Adaptive Behavior

http://adb.sagepub.com

Hierarchical Co-evolution of Cooperating Agents Acting in the Brain-Arena

Michail Maniadakis and Panos Trahanias

Adaptive Behavior 2008; 16; 221
DOI: 10.1177/1059712308087285

Adaptive Behavior

Hierarchical Co-evolution of Cooperating Agents Acting in the Brain-Arena

Michail Maniadakis, Panos Trahanias
Institute of Computer Science, Foundation for Research and Technology – Hellas (FORTH), Heraklion, Crete, Greece and Department of Computer Science, University of Crete, Heraklion, Crete, Greece

Recently, many brain-inspired models have been used in attempts to support the cognitive abilities of artificial organisms. In this article, we introduce a computational framework to facilitate these efforts, emphasizing the cooperative performance of brain substructures. Specifically, we introduce an agent-based representation of brain areas, together with a hierarchical cooperative co-evolutionary design mechanism. The proposed methodology is capable of designing biologically inspired cognitive systems, considering both the specialties of brain areas and their cooperative performance. The effectiveness of the proposed approach is demonstrated by designing a brain-inspired model of working memory usage. The co-evolutionary scheme enforces the cooperation of agents representing the involved brain areas, facilitating the accomplishment of two different tasks by the same model. Furthermore, we investigate the performance of the model in lesion conditions, highlighting the distinct roles of agents representing brain areas. The implemented model is embedded in a simulated robotic platform to support its cognitive and behavioral capabilities.

Keywords co-evolution · cooperative co-evolution · working memory · delayed response · cognitive robotics · cortical model

1 Introduction

The long-term vision of developing artificial organisms (robots) with mammal-like mental abilities can be facilitated by the development of brain-inspired cognitive computational systems. The brain is described as a group of cooperating specialists, which achieve the overall cognitive function by splitting the task into smaller elements (Reilly, 2001). These specialists (brain areas) need to cooperate in order to guide the organism for accomplishing tasks. Thus, the problem of implementing efficient brain-like cognitive systems can be stated in terms of designing a distributed architecture consisting of specialized, yet cooperating components.

Along this line, the modern software engineering approach for designing complex distributed systems (i.e., agent technology) can be employed to support the design procedure. This is because the agent technology matches adequately the distributed nature of the biological prototype. Specifically, each brain area can be represented by a separate agent, having a distinct role in the performance of the composite cognitive system. By adopting this approach, we gain considera-

Correspondence to: Michail Maniadakis, Institute of Computer Science, Foundation for Research and Technology, Hellas (FORTH), P.O. Box 1385, Heraklion, 711 10 Crete, Greece.
E-mail: mmaniada@ics.forth.gr

Figures 6–7, 9–10, 13–15 appear in color online: http://adb.sagepub.com

(2008), Vol 16(4): 221–245.
DOI: 10.1177/1059712308087285

ble design advantages because we are able to decompose the problem into smaller and easily solved tasks, highlighting at the same time the cooperation among partial brain areas. Despite the fact that many years ago Minsky (1986) described the brain as a set of cooperating agents, this view has only been used to design abstract cognitive systems, without addressing the connectivity of specific brain areas. To the best of our knowledge (see also Singh, 2003), this approach has not been adopted before in fields such as computational cognitive neuroscience aiming at implementing brain-like artificial systems.

In the current work, we follow an agent-based approach to support complex brain-modeling efforts. In contrast to the majority of agent technology applications where agents interact by means of the external environment, following the above-mentioned approach, the interaction of agents occurs internally to the cognitive system. Only the composite computational structure comes in contact with the external environment, mediated by the body of the artificial organism. In other words, the brain of the robot defines the operating environment of agents. Thus, a "brain-arena" is virtually specified, where many active agents co-exist, each one cooperating with, and modulating, the performance of the others. The successful coupling of agents and their effective cooperation is evaluated in terms of the final performance of the artificial organism.

The interactive non-linear dynamics among the components of the brain makes the design of the agent-based system a difficult task (Jennings, 2000). Various methodologies addressing agent design have been proposed in the literature. In this context, evolutionary techniques have been recently employed (Landau & Picault, 2001; Lee, 2003). However, by utilizing an ordinary evolutionary scheme that employs a single, compound representation to map problem solution to genotype, the structural nature of the problem is usually underestimated, or overlooked. Additionally, the operators of the evolutionary process typically have a uniform range of application, without considering the particular characteristics of the structure of the problem. In combination, the use of compound representations and uniform operators of variation makes it unlikely for partial solutions to persist, as no mechanism for protecting them is present. It also makes it unlikely for different combinations of partial solutions to be explored (De Jong, 2003; Thierens, 1999). Partial specifications of the genotype representing the components of the solution should be considered explicitly by the evolutionary process, in order to facilitate the exploration of cooperative dynamics among agents.

This issue is effectively addressed by co-evolutionary algorithms, utilizing separate populations to evolve each component of the solution (Potter & De Jong, 2000). Specifically, the evolution of partial populations supports separate exploration of each agent characteristic, while the composite co-evolutionary scheme investigates the dynamics of agent interaction in order to enforce their cooperative functioning. Thus, co-evolutionary algorithms are particularly appropriate for the design of systems consisting of cooperating agents.

We have recently introduced a computational framework to design brain-like cognitive systems following a co-evolutionary agent-based approach (Maniadakis, 2006; Maniadakis & Trahanias, 2005b, 2005c, 2006b). The organization of agent structures follows a biologically inspired block diagram outlining the connectivity of cortical areas, and the task-specific information they convey to each other. The co-evolutionary design mechanism assigns each agent a role similar to a respective brain area, emphasizing their biologically inspired characteristics in both an individual and a cooperative level. Additionally, co-evolution investigates the dynamics of partial component interaction, highlighting how each agent shapes communication and cooperation internally in the system. Thus, the co-evolutionary methodology supports the design of agents that act and cooperate successfully in the "brain-arena," accomplishing the desired performance of the overall system.

In the current work, we introduce a hierarchical extension of this approach, which exploits the inherent ability of co-evolutionary methods to integrate partial structures. Particularly, a hierarchical cooperative co-evolutionary (HCCE) scheme is utilized to co-evolve a large number of partial populations organized in gradually more complex groups. The HCCE is able to consider the cooperative performance of substructures in different operating conditions, and additionally enforce the reproduction of biological findings related to the adoption of diverse behavioral strategies.

Specifically, partial evolutionary processes performing at the lower levels of the HCCE hierarchy are driven by their own dynamics, aiming at replicating the performance of a particular brain area. An evolutionary process at the next higher level tunes lower level processes adapting one another, in order to achieve effective coupled performance of partial structures. The architecture of multiple co-evolutionary processes tuned by a higher level evolution can be repeated for as many levels as necessary, forming a tree-like hierarchy. Overall, a large number of components can be co-evolved, assembling successfully a complex distributed system.

Following recent trends aiming at the study of computational models in lesion conditions (Aharonov, Segev, Meilijson, & Ruppin, 2003; Goel, Pullara, & Grafman, 2001; Polk, Simen, Lewis, & Freedman, 2002), the agent-based co-evolutionary framework facilitates systematic modeling of biological lesion experiments. Specifically, lesion conditions are simulated by simply deactivating appropriate agent structures. Thus, the pre- and post-lesion performance of the model can be considered during the co-evolutionary design process. Furthermore, appropriate fitness functions are specified, indicating the performance of the model when all partial structures are present, and also indicating its performance when some structures are deactivated. Following this approach, biological lesion results can be replicated by the co-evolutionary design process, enforcing the similarity of the model to the brain prototype.

The effectiveness of the proposed computational framework is demonstrated by means of a specific modeling study that addresses the training of rodents accomplishing delayed response (DR) tasks in a T-maze, adopting either a delayed matching to sample (DMS) or a delayed non-matching to sample (DNMS) strategy. Separate agent structures are utilized to represent brain areas involved in working memory (WM) encoding and reward signal consideration, facilitating the training process. The composite system is embedded in a simulated robot, which is trained to perform in either DMS or DNMS modes. Furthermore, simulated lesion is performed in higher level motor structures resulting in reduced ability of the robot to move in a purposeful mode.

The rest of the article is organized as follows. In Section 2, we present the agent structures employed for the representation of partial brain areas. Then, we discuss the HCCE scheme, which is utilized to design brain-inspired systems consisting of autonomous, still cooperating agents. Experimental results regarding the proposed computational framework are presented in Section 3. In particular, we describe the design of a distributed computational model mimicking aspects of brain functionality, and additionally we evaluate HCCE effectiveness, comparing it with enforced subpopulation co-evolution (Gomez, 2003; Gomez & Miikkulainen, 1999), and with unimodal evolution. Then, a detailed discussion highlights the basic features of the proposed methodology. Finally, conclusions and suggestions for further work are drawn in Section 6.

2 Computational Model

The agent-oriented view is adopted in many computational systems, because it is able to represent the decentralized characteristics of real-life problems, the multiple loci of control, and the multiple perspectives of the problem with respect to partial structures (Jennings, 2001). For a biologically inspired system consisting of agents representing brain areas, the design of partial structures should ideally be based on the natural principles of the central nervous system. Along this line, we implement two different neural network based agents, which are utilized as primitive building blocks for the design of the composite cognitive system. Specifically, we implement: (i) a computational cortical agent to represent brain areas, and (ii) a link agent to support information flow across cortical modules. The computational structures employed in the present study constitute an enhanced version of the agents presented in Maniadakis and Trahanias (2006b). The new agents follow a more flexible formulation, which emphasizes their reusability, offering advanced modeling abilities of central nervous system performance.

2.1 Link Agent

The structure of the link agent is appropriately designed to support connectivity among cortical modules. Using the link agent, any two cortical modules can be connected, formulating any desired networked structure. As a result, link structures can be properly employed to design artificial systems simulating the connectivity of brain areas.

Each link agent is specified by the projecting axons between two cortical agents (Figure 1a). Its formation is based on the representation of cortical

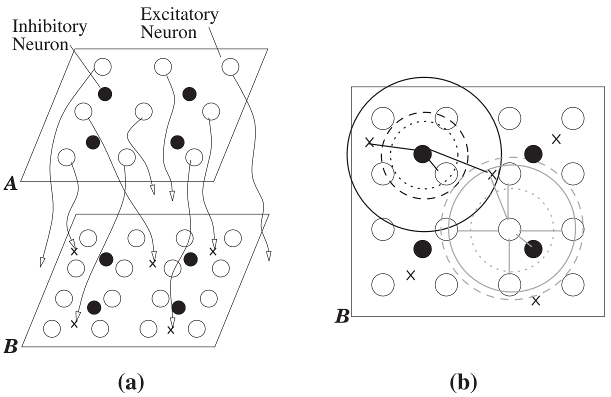


Figure 1 Schematic representation of the computational model. (a) A link agent that supports information flow from cortical agent A to B. (b) The synapse definition in cortical agent B, for one inhibitory (black lines) and one excitatory (gray lines) neuron. Neighborhood radius for afferent axons is illustrated by a circular solid line, for neighboring excitatory neurons by a dashed line, and for neighboring inhibitory neurons by a dotted line.

agents by planes with excitatory and inhibitory neurons (see below). Only excitatory neurons are used as outputs of the projecting cortical agent. The axons are defined by their (x, y) coordinates on the receiving plane. Cortical planes have a predefined dimension and thus projecting axons are deactivated if they exceed the borders of the plane. This is illustrated graphically in Figure 1a, where only the active projections are represented with an " × " on their termination. The proposed link structure facilitates the connectivity of sending and receiving cortical agents supporting the flow of information within the composite system.

2.2 Cortical Agent

Each cortical agent is represented by a rectangular plane. A cortical agent consists of a predefined population of excitatory and inhibitory neurons. Both sets of neurons are uniformly distributed. Thus, an excitatory and an inhibitory grid are defined on the cortical plane. Both types of neurons follow the Wilson–Cowan model with sigmoid activation, similar to Tkaczyk (2001) and Maniadakis and Trahanias (2006b). Let x represent the firing rate of a neuron. It is updated based on the incoming signals, following the equation:

$$\mu\Delta x = -x + S(W_A A + W_E E - W_I I) \qquad (1)$$

where μ presents the membrane time constant, W_A are the synaptic weights of the afferent axon signals, and W_E and W_I are the synaptic weights of neighboring excitatory and inhibitory neurons, respectively. $S(y) = 1/[1 + e^{-\alpha(y-\beta)}]$ is the non-linear sigmoid function, where β and α denote the threshold and the slope, respectively.

On the same plane, the axon terminals from the projecting cortical agents are also located (Figure 1b). Three synapse types specify the connectivity of cortical agents. All neurons receive input information from (i) projecting axons, or (ii) excitatory neighboring neurons, or (iii) inhibitory neighboring neurons. The connectivity of neurons follows the general rule of locality (Redish, Elga, & Touretzky, 1996), and thus synapse formation is based on circular neighborhood measures. A separate radius for each of the three synapse types defines the connectivity of neurons. This is illustrated graphically in Figure 1b, which further explains the case of Figure 1a. All excitatory neurons share common neighborhood measures. The same is also true for all inhibitory neurons.

The performance of cortical agents is adjusted by the experiences of the artificial organism, obtained through environmental interaction, similar to epigenetic[1] learning (Cotterill, 2001). To enforce experience-based subjective learning of robots, each set of synapses is assigned a Hebbian-like biologically plausible learning rule, similar to Floreano and Urzelai (2000). We have implemented a pool of 10 Hebbian-like rules that can be appropriately combined to produce a wide range of functionalities. Learning rules are the same as those presented in Maniadakis and Trahanias (2006b), and thus they are omitted here. Agent plasticity allows synaptic adjustments at run-time based on environmental experience. The most common, but harder to evolve, alternative for genetically encoded synaptic strengths results in a rather unmanageable problem complexity, and at the same time prevents experience-based adjustment.

2.3 Reinforcement Learning

Reinforcement learning models (particularly those based on temporal difference learning) have been very popular in robotic applications in recent years. Despite the effectiveness of reinforcement learning approaches, the biological reliability of this learning scheme has been criticized (Pennartz, 1997; Sporns & Alexander, 2002). These researchers suggested that Hebbian learning mechanisms are able to facilitate training processes based on reinforcement signals. As a result, the Hebbian self-organized dynamics of cortical agents can be exploited to support learning of the artificial organism.

In the current work, we investigate how reinforcement learning capabilities can emerge in the system by properly combining a set of different Hebbian-like learning rules. The idea is based on treating the reward as an ordinary signal, which can be properly given as input in a pre- and a post-synaptic neuron to coordinate their activations. The self-organized learning rule, which adjusts the weight of the synapse connecting the neurons, is then responsible for either strengthening or weakening their connection. As a result, the external reinforcement signal is able to take advantage of the internal plasticity dynamics of the agent, in order to modulate its performance accordingly. Similar computational models exhibiting reinforcement learning characteristics have been also utilized by others (e.g., Blynel & Floreano, 2002).

3 Hierarchical Cooperative Co-evolution

Similar to a phylogenetic process, the specification of parameter values for the agents of a brain-like system is accomplished by using an evolutionary procedure. A preliminary work utilizing ordinary evolutionary computing for the design of simple models corresponding to brain structures appeared in Rolls and Stringer (2000). However, using unimodal evolution, it is not possible to explore effectively partial solutions, which correspond to brain substructures (Potter & De Jong, 2000). Co-evolutionary algorithms have been recently proposed that facilitate exploration in problems consisting of many decomposable subcomponents. They involve two or more co-evolved populations with interactive performance.

Most of the co-evolutionary approaches presented in the literature can be classified as competitive (Olsson, 2001; Rosin & Belew, 1997) or cooperative (Casillas, Cordón, Herrera, & Merelo, 2001; Potter & De Jong, 2000). Competitive approaches are based on an antagonistic scenario, where the success of one population implies the failure of the other. In contrast, cooperative approaches follow a synergistic scenario, where individuals are favored when they successfully cooperate with individuals from the other populations. We note that distinct populations are usually referred to as species in the co-evolutionary literature, and thus this term will be employed henceforth.

The design of biologically inspired cognitive systems fits very well with cooperative co-evolutionary approaches, because separate co-evolved species can be used to perform design decisions for each partial model of a brain area. Specifically, the co-evolutionary design mechanism enforces the co-adaptation of partial components on one another, facilitating their cooperation. Thus, in the following we only consider cooperative co-evolution.

Recently, we have presented a new evolutionary scheme to improve the performance of cooperative co-evolutionary algorithms (Maniadakis, 2006; Maniadakis & Trahanias, 2006b). The main novelty of the proposed scheme lies in that it explicitly addresses the cooperator selection issue. The present work extends this scheme to a hierarchical multilevel architecture (see also Maniadakis & Trahanias, 2007). Our method combines the hierarchical evolutionary approach (Delgado, Zuben, & Gomide, 2004), with the maintenance of successful cooperator assemblies (Moriarty & Miikkulainen, 1997), to develop a powerful co-evolutionary scheme capable of co-evolving a large number of species. We need to note that Pareto evolutionary criteria have been recently used in co-evolutionary approaches as a means to support their effectiveness (De Jong, 2003, 2004; Ficici & Pollack, 2001; Garcia-Pedrajas, Ortiz-Boyer, & Hervas-Martinez, 2004; Iorio & Li, 2004). However, the integration of Pareto issues to hierarchical co-evolutionary schemes is not straightforward. This is because it is not clear how fitness values should be propagated along the co-evolutionary hierarchy, or how effective Pareto fronts should be formulated when partial populations have to satisfy multiple but different (for each population) fitness criteria. At the same time, it seems that Pareto approaches are very time-consuming when a large number of populations need to be co-evolved, because many individual combinations (the product of Pareto fronts) should be repeatedly tested. Overall, we would like to note that the issue of Pareto optimality in hierarchical co-evolutionary schemes is out of the scope of the present work.

Two different types of species (populations) are utilized to support the co-evolutionary process, encod-

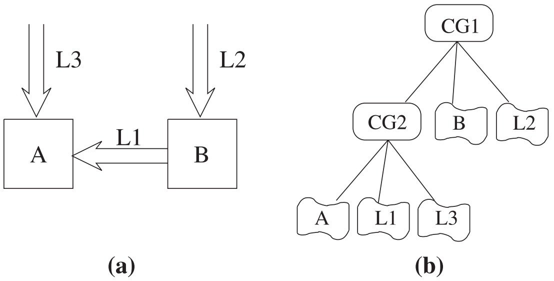


Figure 2 Hierarchical cooperative co-evolutionary design of agents. (a) Schematic representation of a hypothetical connectivity of agents. (b) Representation of the hierarchical co-evolutionary scheme utilized to evolve partial structures. CGs are illustrated with oval boxes, while PSs are represented by free shapes.

ing the configurations of either a primitive agent structure (PS) or a co-evolved agent group (CG). Particularly, PS species specify partial elements of the model, encoding the exact structure of either cortical or link agents. A CG consists of groups of PSs with common objectives. Thus, CGs specify configurations of partial solutions by encoding individual assemblies of cortical and link agents. The evolution of CG modulates partly the evolutionary process of its lower level PS species, enforcing their cooperative performance. A CG can also be a member of another CG. Consequently, several CGs can be organized hierarchically in a tree-like architecture, with the higher levels enforcing the cooperation of the lower ones.

The HCCE-based design mechanism of brain-inspired systems is demonstrated by means of an example (Figure 2). We assume the existence of two cortical agents connected by three link agents representing their afferent and efferent projections (Figure 2a). This assumption is typical for mammalian central nervous system organization (e.g., separate brain areas serve visual or motor competencies, which further cooperate exchanging information, in order to accomplish advanced real-life behaviors). One hypothetical HCCE process employed to specify agent structures is illustrated in Figure 2b.

Similar to Delgado et al. (2004) and Maniadakis and Trahanias (2006b) all individuals in all species are assigned an identification number, which is preserved during the co-evolutionary process. The identification number is employed to form individual assemblies among different species. Specifically, each variable in the genome of a CG is joined with one lower level CG or PS species. The value of that variable can be any identification number of the individuals from the species it is joined with. PSs encode the structure of either cortical or link agents. The details of the encoding are very similar to those presented in Maniadakis and Trahanias (2006b), and thus they are omitted here. CGs enforce cooperation of PS structures by selecting the most appropriate individuals to cooperate among species. Additionally, a new genetic operator, called replication (Maniadakis & Trahanias, 2006b), exploits the most successful individuals in each partial species, being used as a basis for further exploration.

In order to test the performance of a complete problem solution, the population at the highest level is sequentially accessed. The values of CG individuals at various levels are used as guides to select cooperators among PS species. Then, PS individuals are decoded to specify the structure of cortical and link agents, and the performance of the proposed overall solution is tested on the desired task. The formulation of complete problem solutions is demonstrated in Figure 3, presenting a snapshot of the exemplar HCCE process discussed above.

The proposed hierarchical scheme supports the simulation of lesion conditions (Maniadakis & Trahanias, 2005b, 2005c), which is typical for biological experiments related to the performance of mammalian central nervous system. Specifically, the deactivation of a CG, together with the PS structures corresponding to its lower level species, simulates lesion of the respective brain areas. As a result, all necessary lesion conditions can be explored, during the HCCE-based design procedure, highlighting the role of partial structures in the composite model.

Furthermore, even if the majority of existing cooperative co-evolutionary methods assume that all species share a common fitness function (Casillas et al., 2001; Krawiec & Bhanu, 2003; Wiegand, Liles, & De Jong, 2001), our method allows the employment of separate fitness measures for different species. This matches adequately the distributed agent-based modeling of brain areas, because different objectives can be defined for each partial structure preserving their autonomy. The same feature additionally supports the modeling of biological lesion data, because properly formulated fitness functions can be utilized to specify the desired pre- and post-lesion performance of the model.

For each species s, a fitness function f_s is designed to drive its evolution. All PS species under a CG share

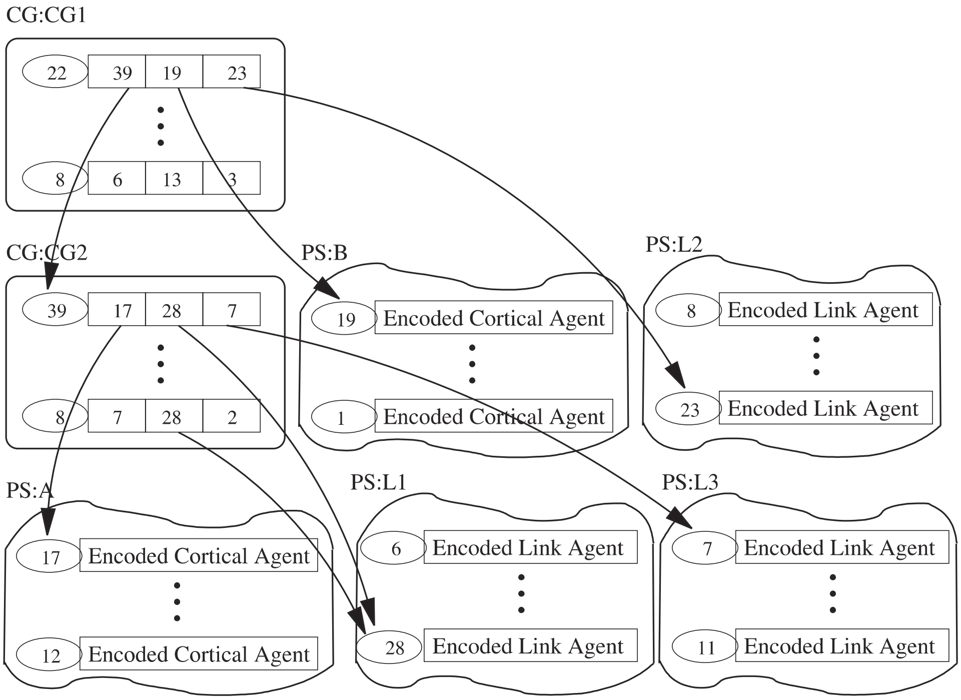


Figure 3 An overview of the hierarchical co-evolutionary scheme, with CG species tuning the evolutionary processes of PS species. Identification numbers are represented with an oval.

a common f_s. The fitness function is formulated to evaluate the performance of the model in different conditions, which corresponds to the pre- and post-lesion state of the model. Specifically, a partial fitness function $f_{s,t}$ evaluates the ability of an individual to serve task t, while the overall fitness function is estimated by

$$f_s = \prod_t f_{s,\,t}. \tag{2}$$

Furthermore, the cooperator selection process at the higher levels of hierarchical co-evolution will probably select an individual to participate in many assemblies (e.g., the case of individual 28 of PS species L1, in Figure 3). Let us assume that an individual participates in K assemblies, which means that it will have K fitness values $f_{s,t}$ regarding its ability to serve task t. Then, similarly to most existing co-evolutionary approaches, the individual will be assigned the maximum of the fitness values achieved by all the solutions formed with its membership:

$$f_{s,\,t} = \max_k\{f^k_{s,\,t}\}, \tag{3}$$

where $f^k_{s,\,t}$ is the fitness value of the kth solution formed with the membership of the individual under discussion. This value represents the ability of the individual to support the accomplishment of the t-th task.

The above equations describe the fitness assignment in each species of the hierarchical co-evolutionary process. Just after the testing of cooperator assemblies and the assignment of their fitness values, an evolutionary step is performed independently on each species, formulating the new generation of its individuals. First, individuals of the species are sorted according to their fitness values. Then, a replication operator reduces the very large number of cooperations for individuals. Next, a predefined percentage of individuals are probabilistically crossed over. An individual selects its mate from the whole population, based on their accumulative probabilities. Finally, mutation is performed in a small percentage of the resulting population. This process is repeated for a predefined number of evolutionary epochs, driving each species to the accomplishment of its own objectives and additionally enforcing agents to accomplish successful cooperative performance.

4 Results

The effectiveness of the proposed approach is illustrated by designing a brain-inspired cognitive system that models primary motor–premotor–prefrontal–parietal cortex interaction, emphasizing on Working Memory (WM) usage.

Existing computational models of WM demonstrate the persistence of neural activity patterns by means of recurrent circuits (Compte, Brunei, Goldman-Rakic, & Wang, 2000; Iida & Tanaka, 2002). However, these models are not operative, in the sense that they are not linked to other structures to affect their performance. Computational models aiming at the accomplishment of WM guided tasks have been also proposed in the literature (Moody, Wise, Pellegrino, & Zipser, 1998), which, however, employ a single artificial neural network structure, without addressing the performance of partial brain areas.

The present model highlights the roles of the cortical areas involved in the training of animals by means of external reinforcement stimuli, in order to accomplish Delayed Response (DR) tasks. The model is embedded in a simulated robot furnishing it with cognitive abilities, and additionally proving the development of WM-like activation in the system. We employ a two-wheeled simulated robotic platform equipped with eight distance, light, and reward sensors, uniformly distributed in a circular manner around the robot, to support environmental interaction.

4.1 Brain Areas and Training Strategies

Several biological experiments have revealed the behavioral organization of the areas involved in WM, as illustrated in Figure 4. These studies are usually based on DR tasks, which need to retain memory related to a sample cue for a brief period, in order to decide upon future behavioral response (e.g., Ragozzino & Kesner, 2001). The primary motor cortex (M1) encodes primitive motor commands, which are expressed to actions by means of the spinal cord (SC). The prefrontal cortex (PFC) is reciprocally connected to the posterior parietal cortex (PPC) encoding WM (Compte et al., 2000). Prefrontal activity is projected on the premotor cortex (PM) to formulate the orders that modulate M1 performance according to the higher level plans, developing purposeful motion (Fuster, 2000). Overall, separate brain structures have to coordinate their activities, in order to manipulate successfully WM.

Following a reinforcement-based training process, animals are able to learn a variety of motion strategies (e.g., DMS or DNMS), depending on the rewards provided. Prefrontal and premotor areas receive information relevant to external environmental reinforcement, modulating the plans of future motion in the desired way (Murray, Bussey, & Wise, 2000). Moreover, several experiments highlight the performance of the brain in lesion conditions. Specifically, lesion at higher level structures affects the learning of the organism (Murray et al., 2000), resulting in significantly reduced levels of purposeful motion and planning ability (Granon, Vidal, Thinus-Blanc, Changeux, & Poucet, 1994; Ragozzino & Kesner, 2001).

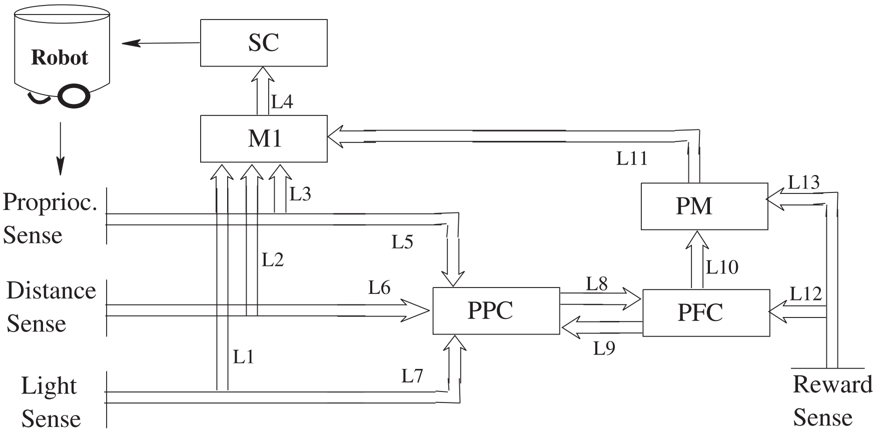


Figure 4 A schematic overview of the composite model. Cortical agents are illustrated with blocks, while link agents are illustrated with double arrows.

The experimental process followed in the present computational study aims at replicating (i) the training process of animals for the accomplishment of DR tasks in a T-maze, and (ii) the biological findings related to the effects of lesion at higher level motor structures. Specifically, the composite computational model should be trainable by an external factor, adopting either a DMS or a DNMS strategy. This is similar to the pre-lesion performance of animals (Ragozzino & Kesner, 2001). The accomplishment of these DR tasks is supported by the development of WM-like activation in PPC–PFC, which are the brain areas most closely linked to WM (Compte et al., 2000). Furthermore, when lesion occurs on the higher level structures, the robot should be able to drive but only in a purposeless mode, simulating reduced planning ability of animals (Granon et al., 1994; Pontecorvo, Sahgal, & Steckler, 1996). Overall, three tasks are designed to demonstrate the effectiveness of the computational procedure, highlighting the distinct role of the agents in the model, and additionally their successful cooperation in the composite system. These are described in the following sections. The first two tasks are related to teaching the robot to perform the right DR, adopting either a DMS or a DNMS strategy (Pontecorvo et al., 1996).

4.1.1 DMS Strategy Let us consider first how the robot is trained to adopt the DMS strategy. The process is separated into several trials. Each trial includes two sample–response pairs, testing the memorization of two different sample cues by the robot, and the selection of the appropriate DR. In the sample phase of the first pair, the robot starts at the top of the T-maze and is driven to the left, while in the sample phase of the second pair, it is driven to the right. This is achieved by utilizing one extra properly placed wall that enforces robot turning to a pre-specified direction (see “sample motion” column in Figures 6 and 7). During the response phase of each trial, the intermediate wall is removed and the robot is free to make its own turning choice (see “response motion” column in Figures 6 and 7).

In order to make a correct DR, the robot should first develop WM-like activity, encoding the side of turning during the sample phase. Similar to the biological prototype, WM is encoded by PPC–PFC interactions. Two different states “a” and “b” are defined, associated with the two possible turning directions. For each state, separate activation averages, p_l, are computed, with l identifying excitatory neurons. The formation of WM related to the side of sample turning is evaluated by considering activation in PFC–PPC structures:

$$E^j_{\mathrm{wm}} = \left(\frac{v_{\mathrm{a}}}{m_{\mathrm{a}}} + \frac{v_{\mathrm{b}}}{m_b}\right) \times \min\left\{ \sum_{p_l^{\mathrm{a}} > p_l^{\mathrm{b}}}^{l} (p_l^{\mathrm{a}} - p_l^{\mathrm{b}}),\ \sum_{p_l^{\mathrm{b}} > p_l^{\mathrm{a}}}^{l} (p_l^{\mathrm{b}} - p_l^{\mathrm{a}}) \right\} \tag{4}$$

Here, $j \in$ {PPC, PFC}, and m_{a}, v_{a}, m_{b}, and v_{b} are the means and variances of average activation at states “a” and “b.” The first term enforces consistent activation, while the second supports the development of separate activation patterns for each state “a” and “b.” Thus, $E^{\mathrm{PPC}}_{\mathrm{wm}}$ and $E^{\mathrm{PPC}}_{\mathrm{wm}}$ evaluate the development of distinct memories relative to the locations “a” and “b” in PPC and PFC.

After encoding the distinct sample cues in WM, the robot should learn to manipulate it properly, in order to accomplish the DR tasks. In particular, just after the sample phase, the robot is artificially transferred to the initial starting position for the response phase. The robot drives freely to the end of the corridor making its own left and right turn choices. An external human factor specifies a reward area in the left or right side of the T-maze, depending on the initial sample cue. If the robot drives to this area, it receives a reinforcement signal, simulating the positive reward of animals with food or water. In the DMS training process, the response is considered correct if the robot turns to the same side with the sample turning. Thus, the robot has to consider the presence or non-presence of the reinforcement signal in order to realize what is the correct strategy it should adopt in the future trials.

The success of DMS training is tested for T trials. We note that for each trial two different responses are considered, each related to either the left or the right sample motion. The success of the training process is evaluated computationally by the total positive reinforcement that the robot has received. This is described computationally by

$$E_{\mathrm{tr}} = \left(\sum_{T,\,\mathrm{left}} \sum_{M} r\right)\left(\sum_{T,\,\mathrm{right}} \sum_{M} r\right)\left(1 - \sqrt{\frac{B}{2TM}}\right)^3, \tag{5}$$

where M are the number of simulation steps of the response phases for each of the T trials, r is the maximum instant activation of the reward sensors in one simulation step, and B is the total number of robot bumps. The first term seeks for maximum reward stimuli when the correct response of the robot is considered the left side, while the second seeks for maximum reward when the correct response is the right side. The last term minimizes the number of robot bumps on the walls. The higher the reward the robot received, the more successful the training process. In other words, high values of the E_{tr} measure indicate successful cooperation among partial modules related to WM encoding, learning, and motion expression.

4.1.2 DNMS Strategy

Having described the procedure of training the robot to adopt the DMS strategy, we turn to the DNMS task. Specifically, the computational structure is re-initialized, and we test if the same cognitive system is able to adopt successfully the DNMS response strategy, given a different set of reward stimuli.

The experimental process is again separated in T trials. The process is very similar to that described for DMS. Each trial includes two sample–response pairs, but this time the reward stimulus is located to the side that is opposite to the sample turning (i.e., if the extra wall forces the robot to turn left, then, during the response phase, the reward is located to the right, and vice versa). The robot should again memorize the side of sample turning in different activation patterns of WM, and then consider the location of the reinforcement signals in order to adopt the correct sample–response mapping. The measures evaluating the success of the DNMS training process and the adoption of the correct response strategy by the robot are the same as those described in Equations 4 and 5.

Overall, we use two sets of evaluation measures $\{E^{j}_{\mathrm{wm,\,dms}}, E_{\mathrm{tr,\,dms}}\}$ and $\{E^{j}_{\mathrm{wm,\,dnms}}, E_{\mathrm{tr,\,dnms}}\}$ related to WM encoding and correct delayed response, for either the DMS or the DNMS task. Similar to the training process of animals, the human designer enforces the robot to develop the correct strategy, by properly locating the reward signals. For both tasks, several trials (each consisting of two sample–response pairs) are performed, and the robot should progressively realize what is the correct strategy. In the first trials, the robot does not know if the testing procedure is related to the DMS task or the DNMS task, and it performs a random turning responses. In order to discover what is the correct strategy, the composite cognitive system has to consider (i) what was the sample motion encoded in WM, compared to the expressed DR, and (ii) if it has received reward stimuli or not. After some explorative trials, the robot should be able to successfully conclude which response strategy it should adopt.

4.1.3 Lesion Effect: Wall Avoidance Strategy

The third task is relevant to the lesion of higher level motor structures, implying that the robot can no longer be taught (Murray et al., 2000). This task accounts for primitive motion abilities without purposeful planning. For mobile robots, a task with the above characteristics is wall avoidance navigation. The successful accomplishment of the task is evaluated by the function:

$$E_{\mathrm{wa}} = \left[\sum_{M}(sl + sr - 1)(1.0 - p^{2})\right] \times \left(1 - \frac{2}{M}\left|\sum_{M}\frac{sl - sr}{sl\,sr}\right|\right)^{3}\left(1 - 2\sqrt{\frac{B}{M}}\right)^{3}. \tag{6}$$

Here, we assume that the robot is tested for M steps, sl and sr are the instant speeds of the left and right wheels, respectively, p is the maximum instant activation of distance sensors, and B is the total number of robot bumps. The first term seeks for forward movement far from the walls, the second supports straight movement without unreasonable spinning, and the last term minimizes the number of robot bumps on the walls.

4.2 Modeling

The WM model is implemented following the agent-based co-evolutionary computational framework, demonstrating the ability of the latter to design complex systems consisting of autonomous yet cooperative modules.

4.2.1 Computational Representation

In order to implement the desired model illustrated in Figure 4, the cortical and link agents described above are utilized to represent the involved brain areas. Particu-

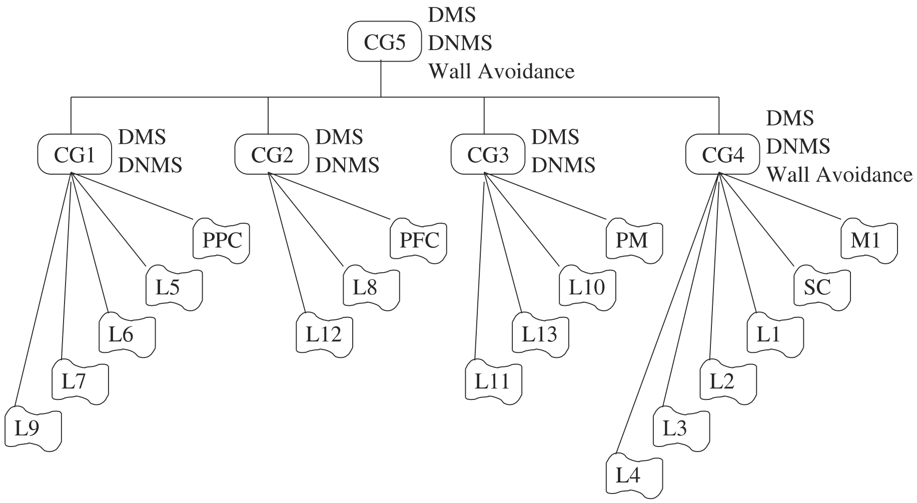


Figure 5 A schematic overview of the HCCE process employed to design the composite computational model. CGs are illustrated with oval boxes, while PSs are represented by free shapes.

larly, 18 components are used (five cortical agents and 13 link agents), while the detailed structure of the model is specified by 332 parameters in total. Then, an HCCE-based design process is employed to make a parametric specification of the model as described below.

4.2.2 Co-evolutionary Design Procedure The HCCE scheme that assembles the distributed model is shown in Figure 5. According to the experimental scenario followed in the present study, each agent needs to serve more than one task. The tasks served by each group of agents are illustrated in the same figure, at the right side of each CG. Specifically, the structures under *CG*1, *CG*2, and *CG*3 are mostly related to the DR training process for the accomplishment of either the DMS or the DNMS response strategy. The structures under *CG*4 need also to serve these tasks, but additionally, in the case of lesion at higher level structures, they should also be able to drive the robot in a wall avoidance mode. Finally, the top level *CG*5 enforces cooperation within partial configurations, aiming to support the accomplishment of all three tasks.

The testing phase for the individuals of the co-evolutionary scheme proceeds as follows. The individuals of the top-level species are sequentially accessed. Each individual of *CG*5 guides cooperator selection among its lower level CG and PS species. Individuals of PS species are decoded to detailed agent structures. First, we test the ability of the composite model to adopt the DMS strategy, and its components are evaluated with respect to their support for this task. Then, the model is re-initialized, and tested on learning to adopt the DNMS strategy. Next, a simulated lesion is performed, by deactivating the agents under *CG*2 and *CG*3. The remaining structures are tested on wall avoidance navigation, representing purposeless robot motion. The fitness functions that guide the evolution of species are designed according to the target tasks of each CG. Individuals are assigned a combination of evaluation indices for the accomplishment of tasks where the composite model is performing, and the accomplishment of tasks with performance of the eliminated model. Note that all PSs share the same fitness functions with their higher level CG.

According to the employed scenario, the agent structures grouped under *CG*1 serve the success on the tasks DMS and DNMS. Thus, the fitness function employed for the evolution of *CG*1 and its lower level species is based on measures evaluating the success of the respective tasks. Following the formulation introduced in Equations 2 and 3:

$$f_{\mathrm{CG1}} = f_{\mathrm{CG1,\,t1}} f_{\mathrm{CG1,\,t2}} \text{ with}$$

$$f^{k}_{\mathrm{CG1,?\,t1}} = E^{\mathrm{PPC}}_{\mathrm{wm,\,dms}}(1 + E_{\mathrm{tr,\,dms}}),$$

$$f^{k}_{\mathrm{CG2,?\,t2}} = E^{\mathrm{PFC}}_{\mathrm{wm,\,dnms}}(1 + E_{\mathrm{tr,\,dnms}}) \qquad (7)$$

where k represents each membership of an individual in a proposed solution. We note that the evaluation measures employed for the evolution of CG1 are formulated in a subjective manner, examining those aspects of the tasks that are relevant to PPC. Particularly, the partial fitness functions of f_{CG1} aim at the development of WM patterns (i.e., $E^{\mathrm{PPC}}_{\mathrm{wm,\,dms}}$, $E^{\mathrm{PPC}}_{\mathrm{wm,\,dnms}}$) and their proper formulation to be easily manipulated for expressing correct DRs (i.e., $E_{\mathrm{tr,\,dms}}$, $E_{\mathrm{tr,\,dnms}}$).

The agents grouped under *CG*2 also serve the success on tasks DMS and DNMS. Thus, according to the formulation of Equations 2 and 3, the fitness function employed for the evolution of *CG*2 is

$$f_{\mathrm{CG2}} = f_{\mathrm{CG2,\,t1}} f_{\mathrm{CG2,\,t2}} \text{ with}$$

$$f^{k}_{\mathrm{CG2,?\,t1}} = E^{\mathrm{PFC}}_{\mathrm{wm,\,dms}}\,(1 + E_{\mathrm{tr,\,dms}}),$$

$$f^{k}_{\mathrm{CG2,?\,t2}} = E^{\mathrm{PFC}}_{\mathrm{wm,\,dnms}}\,(1 + E_{\mathrm{tr,\,dnms}}) \tag{8}$$

where k represents each membership of an individual in a proposed solution. We note again the subjective evaluation of the tasks, which now aims at the development of WM patterns at PFC (i.e., $E^{\mathrm{PFC}}_{\mathrm{wm,\,dms}}$, $E^{\mathrm{PFC}}_{\mathrm{wm,\,dnms}}$), and their successful manipulation (i.e., $E_{\mathrm{tr,\,dms}}$, $E_{\mathrm{tr,\,dnms}}$).

The agents grouped under *CG*3 also serve the success on DMS and DNMS tasks. However, this time, only the manipulation of WM is considered, as the premotor area is not involved in WM development. Thus, the fitness function employed for the evolution of *CG*3 is:

$$f_{\mathrm{CG3}} = f_{\mathrm{CG3,\,t1}} f_{\mathrm{CG3,\,t2}} \text{ with}$$

$$f^{k}_{\mathrm{CG3,?\,t1}} = E_{\mathrm{tr,\,dms}}, \quad f^{k}_{\mathrm{CG3,?\,t2}} = E_{\mathrm{tr,\,dnms}}, \tag{9}$$

where k represents each membership of an individual in a proposed solution.

Furthermore, according to the lesion scenario, the agent structures grouped under *CG*4 also serve the success on the tasks DMS, DNMS, and wall avoidance. Thus, the fitness function employed for the evolution of *CG*4 and its lower level species is based on the measures evaluating the success of the respective tasks. Following the formulation introduced in Equations 2 and 3:

$$f_{\mathrm{CG4}} = f_{\mathrm{CG4,\,t1}} f_{\mathrm{CG4,\,t2}} f_{\mathrm{CG4,\,t3}} \text{ with}$$

$$f^{k}_{\mathrm{CG4,?\,t1}} = E_{\mathrm{tr,\,dms}}, \quad f^{k}_{\mathrm{CG4,?\,t2}} = E_{\mathrm{tr,\,dnms}},$$

$$f^{k}_{\mathrm{CG4,?\,t3}} = E_{\mathrm{wa}}, \tag{10}$$

where k is as above.

Finally, the top level CG enforces the integration of partial configurations in a composite model, aiming at the successful cooperation of substructures in order to facilitate the accomplishment of all three tasks. The fitness function employed for the evolution of *CG*5 supports successful DMS training, successful DNMS training, and wall avoidance. It is defined accordingly, following the formulation introduced in Equations 2 and 3, by:

$$f_{\mathrm{CG5}} = f_{\mathrm{CG5,\,t1}} f_{\mathrm{CG5,\,t2}} f_{\mathrm{CG5,\,t3}} \text{ with}$$

$$f^{k}_{\mathrm{CG5,?\,t1}} = E^{\mathrm{PPC}}_{\mathrm{wm,\,dms}}\, E^{\mathrm{PFC}}_{\mathrm{wm,\,dms}}(1 + E_{\mathrm{tr,\,dms}}),$$

$$f^{k}_{\mathrm{CG5,?\,t2}} = E^{\mathrm{PPC}}_{\mathrm{wm,\,dnms}} E^{\mathrm{PFC}}_{\mathrm{wm,\,dnms}}(1 + E_{\mathrm{tr,\,dnms}}),$$

$$f^{k}_{\mathrm{CG5,?\,t3}} = E_{\mathrm{wa}}, \tag{11}$$

where k is as above.

The hierarchical co-evolutionary process described above employed populations of 200 individuals for all PS species, 300 individuals for *CG*1, *CG*2, *CG*3, and *CG*4, and 400 individuals for *CG*5. Additionally, an elitist evolutionary strategy was followed in each evolutionary step, with the 10 best individuals of each species copied unchanged in the respective new generation, supporting the robustness of the evolutionary process.

4.2.3 Obtained Model: Comments The HCCE scheme discussed above has been evolved for 200 epochs. The process converged successfully, formulating a complex model where components with different objectives (e.g., those under *CG*1 and those under *CG*4) cooperate successfully. In accordance with the design scenario of our study, the very same model undergoes two different training procedures, adopting either the DMS or the DNMS response strategy. Robot performances during both training procedures are illustrated in Figures 6 and 7. In both cases, the response of the robot in the first explorative trial is incorrect. However, in

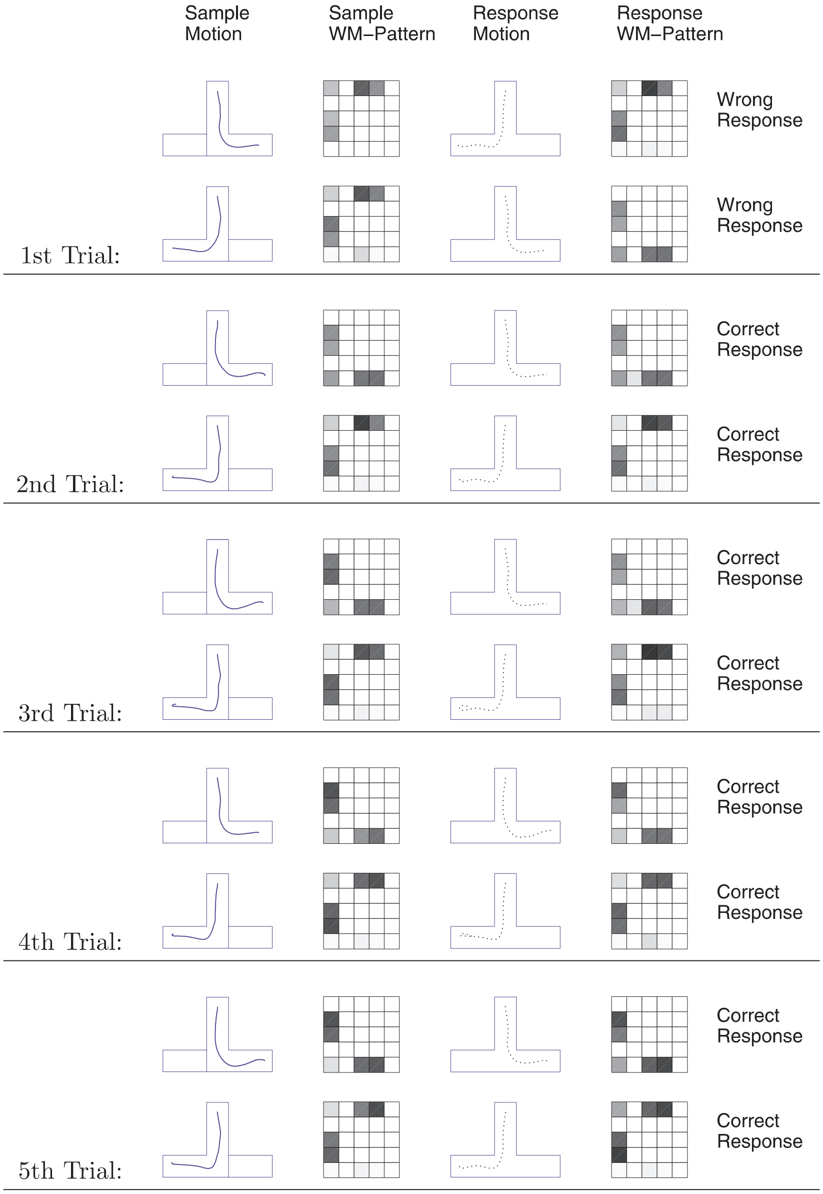


Figure 6 The process of training the robot to adopt the DMS strategy. Each trial (separated by a horizontal line) involves two sample–response pairs. For each pair, the first two columns demonstrate sample robot motion (the starting position is at the top of the corridor), and the WM activation pattern developed in PFC. The third and fourth columns demonstrate robot response and the preserved WM pattern in PFC, respectively. The fifth column comments on the success of the response.

the second trial the WM activation pattern in PFC successfully matches the side of robot turning and the desired response rule, gaining positive reinforcements. Thus, after clarifying what is the correct response strategy, it is also adopted for all the remaining trials.

Additionally, we investigated which strategy is followed by the robot when no reinforcement signals are provided in the environment. We found that neither the DMS nor the DNMS strategy has been followed. Still, in the majority of trials, during the response phase the robot shows a tendency to make right side turnings (only three out of 10 turnings were to the left side). This fact shows that reinforcement signals are important for effectively modulating the

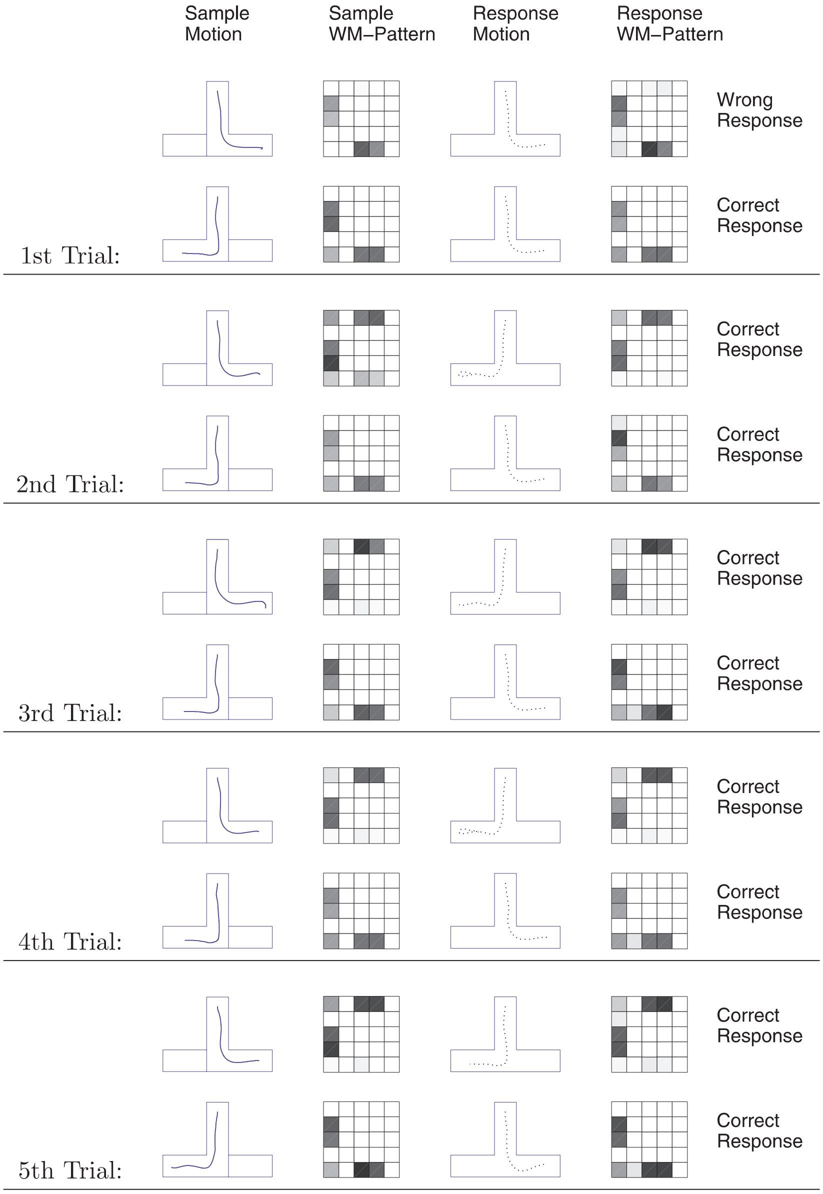


Figure 7 The process of training the robot to adopt the DNMS strategy. Each trial (separated by a horizontal line) involves two sample–response pairs. For each pair, the first two columns demonstrate sample robot motion (the starting position is at the top of the corridor), and the WM activation pattern developed in PFC. The third and fourth columns demonstrate robot response and the preserved WM pattern in PFC, respectively. The fifth column comments on the success of the response.

performance of the robot. The computational system is designed with the capability of adopting either one behavior or the other, but this does not occur unless reinforcement signals specify this choice. This is similar to recurrent neural networks with parametric bias (RNNPBs) employing a single artificial neural network to store/recall a set of behaviors indicated by different bias values (Ito, Noda, Hoshino, & Tani, 2006). In a comparable way, for our model external reinforcement signals bias the selection of the adopted strategy and the behavior expressed by the robot. However, in contrast to the evolutionary design followed in our study, RNNPBs employ back propagation through time to accomplish parametric specification of the network.

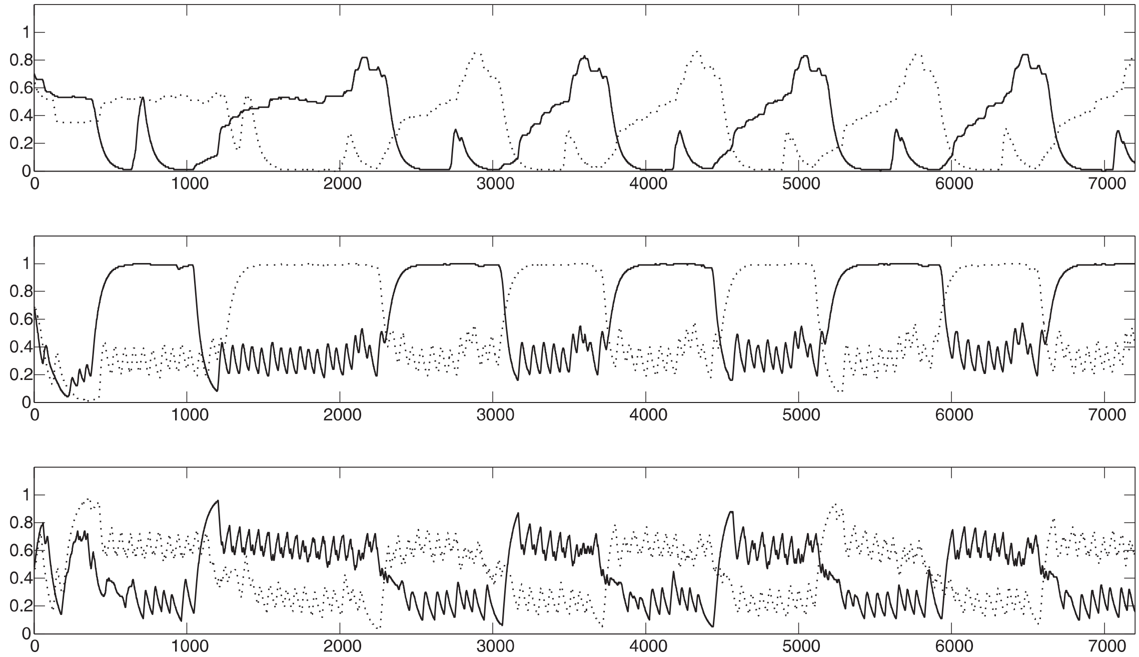


Figure 8 The adjustment of synaptic weights at PFC. Each plot corresponds to a different synapse. The solid line demonstrates weight values during five trials of the DMS task, while the dashed line demonstrates the corresponding weight values during five trials of the DNMS task.

Trying to gain more insight into the internal dynamics of the model, we found that the adopted strategy is encoded in a temporal pattern of PFC synapse adaptation. In particular, Figure 8 demonstrates the weight change of three different PFC synapses during training for adopting DMS and DNMS strategies. Evidently, in the first trial, where the model still explores the desired strategy, there is no clear weight pattern formulated. Afterwards, external reinforcement signals determine what is the desired strategy, and then distinct temporal weight patterns are formulated for each strategy. We note that in each trial, the robot is able to consider the presence or non-presence of the reward stimulus for about 30 simulation steps. Thus, PFC synaptic weights can be properly adapted to the correct motion strategy. In contrast to the above, PPC and PM synapses keep the same values during accomplishing both tasks. Thus, PFC is the only module with dynamic adjustment of synaptic weights, which implies that PFC is responsible for choosing between the DMS or DNMS strategies. This is explained by the fact that PFC is provided access to both the patterns of memorized information and the external reward signals. As a result, it is responsible for combining different types of information, formulating sample–response rules that correctly describe the adopted strategy.

Additionally, Figure 9 demonstrates intermediate snapshots of PPC, PFC, and PM activations during the last (fifth) DMS trial. The same is also demonstrated in Figure 10 for the last DNMS trial. In the initial steps of the sample phase, the robot navigates to the corridor without knowing the side of enforced turning, which therefore means that there is no clear WM pattern formulated yet. Just after the robot turns to the left or right side, the information that should be memorized is clarified, and distinct WM activation patterns are created. These patterns are also preserved during the response phase, driving the robot according to the response rule of the selected DMS or DNMS strategy. Additionally, as expected (because of the employed fitness functions) distinct activation patterns are developed in PPC. Interestingly enough, the PM module also develops distinct activation patterns. Even if PM patterns are less distinctive, they remain strongly correlated to PFC activation and robot response. This is because in order to have WM (located in PFC) driving the robot in two different directions, information has to pass through PM.

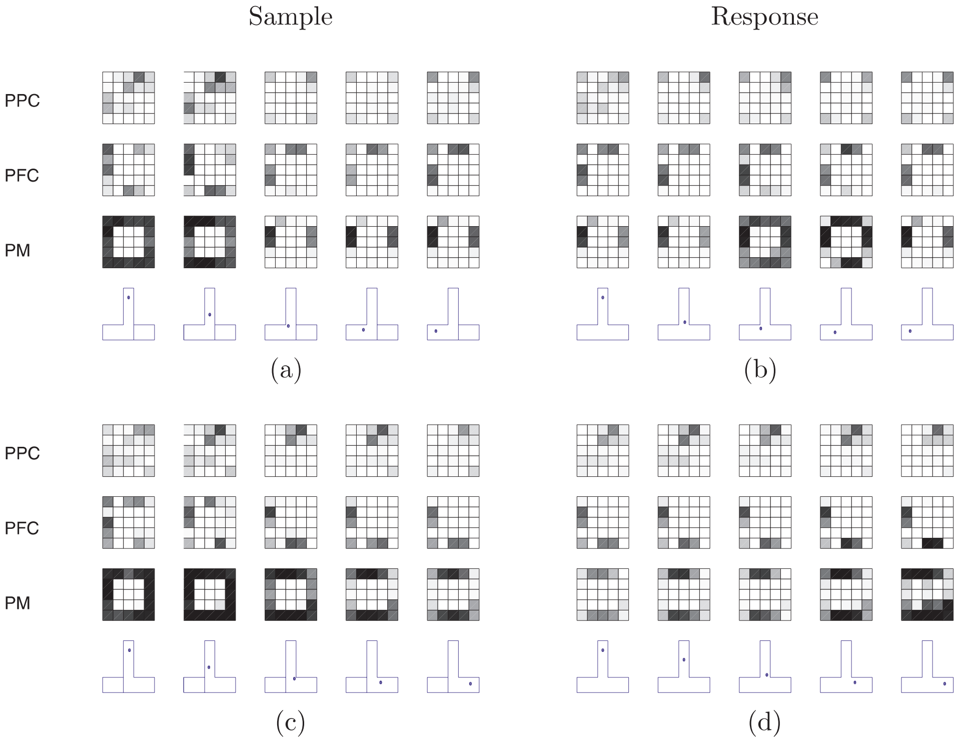


Figure 9 The activation of cortical agents representing PPC, PFC, and PM, during robot performance according to the DMS strategy. The corresponding location of the robot in the environment is demonstrated with a small circle. Each line (a)–(b) or (c)–(d) corresponds to a different sample–response pair. (a) and (c) demonstrate neural activation in the memorization phase, while (b) and (d) demonstrate neural activation in the response phase.

In the case of lesion, the components that are responsible for learning are deactivated, and thus the robot can no longer be taught. However, the system is not completely collapsed, but remains functional at a certain reduced level of performance. In particular, the robot is still able to move, but making nearly random turn choices at the end of the corridor, guided by the wall avoidance strategy (Figure 11).

We have also performed experiments trying to design the composite model without employing post-lesion relevant fitness functions. These design procedures have been significantly less likely to be successful, because during navigation, the robot very often bumps on the walls. Furthermore, we have investigated the performance of these models under simulated lesion conditions. We found that in the post-lesion case, the model was completely collapsed, and the robot was not able to move at all. In other words, when no post-lesion relevant fitness function is employed, the model seems to work in a holistic mode where all modules try to solve the same task without adopting specialized roles.

We have also searched for a possible adaptation of M1 dynamics when the model shifts from pre- to post-lesion operation. Such adaptation is also known to occur in real brains, and it is expected to occur in our model, as a result of the available self-organization dynamics of the employed cortical agents. We found that M1 internal synapses, which direct information from excitatory neurons to inhibitory neurons, have significantly higher values in the pre-lesion case, compared to post-lesion case. For example, Figure 12 demonstrates the weight value of one such synapse for pre- and post-lesion operations. This adaptation is necessary because M1 accepts significantly different amounts of information in pre- and post-lesion cases, and therefore it has to self-organize in order to be adequately functional in both. Specifically, in the pre-lesion case where all components of the model are active, there is a larger number of axon projections to M1, which is significantly reduced after lesion. The enhancement of excitatory-to-inhibitory synapses in the pre-lesion case assists activation of inhibitory neurons, which in

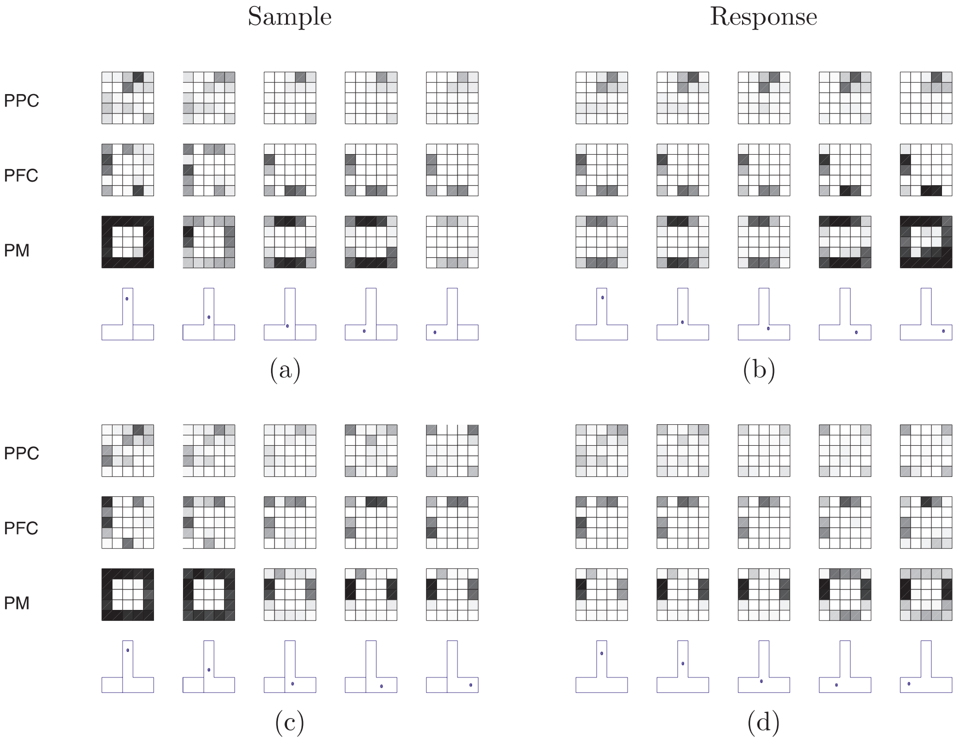


Figure 10 The activation of cortical agents representing PPC, PFC, and PM, during robot performance according to the DNMS strategy. The corresponding location of the robot in the environment is demonstrated with a small circle. Each line (a)–(b) or (c)–(d) corresponds to a different sample–response pair. (a) and (c) demonstrate neural activation in memorization phase, while (b) and (d) demonstrate neural activation in the response phase.

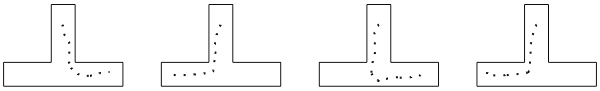
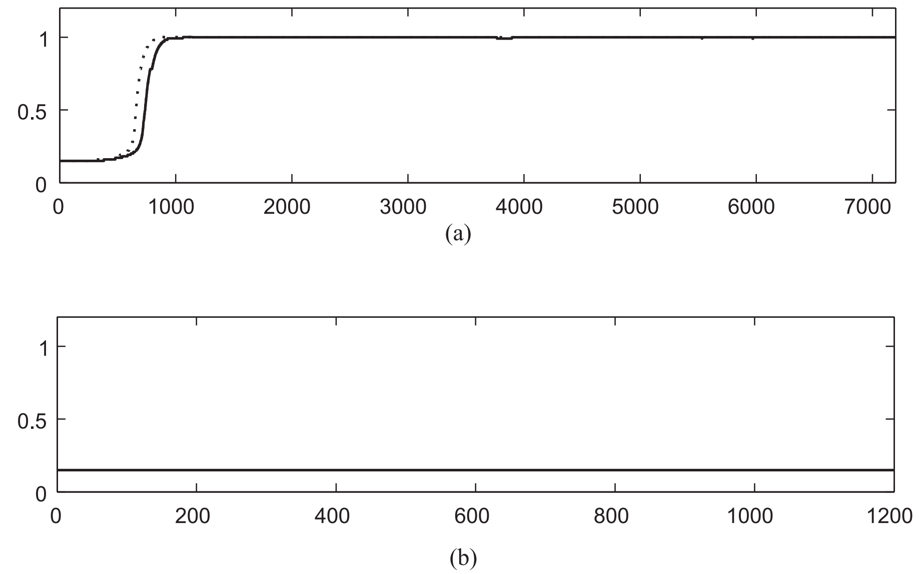


Figure 11 Sample results of robot performance driven by M1–SC. The robot moves in a wall avoidance mode, without following a purposeful motion strategy.

turn reduces the activity of excitatory neurons. Thus, when the whole model is active, increased inhibitory activation reduces the overall M1 activation to meaningful levels, avoiding overflow effects.

The adaptation of M1 internal dynamics discussed above clearly distinguishes our approach from hierarchical reinforcement learning (HRL) methods. This is because typically in HRL the components of the model operate in different time-scales, with the low-level components emitting static primitive actions, while higher-level components try to solve more complex tasks by sequentially invoking lower-level ones (Barto & Mahadevan, 2003; Makar, Mahadevan, & Ghavamzadeh, 2001)1. This pure synthesis of complex behaviors by sequential recall of simpler actions is in contrast to our work where higher-level behaviors modulate the dynamics of lower-level components, being adequately functional in the new operating conditions, accomplishing the emergence of more complex behavioral patterns.

Overall, the brain-inspired computational model implemented following the agent-based co-evolutionary framework replicates successfully pre- and post-lesion rat performance in DR tasks. This is achieved by means of the powerful HCCE design mechanism, which supports the implementation of distributed biologically inspired cognitive systems, considering both the individual and cooperative characteristics of substructures. Additionally, partial structures exploit their internal self-organization dynamics adapting to one another, in order to facilitate successful operation of the composite system.



(a)

(b)

Figure 12 The adjustment of an excitatory-to-inhibitory synapse in M1, during (a) pre-lesion operation and (b) post-lesion operation of the model. The solid line in (a) corresponds to the weight adjustment during DMS, while the dashed line corresponds to the weight adjustment during DNMS. In both tasks, the weight is adjusted to its maximum value. However, (b) shows no enhancement of this synapse during post-lesion operation of the model.

4.3 Comparing HCCE, Enforced Subpopulation, and Unimodal Evolution

In the set of experiments described below, we utilize as a test-bed the problem discussed above, in order to investigate the suitability of HCCE in designing distributed brain-like models. Additionally, we compare HCCE with two other evolutionary schemes: enforced subpopulation (ESP; Gomez, 2003), and ordinary unimodal evolution.

4.3.1 Hierarchical Cooperative Co-evolution We have performed five independent runs of the co-evolutionary procedure discussed in Section 4.2, evaluating the speed and robustness of the HCCE scheme. The results obtained are illustrated in Figure 13, where each column corresponds to a different run.

In the first run, the progress of CG2 evolution is not stabilized, being unable to formulate distinct WM activation patterns. The progress converges to a suboptimal, still satisfactory solution (in terms of behavioral performance). The next run was one of the most successful. After the formulation of WM patterns in PPC and PFC, they are effectively exploited by the learning process, which is capable of accomplishing the desired DMS or DNMS strategy. The third run was unsuccessful, as the WM-like activation pattern is formulated in the PPC (CG1) component but not in PFC (CG2). Additionally, the reinforcement learning was not successful, as the strategy followed was “always turn to the right.” The next run was also very successful. It progressively formulated WM patterns, and additionally exploited them to adopt either the DMS or the DNMS strategy during learning. The last run was suboptimal, mainly because the navigation skills of the robot have not been designed effectively. The robot bumped on the walls quite often, both in the pre- and post-lesion operation modes.

In an attempt to formulate general comments on the progress of the HCCE scheme, we can state that the robot navigation skills are very important for the success of the experiment. This is because if the robot is not able to navigate successfully in the T-maze, it has no chance to obtain any reinforcement. Additionally, the development of WM patterns is very critical for the success of the composite scheme, because if they are not sufficiently developed, the simulated robot cannot remember the sample cue in order to express the appropriate DR. Furthermore, WM patterns are also very important for the success of the reinforcement learning process as the memorized

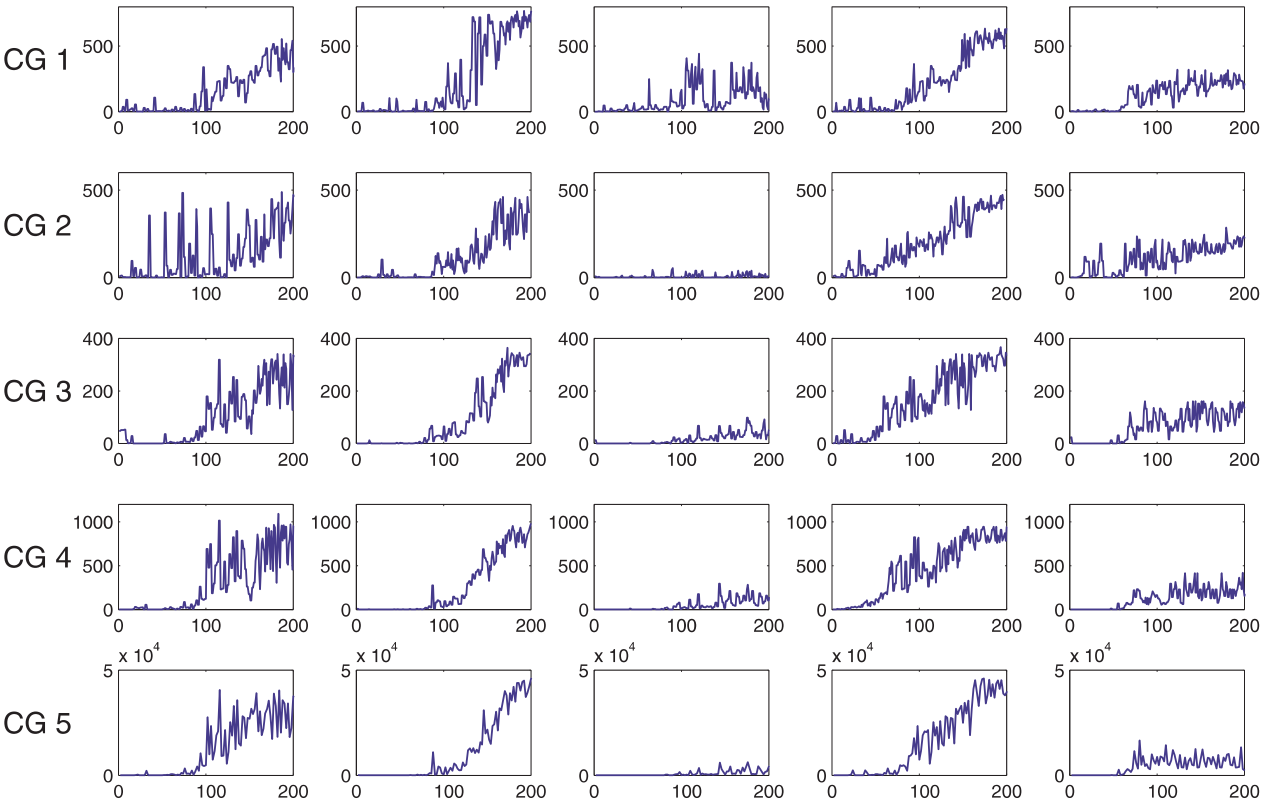


Figure 13 Graphical illustration of the progress of five different HCCE procedures. Each column is related to the results observed on the respective run. Each plot illustrates the maximum fitness value in a generation, against evolutionary epochs.

information has to be combined with the location of the reward signal in order to infer the desired response strategy.

4.3.2 Enforced Subpopulation Additionally, we have investigated whether a different co-evolutionary scheme is capable of solving the same problem, specifying successfully the structure of cortical and link agents. In particular, we have approached the problem discussed above by utilizing the ESP co-evolutionary scheme. In the current work, we have implemented the ESP algorithm described in Gomez (2003), without however activating the stagnation check that practically re-initializes populations when the process becomes stalled.

All populations of the ESP scheme are evolved according to a common set of objectives, utilizing the same fitness function (Gomez, 2003). Hence, the results of accomplishing the three tasks (DMS training, DNMS training, and wall avoidance) by either the composite or the eliminated configurations of the model are combined to a single measure. Similar to the function f_{CG5} that evolves the top-level CG of the HCCE scheme (see Equation 11), the fitness of ESP-assembled solutions is measured by

$$f = [E^{PPC}_{wm,\,dms}\, E^{PFC}_{wm,\,dms}(1 + E_{tr,\,dms})] \times [E^{PPC}_{wm,\,dnms} E^{PFC}_{wm,\,dnms}(1 + E_{tr,\,dnms})]E_{wa}. \qquad (12)$$

This objective implies that the progress of ESP evolution is directly comparable with the progress of the HCCE scheme. We used 18 different species to specify the structure of the 18 components of the model. Similar to HCCE, each species consists of 200 individuals. All species are evolved according to the criteria described by Equation 12. Additionally, 2,000 assemblies of components are randomly generated in each evolutionary epoch (Gomez, 2003), trying

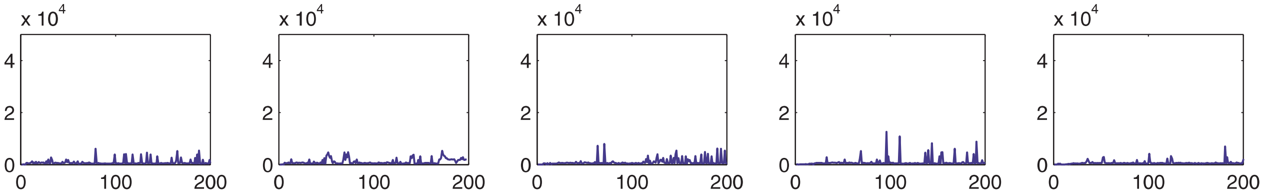


Figure 14 The results of five different runs of the ESP procedure. Each plot demonstrates the fitness value of the best candidate solution in a generation, against evolutionary epochs (compare with the last line of Figure 13).

to identify successful solutions of the overall problem.

We have performed five independent runs of the ESP scheme, which are evolved for 200 epochs. The probability of applying crossover and mutation operators over the structure of a cortical or a link agent is the same as the respective probabilities of the HCCE scheme. The results of these processes are illustrated in Figure 14, and are directly comparable with the last line of Figure 13. Evidently, none of the ESP processes was successful. This is mainly because the ESP process is not directed towards creating gradually more complex assemblies. In contrast, it is expected that because of the large number of complex assemblies being evaluated, satisfactory distributed configurations will be randomly formulated. Unfortunately, as indicated by the present results, this is not the case when many components need to be co-evolved.

4.3.3 Unimodal Evolution Finally, we have also approached the same problem by utilizing a unimodal evolutionary scheme. In particular, a single, large chromosome has been employed to encode the structure of all cortical and link agents of the model. Following the unimodal approach, it is not possible to evolve system components autonomously, and thus the parts of the genotype corresponding to candidate structures of systems components participate in only one composite solution. Additionally, a single fitness function applies to all components. Still, their separate role in the model can be revealed by testing the performance of candidate solutions in accomplishing the three different tasks.

The objective function that guides the evolutionary process is defined according to the fitness function of the top-level CG of the HCCE scheme. Hence, similar to f_{CG5} (see Equation 11), the fitness function is given by

$$f = [E^{\mathrm{PPC}}_{\mathrm{wm,\,dms}} E^{\mathrm{PFC}}_{\mathrm{wm,\,dms}} (1 + E_{\mathrm{tr,\,dms}})] \times [E^{\mathrm{PPC}}_{\mathrm{wm,\,dnms}} E^{\mathrm{PFC}}_{\mathrm{wm,\,dnms}} (1 + E_{\mathrm{tr,\,dnms}})] E_{\mathrm{wa}}, \qquad (13)$$

which implies that the progress of unimodal evolution is directly comparable with the progress of the HCCE scheme. In the current set of experiments, a population of 400 individuals has been evolved for 200 steps. The probability of applying crossover and mutation operators over the structure of a cortical or a link agent is the same as the respective probabilities of the co-evolutionary scheme.

We performed five independent runs of the unimodal evolutionary process. The results of each process are illustrated in Figure 15. These results are directly comparable with the last line of Figure 13. Evidently, none of the ordinary evolutionary processes was successful. This is because ordinary evolution

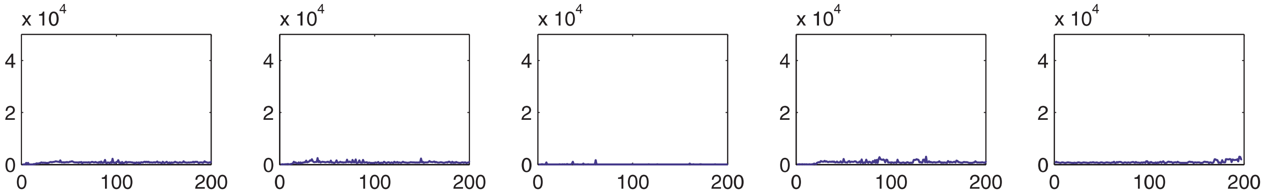


Figure 15 Graphical illustration of the progress of five different unimodal evolutionary procedures. Each plot demonstrates the maximum fitness value of individuals in a generation, against evolutionary epochs (compare with the last line of Figure 13).

employs a single population with individuals encoding compact solutions, and additionally employs a single fitness function, which is not able to highlight the role of each component to the system. These results highlight the unsuitability of unimodal evolution to design distributed structures consisting of autonomous components. Additionally, they highlight the need for a specialized scheme able to consider explicitly the individual characteristics of substructures.

4.3.4 Comments In the present set of experiments, we have utilized three different evolutionary methods (HCCE, ESP, and ordinary unimodal evolution) to address the design of the brain-like computational model. The results obtained are illustrated in Figures 13, 14, and 15. By comparing these figures, we can easily observe that HCCE significantly outperforms both ESP and unimodal processes, when addressing problems where the special characteristics of substructures have to be explored. This is because the proposed co-evolutionary scheme is able to evolve large distributed systems, enforcing the cooperation among components having separate roles. Furthermore, a previous study on HCCE internal dynamics (Maniadakis, 2006) has shown that the replication operator significantly facilitates the successful convergence of the composite co-evolutionary process, because it conveys information from the higher to the lower levels of the hierarchy, in order to modulate and coordinate partial evolutionary processes.

Because of the embodiment of the cognitive system in the simulated robotic platform and the observation of robot performance on several tasks (each one testing a large number of simulation steps), all evolutionary processes demanded several hours to run for 200 evolutionary epochs. The experiments have been performed on a PC with an Intel Pentium 4 processor at 3.00 GHz, and 512 MB RAM. Each HCCE run evolved for approximately 16 h, ESP runs evolved for approximately 75 h, while unimodal evolution also evolved for approximately 16 h. The distribution of processing time for each methodology is explained by the number of composite solution assemblies evaluated by HCCE, ESP, and unimodal schemes in each evolutionary epoch. Specifically, the HCCE scheme evaluates 400 assemblies, ESP evaluates 2,000 assemblies, and unimodal evolution evaluates 400 assemblies. Thus, it is reasonable that ESP needs considerably more processing time, because it inherently performs more evaluations (the individuals encoding component structures have to participate in many composite assemblies, in order to obtain an average estimate of their quality). Alas, despite the increased amount of computational resources spent, the quality of the obtained results is rather poor for ESP.

Overall, from the aforementioned set of experiments, we conclude that HCCE is more effective than both ESP and ordinary unimodal evolution for designing distributed systems consisting of many complex and autonomous components. Moreover, it has been illustrated that HCCE utilizes efficiently the available computational resources, being at least as fast as the unimodal evolution and much faster when compared with ESP.

5 Discussion

The computational framework presented in this article introduces a new engineering perspective in designing brain-inspired cognitive systems, bearing a twofold contribution. First, it follows a novel distributed modeling approach, with each brain area represented by an agent structure. Thus, brain areas are considered as autonomous entities, which co-exist in a common functional environment. The interactive dynamics among the composite computational cognitive system, the body of the artifact, and the external physical environment specify the "brain-arena," where the agents need to interact, communicate, and cooperate with each other. This is a new view that introduces agent technology into the field of computational cognitive neuroscience, supporting the design of complex brain-like systems.

Second, a distributed optimization approach is employed to design the components of the model, facilitating their cooperation. Specifically, we utilize an HCCE scheme that is able to design complex distributed systems consisting of many partial components. The co-evolutionary scheme designs autonomous agents by using separate fitness functions for their evolution. As a result, distinct roles can be assigned to agents, mimicking effectively brain areas. This is a clear advantage of cooperative co-evolution compared to unimodal evolution, which calls for a single fitness function, preventing the consideration of partial performance of substructures. Furthermore, the HCCE design mechanism facilitates the integration of autonomous agents in a unified system, by means of evaluating their cooperative performance. Thus, the proposed co-evolutionary scheme can be utilized as a consistent design mechanism to facilitate the development of complex brain models. However, it is worth emphasizing that it is not necessary for the composite computational model to operate in a hierarchical mode. This is because the organization of hierarchical co-evolution is not reflected on the connectivity of agent structures. The performance of partial components can be either hierarchical or completely parallel. Hence, the HCCE-based design approach does not imply any further constraints. It is introduced only to support the process of designing complex distributed systems.

Overall, the current work aims at providing a consistent, unified framework that supports large-scale brain modeling efforts. In particular, because of the distributed nature of both the model and the design mechanism, the proposed framework is capable of (i) independently exploring the components of a model, (ii) integrating partial models (representing groups of brain areas; Maniadakis & Trahanias, 2005a), and (iii) whenever necessary, redesigning some of the components in order to formulate gradually more advanced structures (Maniadakis & Trahanias, 2006a). These particular features make the proposed computational framework very effective in terms of implementing complex artificial systems sharing similar working principles with biological brains. The proposed approach has been recently utilized to design a much larger brain-inspired cognitive system for a humanoid robot, modeling overlapping observation/execution brain pathways (Maniadakis, Hourdakis, & Trahanias, 2007).

Our approach is in contrast to the majority of existing brain modeling efforts focusing on specific brain areas, without, however, taking into account how the implemented structures could be members of large-scale models (e.g., Ajemian, Bullock, & Grossberg, 2000; Compte et al., 2000; Durstewitz, Seamans, & Sejnowski, 2000; Todorov, 2000). Unfortunately, these approaches make different and often contradictory assumptions, operating at different levels of description. As a result, existing models seem to form a heterogeneous collection, where computational differences among them make their integration to a global system very difficult (Wermter & Sun, 2000).

There is also another category of models employing assemblies of artificial neural networks to represent a set brain areas (Kozma, Wong, Demirer, & Freeman, 2005; Krichmar, Seth, Nitz, Fleischer, & Edelman, 2005; Sporns & Alexander, 2002). However, these approaches suffer in terms of scalability, because they are not supported by a (semi-)automated design procedure that facilitates the re-usability of substructures (e.g., by means of evolution; Harter, 2005). Thus, they cannot be easily employed as a general purpose approach capable to support long-term brain modeling efforts.

For comparative purposes, in the current study we have also employed ESP co-evolution and ordinary unimodal evolution for engineering brain-inspired models; however, none of these was successful. It has been experimentally demonstrated that HCCE is the only effective method (of the three tested) to evolve systems consisting of many components, investigating the functionality of the composite structure in a range of different operating conditions. Previous studies on the characteristics of co-evolutionary design mechanism have also demonstrated the suitability of the underlying scheme (Maniadakis, 2006). It is worth emphasizing that the HCCE scheme can be utilized as a general purpose mechanism for designing distributed systems consisting of many cooperating components. In fact, it can be effectively applied in contexts very different from that discussed in the present study, such as the design of complex modular mechanical structures or the design of teams of cooperating robots.

6 Conclusions

In this article, we have proposed a novel computational framework for designing distributed brain-inspired cognitive systems, following a co-evolutionary agent-based approach. Particularly, neural network agents are employed to represent brain areas following a similar connectivity to the biological central nervous system. An HCCE scheme is utilized to support design specification of agent structures. The proposed co-evolutionary method is suitable for the design of complex distributed systems, because it offers increased search abilities of partial components, and is able to emphasize both their autonomous characteristics and their cooperative performance.

The work presented here constitutes a first attempt towards a rigorous method for the design of biologically inspired artificial cognitive systems, based on cooperative co-evolution. The results obtained attest to its validity and effectiveness in modeling partial brain areas and replicating biological behaviors. Further work is necessary, and currently underway, to investigate the suitability of our approach for large-scale modeling tasks and the endowment of advanced cognitive abilities to simulated and real robots.

Particularly, in the future we aim to implement brain-like systems embedded in robotic systems, considering (i) the accomplishment of many tasks by the same computational model, following many different training processes, and (ii) the computational replication of many biological lesion effects after simulated lesions to the components representing cortical areas. The more biological data the model is able to replicate, the more similar to cortical areas agents will become, facilitating the long-term goal of implementing artificial systems with mammal-like cognitive skills.

Note

[1] Epigenesis here includes all learning processes during lifetime.

References

Aharonov, R., Segev, L., Meilijson, I., & Ruppin, E. (2003). Localization of function via lesion analysis. *Neural Computation*, *15*(4), 885–913.

Ajemian, R., Bullock, D., & Grossberg, S. (2000). *A model of movement coordinates in motor cortex: posture-dependent changes in the gain and direction of single cell tuning curves*. Department of Cognitive and Neural Systems, Boston University.

Barto, A., & Mahadevan, S. (2003). Recent advantages in hierarchical reinforcement learning. *Discrete Event Dynamic Systems: Theory and Applications*, *13*, 41–77.

Blynel, J., & Floreano, D. (2002). Levels of dynamics and adaptive behaviour in evolutionary neural controllers. In B. Hallam, D. Floreano, J. Hallam, G. Hayes, & J.-A. Meyer (Eds.), *From Animals to Animats 7: Proceedings of the 7th International Conference on Simulation of Adaptive Behavior* (pp. 272–281). Cambridge, MA: MIT Press-Bradford Books.

Casillas, J., Cordón, O., Herrera, F., & Merelo, J. (2001). Cooperative co-evolution for learning fuzzy rule-based systems. In P. Collet, C. Fonlupt, J.-K. Hao, E. Lutton, & M. Schoenauer (Eds.), *Proceedings of the 5th Conference on Artificial Evolution (AE-2001)* (pp. 311–322). Berlin: Springer-Verlag.

Compte, A., Brunel, N., Goldman-Rakic, P., & Wang, X.-J. (2000). Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cerebral Cortex*, *10*(1), 910–923.

Cotterill, R. (2001). Cooperation of the basal ganglia, cerebellum, sensory cerebrum and hippocampus: possible implications for cognition, consciousness, intelligence and creativity. *Progress in Neurobiology*, *64*(1), 1–33.

De Jong, E. (2003). Representation development from pareto-co-evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2003)*, *Lecture Notes in Computer Science*, Vol. 2723. Berlin: Springer.

De Jong, E. (2004). Intransitivity in co-evolution. In *Proceedings of the International Conference on Parallel Problem Solving from Nature (PPSN-2004)*, *Lecture Notes in Computer Science*, Vol. 3242. Berlin: Springer.

Delgado, M., Zuben, V. F., & Gomide, F. (2004). Co-evolutionary genetic fuzzy systems: a hierarchical collaborative approach. *Fuzzy Sets and Systems*, *141*(1), 89–106.

Durstewitz, D., Seamans, J., & Sejnowski, T. (2000). Neurocomputational models of working memory. *Nature Neuroscience*, *3*, 1184–1191.

Ficici, S., & Pollack, J. (2001). Pareto optimality in co-evolutionary learning. In *Proceedings of the 6th European Conference on Artificial Life (ECAL-2001)* (pp. 316–325).

Floreano, D., & Urzelai, J. (2000). Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Networks*, *13*, 431–443.

Fuster, J. (2000). Executive frontal functions. *Experimental Brain Research*, *133*, 66–70.

Garcia-Pedrajas, N., Ortiz-Boyer, D., & Hervas-Martinez, C. (2004). Cooperative co-evolution of generalized multilayer perceptrons. *Neurocomputing*, *56*, 257–283.

Goel, V., Pullara, S., & Grafman, J. (2001). A computational model of frontal lobe dysfunction: working memory and the tower of hanoi task. *Cognitive Science*, *25*, 287–313.

Gomez, F. (2003). *Robust non-linear control through neuroevolution*. Ph.D. Thesis, AI-TR-03-303, Department of Computer Sciences, University of Texas at Austin.

Gomez, F., & Miikkulainen, R. (1999). Solving non-markovian control tasks with neuro-evolution. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-1999)* (pp. 1356–1361).

Granon, S., Vidal, C., Thinus-Blanc, C., Changeux, J.-P., & Poucet, B. (1994). Working memory, response selection, and effortful processing in rats with medial prefrontal lesions. *Behavioral Neuroscience*, *108*(5), 883–891.

Harter, D. (2005). Evolving neurodynamic controllers for autonomous robots. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN-2005)* (pp. 137–142).

Iida, M., & Tanaka, S. (2002). Postsynaptic current analysis of a model prefrontal cortical circuit for multi-target spatial working memory. *Neurocomputing*, *44–46*, 855–861.

Iorio, A., & Li, X. (2004). A cooperative co-evolutionary multiobjective algorithm using non-dominated sorting. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*, *Lecture Notes in Computer Science*, Vol. 3103 (pp. 537–548). Berlin: Springer.

Ito, M., Noda, K., Hoshino, Y., & Tani, J. (2006). Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model. *Neural Networks*, *19*, 323–337.

Jennings, N. (2000). On agent based software engineering. *Artificial Intelligence*, *117*, 277–296.

Jennings, N. (2001). An agent-based approach for building complex software systems. *Communications of the ACM*, *44*(4), 35–41.

Kozma, R., Wong, D., Demirer, M., & Freeman, W. J. (2005). Learning intentional behavior in the K-model of the amygdala and ethorinal cortex with the cortico-hyppocamal formation. *Neurocomputing*, *65–66*, 23–30.

Krawiec, K., & Bhanu, B. (2003). Co-evolution and linear genetic programming for visual learning. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2003)* (pp. 332–343).

Krichmar, J., Seth, A., Nitz, D., Fleischer, J., & Edelman, G. (2005). Spatial navigation and causal analysis in a brain-based device modeling cortical-hippocampal interactions. *Neuroinformatics*, *5*, 197–222.

Landau, S., & Picault, S. (2001). Modelling adaptive multi-agent systems inspired by developmental biology. In *Proceedings of Multi-Agent-Systems and Applications (MASA-2001)* (pp. 221–229).

Lee, M. (2003). Evolution of behaviours in autonomous robot using artificial neural network and genetic algorithm. *Information Sciences*, *155*, 43–60.

Makar, R., Mahadevan, S., & Ghavamzadeh, M. (2001). Hierarchical multi-agent reinforcement learning. In *Proceedings of Autonomous Agents 2001*.

Maniadakis, M. (2006). *Design and integration of agent-based partial brain models for robotic systems by means of hierarchical cooperative co-evolution.* Ph.D. Thesis, Department of Computer Sciences, University of Crete.

Maniadakis, M., Hourdakis, E., & Trahanias, P. (2007). Modeling overlapping execution/observation brain pathways. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN-07)* (pp. 1255–1260).

Maniadakis, M., & Trahanias, P. (2005a). Co-evolutionary incremental modelling of robotic cognitive mechanisms. In *Proceedings of the VIIIth European Conference on Artificial Life (ECAL-2005)* (pp. 200–209).

Maniadakis, M., & Trahanias, P. (2005b). Distributed brain modelling by means of hierarchical collaborative co-evolution. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC-2005)* (pp. 2699–2706).

Maniadakis, M., & Trahanias, P. (2005c). A hierarchical co-evolutionary method to support brain-lesion modelling. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN-2005)* (pp. 434–439).

Maniadakis, M., & Trahanias, P. (2006a). Hierarchical cooperative co-evolution facilitates the redesign of agent-based systems. In *Proceedings of the 9th International Conference on the Simulation of Adaptive Behavior, (SAB-2006)*, *Lecture Notes in Artificial Intelligence*, Vol. 4095 (pp. 582–593). Berlin: Springer.

Maniadakis, M., & Trahanias, P. (2006b). Modelling brain emergent behaviors through co-evolution of neural agents. *Neural Networks Journal*, *19*(5), 705–720.

Maniadakis, M., & Trahanias, P. (2007). Agent-based brain modelling by means of hierarchical cooperative co-evolution. For publication, submitted manuscript.

Minsky, M. (1986). *The society of mind*. New York: Simon and Schuster.

Moody, S., Wise, S., Pellegrino, G., & Zipser, D. (1998). A model that accounts for activity in primate frontal cortex during a delayed matching-to-sample task. *Journal of Neuroscience*, *18*(1), 399–410.

Moriarty, D., & Miikkulainen, R. (1997). Forming neural networks through efficient and adaptive co-evolution. *Evolutionary Computation*, *5*(4), 373–399.

Murray, E., Bussey, T., & Wise, S. (2000). Role of prefrontal cortex in a network for arbitrary visuomotor mapping. *Experimental Brain Research*, *113*, 114–129.

Olsson, B. (2001). Co-evolutionary search in asymmetric spaces. *Information Science*, *133*, 103–125.

Pennartz, C. (1997). Reinforcement learning by hebbian synapses with adaptive thresholds. *Neuroscience*, *81*(2), 303–319.

Polk, T., Simen, P., Lewis, R., & Freedman, E. (2002). A computational approach to control in complex cognition. *Brain Research Interactive*, *15*, 71–83.

Pontecorvo, J. M., Sahgal, A., & Steckler, T. (1996). Further developments in the measurment of working memory in rodents. *Cognitive Brain Research*, *3*, 205–213.

Potter, M., & De Jong, K. (2000). Cooperative co-evolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, *8*, 1–29.

Ragozzino, M., & Kesner, R. (2001). The role of rat dorsomedial prefrontal cortex in working memory for egocentric responces. *Neuroscience Letters*, *308*, 145–148.

Redish, A., Elga, A., & Touretzky, S. (1996). A coupled attractor model of the rodent head direction system. *NETWORK*, *7*(4), 671–685.

Reilly, G. (2001). Collaborative cell assemblies: building blocks of cortical computation. In S. Wermter, J. Austin, & D. Willshaw (Eds.), *Emergent neural computational architectures based on neuroscience: towards neuroscience-inspired computing*, *Lecture Notes in Computer Science*, Vol. 2036 (pp. 161–173). Berlin: Springer-Verlag.

Rolls, E., & Stringer, S. (2000). On the design of neural networks in the brain by genetic evolution. *Progress in Neurobiology*, *61*, 557–579.

Rosin, C., & Belew, R. (1997). New methods for competitive co-evolution. *Evolutionary Computation*, *5*, 1–29.

Singh, P. (2003). Examining the society of mind. *Computing and Informatics*, *22*(5), 521–543.

Sporns, O., & Alexander, W. (2002). Neuromodulation and plasticity in an autonomous robot. *Neural Networks*, *15*, 761–774.

Thierens, D. (1999). Scalability problems of simple genetic algorithms. *Evolutionary Computation*, *7*(4), 331–352.

Tkaczyk, E. (2001). Pressure hallucinations and patterns in the brain. *Morehead El. Journal of Applicable Mathematics*, *1*, 1–26.

Todorov, E. (2000). Direct cortical control of muscle activation in voluntary arm movements: a model. *Nature Neuroscience*, *3*, 391–398.

Wermter, S., & Sun, R. (2000). *Hybrid neural systems* (pp. 6–18). Heidelberg: Springer-Verlag.

Wiegand, R., Liles, C., & De Jong, A. (2001). An empirical analysis of collaboration methods in cooperative co-evolutionary algorithms. In L. Spector et al. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)* (pp. 1235–1242). San Francisco, CA: Morgan Kaufmann.

About the Authors

Michail Maniadakis received a B.Sc. in informatics (1996) from the University of Piraeus, an M.Sc. in systems automation (1999) from the Technical University of Crete and a Ph.D. in cognitive robotics (2006) from the University of Crete. His research interests are focused on the multi-criteria design of brain-like artificial systems, the advancement of co-evolutionary algorithms, and the design of intelligent robotic systems.

Panos Trahanias received his Ph.D. in computer science from the National Technical University of Athens, Greece. Currently, he is a Professor with the University of Crete, Greece and the Foundation for Research and Technology – Hellas (FORTH). From 1991 to 1993 he was with the Department of Electrical and Computer Engineering, University of Toronto, Canada, as a Research Associate. He has participated in many RTD programs in image analysis at the University of Toronto and has been a consultant to SPAR Aerospace Ltd, Toronto. Since 1993, he has been with the University of Crete and FORTH; currently, he is the Director of Graduate Studies, Department of Computer Science, University of Crete, and the Head of the Computational Vision and Robotics Laboratory at FORTH, where he is engaged in research and RTD projects in autonomous mobile platforms, cognitive robotics, sensory technologies, computational vision, mixed realities and human–robot interaction. Professor Trahanias has extensive experience in the execution and coordination of large research projects. Moreover, his work has been published extensively in scientific journals and conferences. He has participated in the Program Committees of numerous International Conferences; he has been the General Chair of Computer Graphics International 2004, and will be General Co-Chair of Eurographics 2008 and the European Conference on Computer Vision 2010.
E-mail: trahania@ics.forth.gr