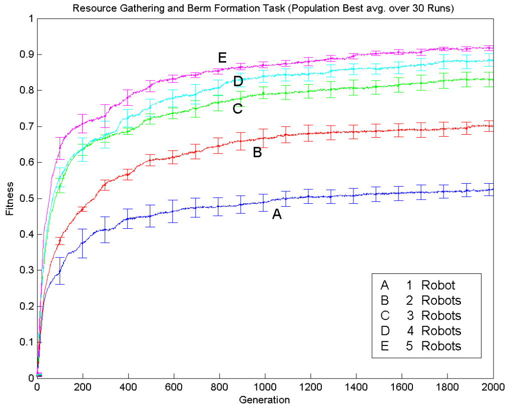


Fig. 6. Evolutionary performance comparison of ANT-based solutions for 1 to 5 robots. Error bars indicate standard deviation.

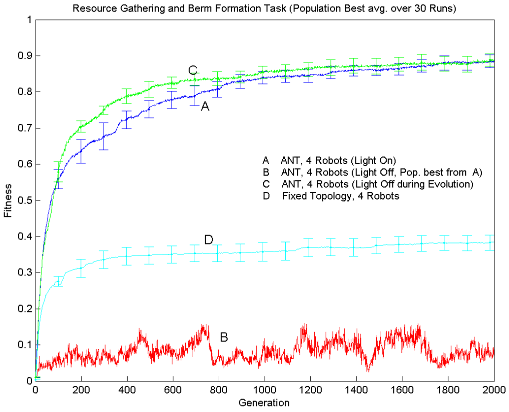


Fig. 7. Evolutionary performance with 4 robots for fixed topology and with light beacon off. Error bars indicate standard deviation.

V. RESULTS AND DISCUSSION

Fig. 6 shows the fitness (population best) of the overall system evaluated at each generation of the artificial evolutionary process. The performance of a fixed-topology, fully-connected network with 12 hidden and output neurons is also shown in Fig. 7. While this is not intended as a benchmark network, in a fixed-topology network there tends to be more "active" synaptic connections present (since all neurons are active), and thus it takes longer for each neuron to tune these connections for all sensory inputs.

The results with ANT controllers in Fig. 6 show that performance increases with the number of robots. With more robots, each robot has a smaller area to cover in trying to gather and dump resources. The simulation runs indicate that a point of diminishing returns is eventually reached. Beyond this point, additional robots have a minimal effect on system performance with the initial resource density and robot density kept constant. The evolutionary process enables the decomposition of a goal task based on global fitness, and the tuning of behaviors depending on the robot density.

In an ANT-based architecture, networks are dynamically formed with decision neurons processing the sensory input and in turn "selecting" motor-control neurons through the coarse-coding framework [3]. The behavioral activity of the controllers (see Fig. 10) shows the formation of small networks of neurons, each of which handles an individual behaviors, such as dumping resources or detecting visual templates (boundary perimeters, target area markings, etc.). Localized regions within the tissue do not exclusively handle these specific user-defined, distilled behaviors. Instead, the activity of the decision neurons indicate distribution of specialized "feature detectors" among independent networks.

A. Behavioral Adaptations

Some of the emergent solutions evolved indicate that the individual robots all figure out how to dump nearby resources into the designated berm area, that but not all robots deliver resource all the way to the dumping area every time. Instead, the robots learn to pass the resource material from one individual to another during an encounter, forming a "bucket brigade" (see Fig. 9). This technique improves the overall efficiency of system as less time is spent traveling to and from the dumping area. Since the robots cannot explicitly communicate with one another, these encounters happen by chance rather than through preplanning. As with other multiagent systems, communication between robots occurs through the manipulation of the environment in the form of stigmergy. The task in [14] is similar in that distributed objects must be delivered to a confined area; however, the hand-designed controller does not scale as well as the "bucket brigade" solution that the ANT framework discovered here.

While the robot controllers can detect and home in on a light beacon (exhibiting "phototaxis"), this source of navigation is not always used. Although not necessary, the light beacon helps in navigation by allowing the robots to locate the dumping area. It is notable that the overall fitness is unaffected when the controllers are evolved with the light source turned off. However, when the light source is turned on, the controllers do make use of it to navigate to the dumping area even though this does not appear to translate into a higher fitness (see Fig. 7). When a controller evolved with the light source on is tested with the light source off, the fitness is substantially lower.

Phototaxis requires a means to determine the light gradient and intensity, both of which are made available in these simulations. The robot model assumes the light-detection sensor is directional and has a limited field of view. Hence, once a robot faces away from the light source, the light detection sensor is in the "Not Visible" state. Although the light intensity input appears to be unused, light direction values appears to be used intermittently (see Fig. 9) in combination with other visual templates. When a robot faces a boundary region, the direction of the light source is used to determine whether to "Turn Right" or "Turn Left." This behavior in turn is triggered when the robot had accumulated resources along the way.

In our simulations, the dumping area is usually next to the boundary region. Where possible, the controllers appear to

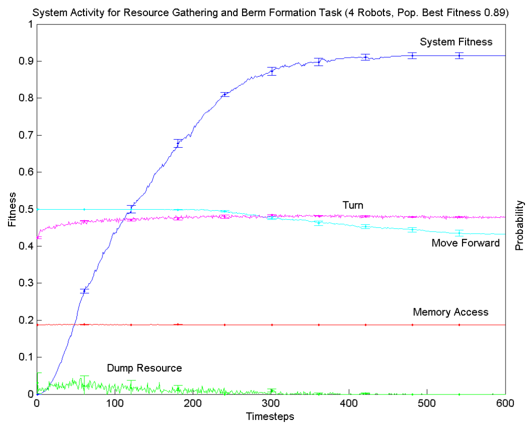


Fig. 8. System activity for the resource-collection task.

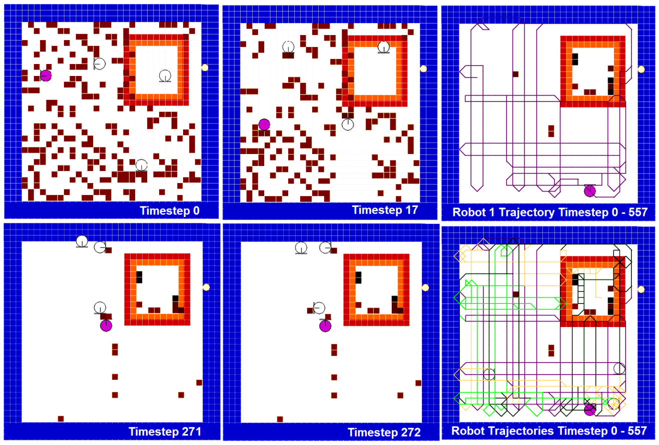

Timestep 0
Timestep 17
Robot 1 Trajectory Timestep 0 - 557
Timestep 271
Timestep 272
Robot Trajectories Timestep 0 - 557

Fig. 9. Snapshots of robots and trajectories of a task simulation (4 robots).

perform "sensor fusion," making use of data from multiple sensors to perform navigation to and from the dumping area.

In these simulation experiments, the robots have no way to measure the remaining time available; hence, the system cannot greedily accumulate resource materials without periodically dumping the material at the designated area. This explains why we see a steady increase in the amount of resource material gathered over time (see Fig. 8).

B. *Controller Scalability*

We examine the fittest solutions from the simulation runs shown in Fig. 11 for scalability in the number of robots while holding the amount of resources constant. Taking the controller evolved for a single robot and running it on multirobot system shows limited performance improvement. In fact, using four or more robots results in a *decrease* in performance, due to the increased antagonism created.

The scalability of the evolved solution depends in large part on the number of robots used during the training runs. The single-robot controller expectedly lacks the cooperative behavior necessary to function well within a multiagent setting. For example, such controllers fail to develop "robot collision avoidance" or "bucket brigade" behaviors. Similarly, the robot controllers evolved with two or more

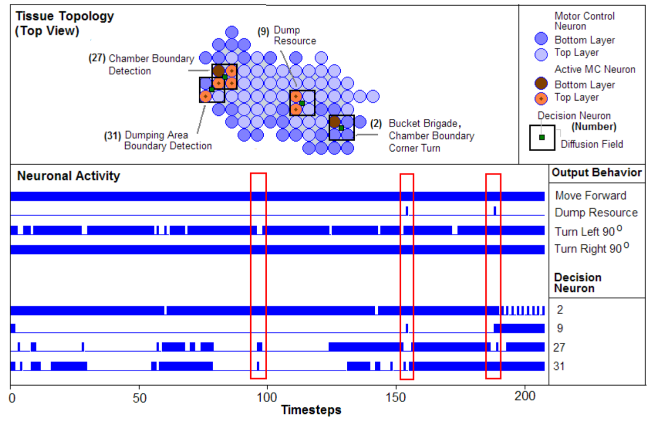


Fig. 10. Tissue Topology and neuronal activity of a select number of decision neurons. These decision neurons in turn "select" (excite into operation) motor control neurons within its diffusion field.

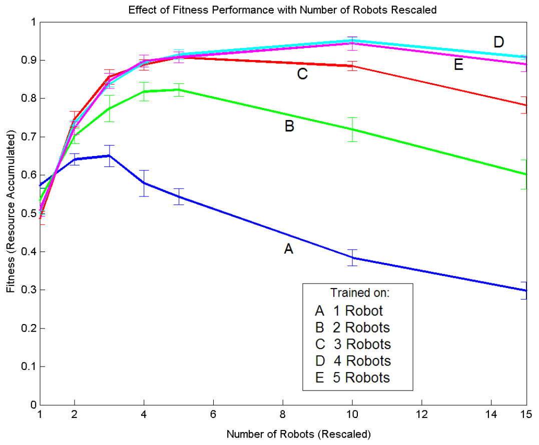


Fig. 11. Scaling of ANT-based solutions from one to five robots.

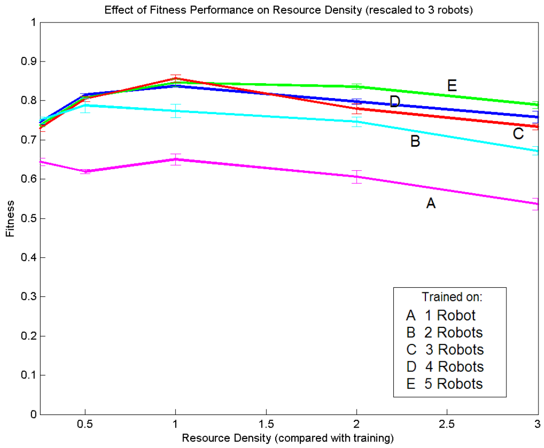


Fig. 12. Scaling of ANT-based solutions with change in resource density.

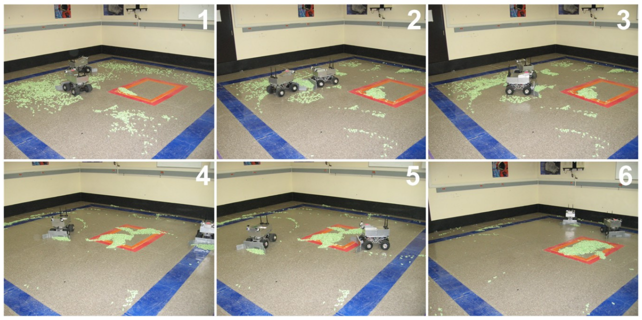


Fig. 13. Snapshots of two rovers performing the resource collection task using an ANT controller. Frames 2 and 3 show the "bucket brigade" behaviour, while frames 4 and 5 show the boundary avoidance behaviour.

robots perform demonstrably worse when scaled down to a single robot, showing that the solutions are dependent on cooperation among the robots. To limit the effects of antagonism, controllers would need to be trained under conditions in which the probability of encounters among robots is sufficiently high.

The effect of changes in resource density is also examined in the evolved solutions (see Fig. 12). Keeping the number of robots and the maximum number of time-steps constant while increasing the amount of resource material, the percentage of resource material accumulated in the dumping area appears to decline steadily. The peak performance appears to occur when the resource density is at or near the training conditions.

VI. CONCLUSIONS

The "Artificial Neural Tissue" (ANT) architecture has been successfully applied in simulation and on hardware to a multirobot resource-collection task. ANT controllers require only a global fitness function that measures the performance of the controller and a generic set of basis behaviors. Because little preprogrammed knowledge is given, an ANT architecture may discover novel solutions that might otherwise be overlooked by a human supervisor. ANT controllers are shown to exploit a number of mechanisms known to be used in unsupervised multiagent systems nature, including environmental templates, stigmergy, and self-organization. An analysis of the scalability of ANT-based multirobot controllers shows ways in which we can reduce the effects of antagonism caused both by increasing the number of robots and scaling the resource density.

The hardware experiments have shown that an ANT controller can be developed in simulation and successfully ported to a real rover system. The interesting emergent behaviors seen in simulation (such as obstacle avoidance and "bucket brigades", see Fig. 13) were also evident in the hardware tests. In addition, a noticeable improvement in fitness was evident when moving from a single to a two-rover system.

VII. ACKNOWLEDGEMENTS

The authors would like to thank the reviewers for their helpful comments. This work was partially funded by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

[1] K. L. Bristow and J. A. Holt, "Can termites create local energy sinks to regulate mound temperature?" in *Journal of Thermal Biology*, vol. 12, 1987, pp. 19–21.

[2] C. Melhuish, J. Welsby, and C. Edwards, "Using templates for defensive wall building with autonomous mobile ant- like robots," in *TIMR99 Toward Intelligent Mobile Robots*, 1999.

[3] J. Thangavelautham and D. G. M. T., "A coarse-coding framework for a gene-regulatory-based artificial neural tissue," in *Advances In Artificial Life: Proc. of the 8th European Conference on ALife*, 2005, pp. 67–77.

[4] J. S. Albus, *Brains, Behavior and Robotics*. BYTE Books, McGraw-Hill, Nov 1981.

[5] M. J. Matarić, M. Nilsson, and K. T. Simsarian, "Cooperative multi-robot box-pushing," in *IEEE/RSJ IROS*, 1995, pp. 556–561.

[6] C. A. Parker, H. Zhang, , and C. R. Kube, "Blind bulldozing: Multiple robot nest construction," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2003, pp. 2010–2015.

[7] M. Wilson, C. Melhuish, A. B. Sendova-Franks, and S. Scholes, "Algorithms for building annular structures with minimalist robots inspired by brood sorting in ant colonies," in *Autonomous Robots*, vol. 17, 2004, pp. 115–136.

[8] J. Werfel, Y. Y., and R. Nagpal, "Building patterned structures with robot swarms," in *IJCAI*, 2005, pp. 1495–1502.

[9] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford Univ. Press, 1999.

[10] P. Grassé, "La reconstruction du nid les coordinations interindividuelles; la theorie de stigmergie," in *Insectes Sociaux*, vol. 35, 1959, pp. 41–84.

[11] E. Bonabeau, G. Theraulaz, J.-L. Deneubourg, S. Aron, and S. Camazine, "Self-organization in social insects," in *Trends in Ecology and Evolution*, vol. 12, May 1997, pp. 188–193.

[12] F. Chantemargue, T. Dagaeff, M. Schumacher, and B. Hirsbrunner, "Implicit cooperation and antagonism in multi-agent systems," University of Fribourg, IIUF-PAI, Internal Report, Tech. Rep., 1996.

[13] R. Stewart and A. Russell, "Emergent structures built by a minimalist autonomous robot using a swarm-inspired template mechanism," in *The First Australian Conference on ALife (ACAL2003)*, 2003, pp. 216–230.

[14] J. Wawerla, G. Sukhatme, and M. Matari, "Collective construction with multiple robots," in *IEEE/RSJ Int.l Conference on Intelligent Robots and Systems*, 2002, pp. 2696–2701.

[15] R. Stewart and A. Russell, "Building a loose wall structure with a robotic swarm using a spatio-temporal varying template," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2004, pp. 712–716.

[16] R. Beckers, O. E. Holland, and J. L. Deneubourg, "From local actions to global tasks: Stigmergy and collective robots," in *Fourth Int. Workshop on the Syntheses and Simulation of Living Systems*. MIT Press, 1994, pp. 181–189.

[17] J. Thangavelautham, T. Barfoot, and D. G.M.T., "Coevolving communication and cooperation for lattice formation tasks (updated)," in *Advances In Artificial Life: Proc. of the 7th European Conference on ALife (ECAL)*, 2003, pp. 857–864.

[18] T. D. Barfoot and G. M. T. D'Eleuterio, "An evolutionary approach to multiagent heap formation," in *1999 Congress on Evolutionary Computation (CEC)*, 1999, pp. 427–435. [Online]. Available: http://sr.utias.utoronto.ca/papers/cec99.pdf

[19] J. Thangavelautham and D. G. M. T., "A neuroevolutionary approach to emergent task decomposition," in *8th Parallel Problem Solving from Nature Conference*, vol. 1, 2004, pp. 991–1000.

[20] F. L. Crabbe and M. G. Dyer, "Second-order networks for wall-building agents," in *Int. Joint Conference on Neural Networks*, vol. 3, 1999, pp. 2178–2183.

[21] V. Trianni and M. Dorigo, "Self-organisation and communication in groups of simulated and physical robots," in *Biological Cybernetics*, vol. 95. Springer Berlin / Heidelberg, Sep. 2006, pp. 213–231.

[22] J. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press, 1975.

[23] T. Hastie, R. Tibshirani, and R. Friedman, *The Elements of Statistical Learning*. New York: Springer, 2001.