A Neuroevolutionary Approach to Emergent Task Decomposition

J Thangavelautham and Gabriele M T D'Eleuterio

Institute for Aerospace Studies, University of Toronto
Toronto, Ontario, Canada, M3H 5T6
thangav@ecf.utoronto.ca, gabriele.deleuterio@utoronto.ca

Abstract. A scalable architecture to facilitate emergent (self-organized) task decomposition using neural networks and evolutionary algorithms is presented. Various control system architectures are compared for a collective robotics (3×3 tiling pattern formation) task where emergent behaviours and effective task -decomposition techniques are necessary to solve the task. We show that bigger, more modular network architectures that exploit emergent task decomposition strategies can evolve faster and outperform comparably smaller non emergent neural networks for this task. Much like biological nervous systems, larger Emergent Task Decomposition Networks appear to evolve faster than comparable smaller networks. Unlike reinforcement learning techniques, only a global fitness function is specified, requiring limited supervision, and self-organized task decomposition is achieved through competition and specialization. The results are derived from computer simulations.

1 Introduction

Some of the fundamental goals of machine learning is to develop learning algorithms that require limited supervision but can facilitate the discovery of novel solutions and better handle environmental uncertainties. We have been inspired by social insects such as ants, bees and termites that form *synergistic* (one whose capabilities exceed the sum of its parts) multiagent systems without any centralized supervision. These insects societies consist of simple individuals and as a collective can solve complex tasks. For example, termites have evolved the ability to survive the harsh Australian outback by building towering nests with internal heating and cooling shafts [3].

Evolutionary algorithms (EAs) are specially suited for non-Markovian collective robotics tasks, since only a global fitness function needs to be specified to obtain a desired collective behaviour. Unfortunately, the application of evolutionary algorithms for increasingly complex robotics control tasks has been very challenging due to the *bootstrap problem* [11]. The bootstrap problem occurs when it becomes increasingly difficult for the EAs (particularly for a monolithic control system topology) to pick out incrementally better solutions for crossover and mutation resulting in premature stagnation of the evolutionary run.

A common solution to the bootstrap problem is to decompose a complex task into a set of simpler tasks through supervised task decomposition. This often requires a priori information of the task and supervisor intervention in determining suitable task decomposition strategies. Yet in nature, biological systems can accomplish such feet with no such external intervention.

In this paper we introduce a modular Emergent Task Decomposition Network (ETDN) architecture that requires limited supervision and can decompose a complex task into a set of simpler tasks through competition and self organization. The ETDN architecture is trained using evolutionary algorithms, with only a prespecified global fitness function. The network is composed of *decision neurons* (mediates competition) and several *expert networks* (that compete for dominance).

Limited supervision provides numerous advantages including the ability to discover novel solutions that would otherwise be overlooked by a human supervisor. This strategy is particularly advantageous for decentralized collective robotics where little is known of the interaction of local behaviours resulting in a desired global behaviour. We empirically compare the training performance (using evolutionary algorithms) of various control systems for the tiling pattern formation task (similar to a segment of the termite nest building task [3]). It was found earlier that a memorisation approach (such as a monolithic look-up table-based control system) is unable to solve more complex versions of the task because of the bootstrap problem [13].

2   Related Work

Dorigo and Colombetti [4] introduced the idea of *incremental learning* (*shaping*) in classifier systems, where a robot learns a simplified version of the task and the task complexity is progressively increased by modifying the learning function. Stone and Veloso [12] developed a task decomposition scheme known as *layered learning*, where the learning function for a complex task (playing soccer) is partitioned into a set of simpler learning functions (corresponding to sub-tasks) and learned sequentially. These traditional task decomposition techniques require the supervisor to have domain knowledge of the task and are very difficult to implement in the collective robotics domain since the necessary local and global (collective) behaviours need to be known.

Jordan and Jacob [8] developed an automated task decomposition technique using a modular neural network architecture for the 'what' and 'where' vision task . The architecture consists of a decision network (mediates competition) and several expert modules (explicitly specialized for predefined subtasks). The expert networks and the decision networks are trained separately according to a set of handcrafted reinforcement learning algorithms.

In evolutionary computation, algorithms such as ESP (Enforced Subpopulations) and SANE (Symbiotic Adaptive Neuro-Evolution) evolve individual neurons, within a fixed neural network topology [6]. ESP uses separate subpopulations to select neurons for each placeholder within the neural network. This

technique has been used to evolve control systems for keep-away soccer robots, sounding rockets and for the double-pole balancing task [14, 6, 7]. Unfortunately, this approach is susceptible to early convergence in some cases, since the algorithm attempts to pick the best set of individual neurons rather than the best combination of neurons within a network [6].

In evolutionary robotics, Nolfi [10] introduced *emergent modular architectures*, where decision neurons arbitrate individual motor control neurons for the garbage collection task [11]. This architecture evolved faster and produced better solutions than other comparable networks including multilayered feed forward and recurrent architectures. Ziemke [15] attempted to determine the scalability of the *emergent modular architecture* and was unsuccessful in his efforts. He found *emergent modular architectures* were particularly inefficient when integrated with memory neurons.

3 Method

Some of the control system architectures presented in this paper consist of artificial neural networks trained using evolutionary algorithms (neuroevolution). We use a fixed network topology, with a 1:1 genotype-to-phenotype mapping and where weights, biases/thresholds are evolved using a global fitness function. Some of the advantages of neuroevolution, over reinforcement learning is that the search space need not be smooth nor differentiable and solutions tend to be more robust against local minima. In addition, unlike traditional machine learning algorithms, EAs could be used to select suitable activation functions during training (as shown with our modular neuron architecture).

3.1 Emergent Task-Decomposition Architecture

We propose an Emergent Task-Decomposition Network (ETDN) architecture that consist of a set of decision neurons that mediate competition and a set of expert network modules that compete for dominance (see Fig. 1b). We exploit network modularity, evolutionary competition, specialization and *functional neutrality* to facilitate emergent (self organized) task decomposition
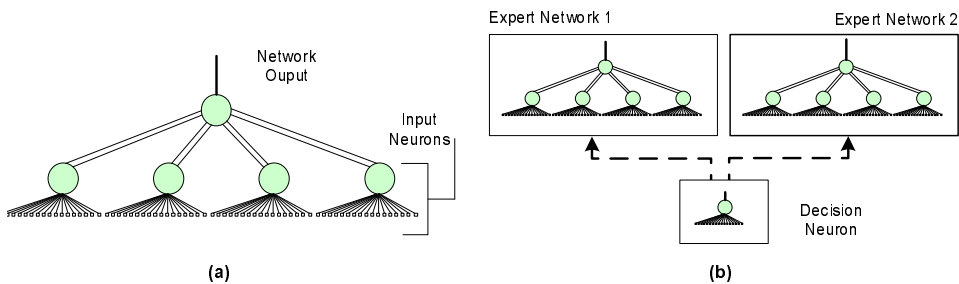

Fig. 1. (a) An example of the non-emergent network used in our experiments. (b) ETDN architecture consisting of a decision neuron that arbitrates between 2 expert networks.

Unlike traditional machine learning methods where handcrafted learning functions are used to train the decision and expert networks separately, our architecture requires only a global fitness function. The intent is for the architecture to evolve the ability to decompose a complex task into a set of simpler tasks with limited supervision. The decision neuron is connected to all the sensory input and is used to select an expert network based on the output state. In turn, the output from the selected expert network triggers a set of predefined actions.

In nature, it is well known that the brain has adopted a modular architecture. Ballard [1] goes further and suggests that a limitation in the number of neurons in the brain (due to limited volume) have forced the brain to evolve a modular architecture. Geschwind and Galaburda[5] suggest competition between different networks of neurons in the brain drives certain networks to become specialized for a particular function . At the molecular level, it is well known the interaction of functional proteins is determined through collaborative competition for limited resources and binding sites, thus leading to functional specialization [2].

3.2 Extending Emergent Task Decomposition

Our proposed ETDN architecture unlike previous work in the field of evolutionary computation, could be generalized for n_E expert networks. Here we discuss two proposed extensions to the ETDN architecture, namely the Binary Relative Lookup (BRL) architecture and Binary Decision Tree (BDT) architecture.

The Binary Relative Lookup (BRL) architecture consists of a set of n_d non-interconnected decision neurons that arbitrate between 2^{n_d} expert networks. Starting from left to right, each additional decision neuron determines the specific grouping of expert networks relative to the selected group. Since the decision neurons are not interconnected (see Fig. 2), this architecture is well suited for parallel implementation.
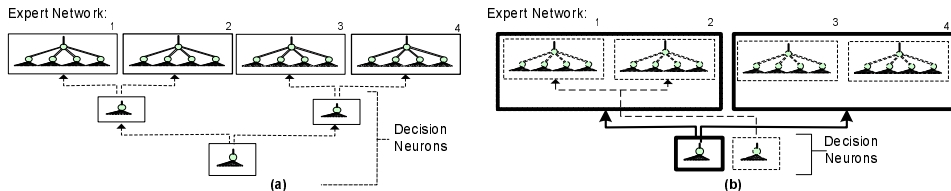

Fig. 2. (left) BDT Architecture with 4 expert networks and 3 decision neurons. (right) BRL Architecture with 4 expert networks and 2 decision neurons.

The Binary Decision Tree (BDT) architecture could be represented as a binary tree where the tree nodes consist of decision neurons and the leaves consist of expert networks. For this architecture, n_d decision neurons arbitrate between $n_d + 1$ expert networks. The tree is traversed by starting from the root and computing decisions neurons along each selected branch node until an expert network is selected. For both architectures the computational cost of the decision neurons, $C_d \propto \log n_E$.

3.3 Modular Neurons

We have also developed a modular threshold activation neuron, where the EAs are used to train the weights, threshold parameters and choice of activation function for each neuron. The inputs and outputs from the modular threshold neurons consist of states rather than arbitrary values. The modular neuron could assume one of four different activation functions listed shown below:

$$
\phi_1: \quad s_{out} = \begin{cases} s_1, & \text{if } p(x) \leq t_1 \\ s_2, & \text{if } p(x) > t_1 \end{cases}
\qquad
\phi_3: \quad s_{out} = \begin{cases} s_1, & \text{if } t_2 < p(x) < t_1 \\ s_2, & \text{otherwise} \end{cases}
$$

$$
\phi_2: \quad s_{out} = \begin{cases} s_1, & \text{if } p(x) \geq t_2 \\ s_2, & \text{if } p(x) < t_2 \end{cases}
\qquad
\phi_4: \quad s_{out} = \begin{cases} s_1, & \text{if } p(x) > (1 - p(x)) \\ \text{rand}(s_1, s_2), & \text{if } p(x) = (1 - p(x)) \\ s_2, & \text{if } p(x) < (1 - p(x)) \end{cases}
$$

$$ \tag{1} $$

Each neuron outputs one of two states $s = (s_1, s_2)$, where t_n is a threshold, $p(x) = (\sum_i w_i x_i)/(\sum_i x_i)$, w_i is a neuron weight and x_i is an element of the active input state vector.

Our intention was to develop a compact modular neuron architecture for implementation on hardware, where a single neuron could be used to simulate AND, OR, NOT and XOR functions. The assumption is that a compact yet sufficiently complex (functional) neuron will speed up evolutionary training since this will reduce the need for more hidden layers and thus result in smaller networks.

4 Example Task: Tiling Pattern Formation Task

The tiling pattern formation task [13] involves redistributing objects (blocks) piled up in a 2-D world into a desired tiling structure (see Fig. 7). The robots need to come to a consensus and form one 'perfect' tiling pattern. This task is similar to a segment of the termite nest construction task that involves redistributing pheromone filled pellets on the nest floor [3]. Once the pheromone pellets are uniformly distributed, the termite use the pellets as markers for constructing pillars to support the nest roof.
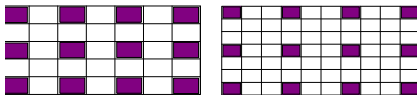

Fig. 3. The 2×2 tiling pattern (left) and 3×3 tiling pattern (right).

More important, the tiling pattern formation task could be decomposed into a number of potential subtasks consisting of emergent behaviours. These may include foraging for objects (blocks), redistributing block piles, arranging blocks in the desired tiling structure locally, merging local lattice structures, reaching a collective consensus and finding/correcting mistakes in the lattice structure. Inspired by nature, we are interested in evolving homogenous decentralized controllers (similar to a nest of termites) for the task [9]. Decentralized control offers

some inherent advantages including the ability to scaleup to a larger problem size. Task complexity is dependent on the intended tile spacing, since more sensor data would be required to construct a 'wider' tiling pattern.

Shannon's entropy function has been shown to be a suitable fitness function for the tiling formation task [13]. The test area spans $M \times M$ squares and is divided into J $l \times l$ cells, A_j, where the fitness value, $f_{i,x,y}$, for one set of initial condition i, after T discrete time steps, with cells shifted x squares in the x-direction and y squares in the y-direction is given as follows:

$$f_{i,x,y} = s \cdot \frac{\sum_{j=1}^{J} p_j \ln p_j}{\ln J} \tag{2}$$

where, $s = 100$ and is a constant scaling factor, I is an index over a set of initial conditions and $p_j = (n(A_j))/(\sum_{j=1}^{J} n(A_j))$, where $n(A_j)$ is the number of blocks in cell A_j. To encourage the desired tiling pattern, the fitness function is applied by shifting the cells a maximum of $l - 1$ squares and the total fitness, $f_i = (\sum_{y=0}^{l-1}(\sum_{x=0}^{l-1} f_{i,x,y}))/l^2$. When the blocks are uniformly distributed over J cells, according to the desired tiling pattern, we have $f_i = 100$ (a successful epoch).
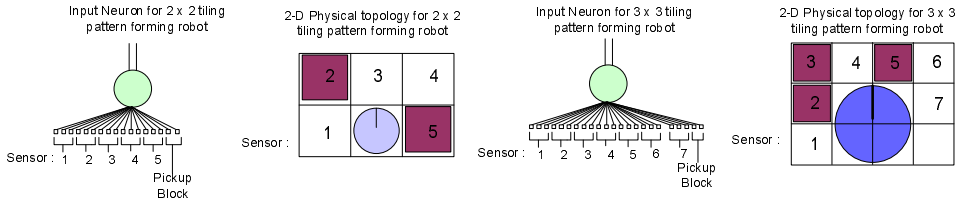
4.1 The Robot

Fig. 4. Input layer neuron and physical topology for the 2×2 (left) and 3×3 (right) tiling pattern forming robots.

The robots are modelled as modified Khepera robots equipped with a gripper turret. We have developed a fast 2-D grid world simulator for our evolutionary experiments and we have verified this simulator using Webots (Fig. 7). For the 2×2 tiling pattern formation task, each robot could detect blocks and other robots in the 5 squares as shown (Fig. 4). For the 3×3 tiling pattern formation task, the robots could detect object in 7 surrounding squares as shown. The output state from the robot controller activates one of two predefined basis behaviours, namely **Move** and **Manipulate Object** described below.

  I Move: Move forward if square to the front empty, otherwise turn left if square to the left empty otherwise turn right.
 II Manipulate Object: Pick up/put down a block from the surrounding squares according to a predefined sequence of steps depending on possession of a block or not. If the robot is unable to pick up or put down a block, the 'move' command is activated.

5 Experimentation

The evolutionary performance of various control systems architectures are compared for the tiling pattern formation task (see Fig. 5). Through experimentation, we found the best nonmodular network for the tiling pattern formation task as shown in Fig. 1(a). We use this network as our expert network module for the ETDN architectures. In our simulations the EA population size is $P = 100$, number of generations $G = 200$, crossover probability $p_c = 0.7$, mutation probability $p_m = 0.025$ and tournament size of $0.06 p_c$ (for tournament selection). The fitness is evaluated after $T = 3000$ timesteps for a 11×11 world (2×2 task) or 16×16 world (3×3 task) with 11 robots and 36 blocks.
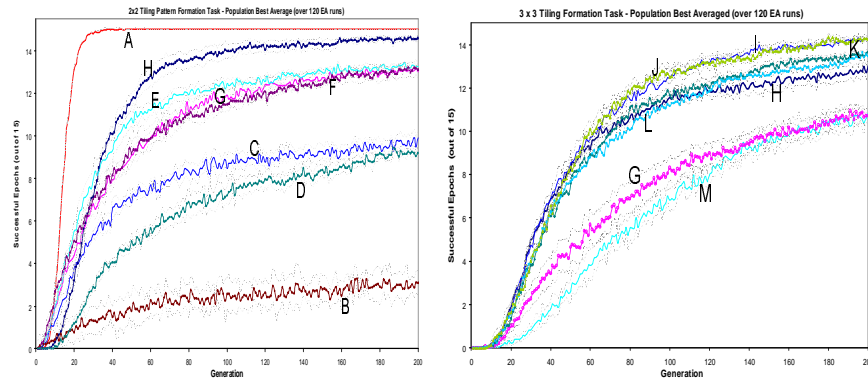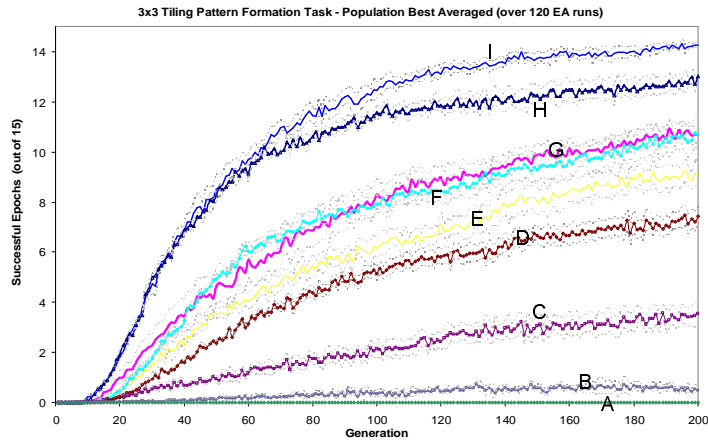

Fig. 5. Evolutionary performance comparison, 2×2 (left), 3×3 (right, bottom) tiling pattern formation task, averaged over 120 EA runs. **(A)** Look-up Table, **(B)** ESP (using Non-Emergent Net), **(C)** Non-Emergent Net(Sigmoid),**(D)** ETDN (Exp. Net 2, Sigmoid), **(E)** Non-Emergent Net. (Threshold), **(F)** ETDN (2 Exp. Nets, Threshold), **(G)** Non-Emergent Net. (Modular), **(H)** ETDN (2 Exp. Nets, Modular),**(I)** BRL (16 Exp. Nets, Modular), **(J)** BRL (32 Exp. Nets, Modular), **(K)** BRL (8 Exp. Nets, Modular), **(L)** BRL (4 Exp. Nets, Modular), **(M)** BDT (4 Exp. Nets, Modular)

6 Results and Discussion

For the 2×2 tiling pattern formation task, the lookup table approach evolves desired solutions faster than the network architectures. This suggests that ETDNs may not be the most efficient strategy for smaller search spaces. Our conventional ETDN architecture consisting of a single threshold activation function evolves slower than the nonemergent architectures.

The ETDN architectures include an additional 'overhead', since the evolutionary performance is dependent on the evolution of the expert networks and decision neurons resulting in slower performance for simpler tasks. However the Emergent Task Decomposition Network architecture that combines our modular neuron architecture outperforms all other network architectures. The performance of the modular neurons appears to offset the 'overhead' of the bigger ETDN architecture. A 'richer' activation function is hypothesized to improve the ability of the decision neurons to switch between suitable expert networks.

For the more difficult 3×3 tiling formation task, a lookup table architecture is unable to find a desired solution. The resulting search space is significantly larger (owing to increased number of sensors required), 2^{4374} potential solutions compared to 2^{486} for the 2×2 tiling formation task. With a very large search space, this architecture falls victim to the bootstrap problem, since EAs are unable to find an incrementally better solution during the early phase of evolution. The ESP algorithm learns slower than the other architectures for both tiling formation tasks. It is suspected that evolving individual neurons in parallel may not be the most effective strategy in facilitating task decomposition for this task.
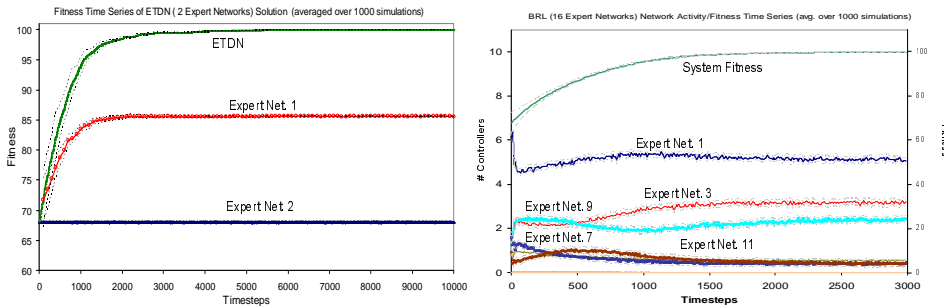

Fig. 6. (left) Fitness of individual expert networks and the entire control system (ETDN, 2 expert nets.) averaged over 1000 simulations. (right) Expert Network activity and overall system fitness for a BRL (with 16 expert networks).

The ETDNs outperform non-emergent network architectures for the more complex 3×3 tiling formation task (regardless of the activation function used). Analysis of a typical solution (for ETDN with 2 expert nets) suggests the expert networks have specialized since the individual fitness performance of the networks is quite low (Fig. 6). It appears the decision neuron arbitrates among the expert networks not according to 'recognizable' distill behaviours but as a set of emergent proximal behaviours (organized according to proximity in sensor space) [10].

Although conventional network architecture can perform limited task decomposition, (evident from solutions to the 3×3 tiling formation task), this process is more prone to *spatial crosstalk* [8] resulting in slower evolutionary performance. For the ETDNs, it appears, the task is distributed over several expert networks resulting in fewer hidden neurons being used as feature detectors within each network, thus reducing the overall effect of spatial crosstalk. The decision neurons within the BRL architecture act more 'independently' when selecting an expert network than the BDT network. We suspect this characteristic, in addition to the fewer number of decision neurons for the BRL network improves the ability to select suitable expert networks. This is evident from Fig. 5, where the BDT with 4 expert networks evolves slower than a comparable BRL network.

Much like biological nervous systems, the larger BRL architecture outperformed (or performed as fast as) the smaller ones (evident after about 80 generations). It is hypothesized that by increasing the number of expert networks, competition among candidate expert networks is further increased thus improving the chance of finding a desired solution. However, as the number of expert networks is increased (beyond 16), the relative improvement in performance is minimal, for this particular task.

Table 1: Success rate of population best (generation 200) scaled up to a 100×100 world (76 robots, 1156 blocks)

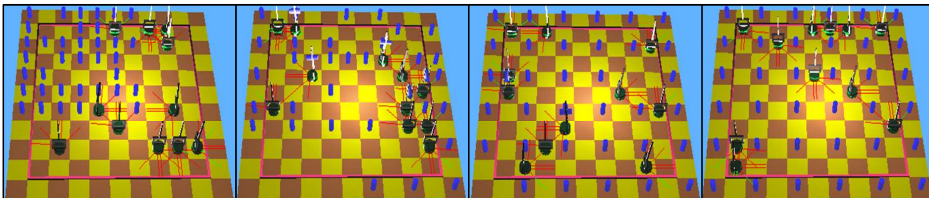| Architecture	% Successful Epochs	% Batch Variance
Non-Emergent (Sigmoid)	0.0	-
Non-Emergent (Modular)	27.0	0.2
ETDN (2 Expert Nets., Modular)	44.5	2.2
BRL (16 Expert Nets., Modular)	70.2	0.4

Fig. 7. Simulation snapshots taken after 0, 100, 400 and 410 timesteps using a evolved solution for the 2×2 tiling formation task on a 11×11 world (11 robots, 36 blocks). The robots reach a consensus and form a 'perfect' tiling pattern after 410 timesteps.

The larger BRL network solutions generally evolve the ability to limit the number of expert networks used (see Fig. 6 for a typical solution). The other expert networks remain dormant throughout the simulations. The ability to limit the number of expert networks used for task decomposition is particularly advantageous, since 'over segmenting' the task into subtasks would result in over-fitting. Minor differences in fitness performance and 'success rates' for the 19×19 training world is 'magnified', once we scaleup to a larger problem size (100×100 world). This is where our ETDN and in particular BRL architectures is more advantageous over conventional network architectures (see Table 1).

7 Conclusion and Future Work

A scalable mechanism for emergent task decomposition using artificial neural networks and evolutionary algorithms is presented in this paper. Unlike reinforcement learning techniques, only a global fitness function is specified (unsupervised) and the network is able to decompose a complex task into simpler tasks through self-organization. Emergent Task Decomposition Networks (ETDN) evolve faster than non-emergent architectures and produce more desired solutions for complex tasks. It is interesting that much like biological nervous systems, larger BRL architectures tend to evolve faster than comparable smaller architectures. We are currently planning to compare the performance of our ETDN architecture with other supervised task decomposition methods for potential multirobot control tasks such as robotic soccer, mining and surface exploration.

References

1. Ballard, D.H., Cortical Connections and parallel processing. The Behavioural and Brain Sciences, Vol. 9, (1986) 279–284
2. Banzhaf, W., On the Dynamics of an Artificial Regulatory Network. In Adv. in ALife: Proc. Of 7th European Conf. on ALife (2003) 217–226
3. Deneubourg J-L. Application de l'ordre par fluctuations 'a la description de certaines 'etapes de la construction du nid chez les termites. Insectes Sociaux, (1977) 117–130
4. Dorigo M., Colombetti M., Robot Shaping: Developing autonomous agents through learning. Artificial Intelligence, No. 71, (1994) 321-370
5. Gecshwind, N., Galaburda A.M., Cereberal Lateralization : Biological Mechanisms, Associations, and Pathology, MIT Press (1987)
6. Gomez, F., Miikkulainen R., Solving Non-Markovian Control Tasks with Neuroevolution In Proc. of the Int. Joint Conf.on Artificial Intelligence (1999)
7. Gomez, F., Miikkulainen, R., Active Guidance for a Finless Rocket using Neuroevolution. In the Proc. of Genetic and Evolutionary Comp. Conf.,(2003)
8. Jacobs R., Jordan M., Barto A., Task decomposition through competition in a modular connectionist architecture. Cognitive Science, No. 15, (1991) 219-250
9. Kube, R., Zhang, H.: Collective Robotics Intelligence : From Social Insects to robots. In Proc. Of Simulation of Adaptive Behavior (1992) 460–468
10. Nolfi, S., Using Emergent modularity to develop control systems for mobile robots., Adaptive Behaviour 5, No. 3, ISAB, (1997) 343-363
11. Nolfi, S., Floreano D.: Evolutionary Robotics : The Biology, Intelligence, and Technology of Self-Organizing Machines, MIT Press (2000) 13-15
12. Stone, P., Veloso M., Layered Learning, Proc. of 11th European Conf. on Machine Learning (2000) 369-381
13. Thangavelautham, J., Barfoot, T., D'Eleuterio, G.M.T., Coevolving Communication and Cooperation for Lattice formation Tasks In Adv. in ALife: Proc. Of 7th European Conf. on ALife (2003) 857-864
14. Whiteson, S. et. al.,Evolving Keep-away Soccer Players through Task Decomposition. In the Proc. of Genetic and Evolutionary Comp. Conf., (2003)
15. Ziemke, T. On 'Parts' and 'Wholes' of Adaptive Behavior, In From Animals to Animats 6: Proc. 5th Int. Conf, on Sim. of Adaptive Behaviour. MIT Press (2000)