3-2010

A self-organizing neural architecture integrating desire, intention and reinforcement learning

Ah-hwee TAN
Singapore Management University, ahtan@smu.edu.sg

Yu-Hong FENG

Yew-Soon ONG

Neurocomputing 73 (2010) 1465–1477

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A self-organizing neural architecture integrating desire, intention and reinforcement learning

Ah-Hwee Tan [a,*], Yu-Hong Feng [a,b], Yew-Soon Ong [a]

[a] School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore
[b] Center for High Performance Computing Research, Shenzhen Institute of Advanced Technology, CAS 1068 Xueyuan Avenue, Shenzhen University Town, Nanshan District, Shenzhen 518055, China

ARTICLE INFO

Article history:
Received 14 January 2009
Received in revised form
13 July 2009
Accepted 19 November 2009
Commuciated by A. Suarez
Available online 22 December 2009

Keywords:
Reinforcement learning
Plan learning
Self-organizing neural networks
BDI
Minefield navigation

ABSTRACT

This paper presents a self-organizing neural architecture that integrates the features of belief, desire, and intention (BDI) systems with reinforcement learning. Based on fusion Adaptive Resonance Theory (fusion ART), the proposed architecture provides a unified treatment for both intentional and reactive cognitive functionalities. Operating with a sense-act-learn paradigm, the low level reactive module is a fusion ART network that learns action and value policies across the sensory, motor, and feedback channels. During performance, the actions executed by the reactive module are tracked by a high level intention module (also a fusion ART network) that learns to associate sequences of actions with context and goals. The intention module equips the architecture with deliberative planning capabilities, enabling it to purposefully maintain an agenda of actions to perform and to reduce the need of constantly sensing the environment. Through reinforcement learning, plans can also be evaluated and refined without the rigidity of user-defined plans. We examine two strategies for combining the intention and reactive modules for decision making in a real time environment. Our experiments based on a minefield navigation domain show that the integrated architecture is able to learn plans efficiently, achieve good plan utilization, and combine both intentional and reactive action execution to yield a robust performance.

1. Introduction

Neural network models have been applied widely to the problems of pattern classification and signal processing. However, they have found to be much less successful in modelling high level cognitive functions of human brain. The main obstacles encountered include the difficulty of achieving deliberative functions using neural networks and the lack of an architectural framework for integrating the low level functionalities of neural models into high level intelligence.

Derived from folk psychology, belief, desire, and intention (BDI) is a popularly used framework for human modelling and logical reasoning [1]. BDI systems encode goal directed behaviours by using plans derived from expert knowledge about the task domain. An abstract plan consists of an ordered set of subgoals or actions that an agent should execute with little sensory feedback from its environment, together with an overall goal which that plan achieves if execution completes successfully. A traditional limitation of BDI architectures, however, is the lack of learning ability. In most cases, plans and capabilities are predefined by developers or captured from human experts.

In view of their complementary strengths, there have been great interests in hybrid architectures that integrate high level symbolic systems, such as BDI systems with various types of machine learning algorithms. Some examples of hybrid systems include CLARION [2], BDI with standard *Q*-learning [3], BDI with decision tree induction [4], and ACT-R with sequence learning [5]. Among these hybrid systems, temporal difference learning using gradient descent based function approximator has been most commonly used. However, the gradient descent methods learn by making small error correction steps iteratively. In addition, there is the issue of instability as learning of new patterns may erode the previously learned knowledge. Consequently, the resultant systems may not be able to learn and operate in real time.

To address the above limitations, Tan et al. [6] presented a hybrid architecture called BDI-FALCON that integrates BDI components, namely desire and intention, with a reinforcement learning system known as temporal difference—fusion architecture for learning and cognition (TD-FALCON) [7,8]. TD-FALCON is a three-channel fusion Adaptive Resonance Theory (ART) network [9] that incorporates temporal difference methods [10,11] into

* Corresponding author.
E-mail addresses: asahtan@ntu.edu.sg (A.-H. Tan).
asysong@ntu.edu.sg (Y.-H. Feng), yuhongf2009@gmail.com (Y.-S. Ong).

0925-2312/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2009.11.012

Adaptive Resonance Theory (ART) models [12,13] for reinforcement learning. By inheriting the ART code stabilizing and dynamic network expansion mechanism, TD-FALCON is capable of learning cognitive nodes encoding multi-dimensional mappings across multi-modal input patterns, involving states, actions, and rewards, in an online and incremental manner. It has displayed superior learning capabilities, compared with gradient descent based reinforcement learning systems in various benchmark experiments [8,14].

Using belief–desire–intention as the system's framework, the BDI-FALCON architecture extends a low level reinforcement learner TD-FALCON into a deliberative reasoner. Specifically, the intention module equips the system with deliberative planning capabilities, by enabling it to purposely maintain an agenda of actions to perform. This is useful in complex environment wherein reactive responses are not adequate. Performing with a plan also reduces the need for an agent to repeatedly sense the environment and may therefore improve efficiency.

Although the BDI-FALCON system illustrates the feasibility and benefits of integrating features of BDI with reinforcement learning, the architecture still adopts a symbolic implementation of the desire and intention modules. As such, the intention module has a limited learning capability and tends to generate a large number of plans, leading to substantial overhead in plan selection and evaluation. In addition, the strict plan matching requirement results in a poor plan utilization, resulting BDI-FALCON to frequently fall back on the reactive TD-FALCON for action selection. Therefore, there is limited improvement in system efficiency in terms of reducing the number of sensing.

In this paper, we present a fully connectionist architecture, known as *cBDI-FALCON*, that realizes the features of BDI, namely desire and intention, as well as reinforcement learning. In contrast to our prior work of BDI-FALCON [6], the intention module in the cBDI-FALCON is also modelled as a fusion ART neural network [9]. Doing so enables plans to be learned and updated through reinforcement learning according to the outcomes of their use in a natural way. In addition, when the confidence value of a plan drops below a certain threshold, it can be pruned from the system. The resultant reduction in the number of plans can thus further improve plan utilization and the system's performance.

Besides the intention and reactive modules, the desire module in the architecture is also modelled as a one-channel fusion ART network. The overall design philosophy is thus aiming towards a unified framework by using a principled set of computational processes for supporting both intentional and reactive behaviour. This unified view allows one to look at the intention and reactive modules at an angle that avoids the over-specialization of the computational processes for supporting the high level and low level cognitive functionalities. In theory, the computations in the system can be executed in parallel using neural networks, enabling the potential of speeding up [15].

To combine planned and reactive capabilities, we have developed two strategies, known as the follow-through and the re-evaluation strategies to coordinate the output produced by the intention and reactive modules. We have conducted extensive experiments to analyse the behaviours of the integrated architecture, in terms of plan utilization, system efficiency, and the overall success rates. To evaluate the system's performance empirically, we have chosen a minefield navigation task, similar to the one developed at the Naval Research Laboratory (NRL) [16]. The task involves an autonomous vehicle (AV) learning to navigate through obstacles to reach a stationary target (goal) within a specified number of steps. Our experimental results show that the integrated neural architecture is able to combine intentional and reactive action execution, leading to improvement both in terms of task completion performance and efficiency.

The rest of the paper is organized as follows. Section 2 presents a brief review of the related works on hybrid systems. Section 3 introduces the fusion ART network model [9], together with its learning and prediction algorithms. Section 4 presents the cBDI-FALCON architecture and its three main components, namely the desire module, the intention module, and the reactive module. Section 5 reports our case study on the minefield navigation problem. The final section provides a discussion of the results achieved and highlights the outstanding issues.

2. Related works

Hybrid systems integrating high level capabilities, such as planning, and low level reactive modules involving learning has been an active research area. Sun described a two-level model, known as CLARION [2], for learning reactive plans and extracting plans from reinforcement learners. The first three layers of the bottom level form a backpropagation network learning and computing Q-values. The fourth layer (the top level with only one node) determines probabilistically the action to be performed based on a Boltzmann distribution. Given a specific problem scenario, plans can be generated on the spot using a beam search strategy that chains up actions with optimal Q-values at each step [17]. A key strength of the approach is the capability to perform probabilistic planning. This is useful in environments where an action may lead to non-deterministic outcomes. The plan extraction process however assumes that the next state after performing each and every action can always be determined beforehand. In addition, plans are extracted on-the-fly and the system does not consider the use of a plan library as in BDI systems.

Using a goal-directed (top down) approach, Wallis discussed the notion of *goal-tagged activities* [18], that achieved planning by low-level adherence to high-level goals without the need for explicit symbolic representation. This was achieved by the chaining of implicitly goal-encoded *activities* (which are reactive modules similar to plans in the BDI sense).

Plan learning is akin to the topic of sequence learning, wherein simple recurrent networks (SRN) [19,20] has been extensively applied. In view that SRN performs implicit learning through the use of recurrent connections, Lebiere and Wallach proposed a sequence learning model of ART-R [5], where the mappings between sensory stimuli and target actions were encoded as chunks explicitly. The model however relies on a set of productions to encode the procedure knowledge of making declarative chunks. The approach is thus symbolic and it is not clear how such procedure knowledge can be learned from the environment.

Working from the BDI perspective, Heinze et al. [21] demonstrated the synergistic coupling of a machine learning architecture (CLARET) with a BDI top-layer to achieve plan recognition of aircraft. CLARET was used to recognize spatial trajectories of aircraft and other observable objects, and the BDI layer processed these observations with higher-level, goal-directed reasoning. Guerra-Hernandez et al. [4] expanded the BDI architecture to incorporate learning in multi-agent systems using a first order method called induction of decision tree. Another extended BDI architecture was proposed by Norling [3] for human modelling by incorporating cognitive processes of folk psychology for recognition primed decision making. A standard table lookup version of Q-learning was adopted to learn reactive rules for path finding in a grid world. Subagdja and Sonenberg [22] further extended BDI architecture to incorporate learning, through the generation and testing of hypothesis for formulating plans.

As a precursor to our work, Karim et al. [23] proposed a hybrid system consisting of a high level BDI system and a low level reactive FALCON [24], in which BDI-styled plans were learned out of FALCON's reactive action execution. The Plan Generation System (PGS) used a strategy to build plans by appending actions as the system performs. The hybrid architecture was originally illustrated on a minefield navigation domain and was subsequently expanded and applied to a multi-agent predator–prey domain [25].

The approach presented in this paper is similar to that of PGS presented by Karim et al. [23,25]. However, in our integrated architecture, the desire and intention modules are also modelled as self-organizing neural networks. Consequently, we offer a whole new set of plan learning, selection and evaluation algorithms based on neurally plausible processes, and expand the analysis significantly by experimenting with various strategies for integrating the intentional and reactive learning modules.

3. Fusion ART

Fusion Adaptive Resonance Theory (ART) [9] employs a multi-channel architecture (Fig. 1), comprising a category field F_2 connected to a fixed number of (K) pattern channels or input fields through bidirectional conditionable pathways. The model unifies a number of network designs, most notably Adaptive Resonance Theory (ART) [26–29], Adaptive Resonance Associative Map (ARAM) [30,31] and Fusion Architecture for Learning, COgnition, and Navigation (FALCON) [24,8,14], developed over the past decades for a wide range of functions and applications. The network dynamics of fusion ART based on the fuzzy ART operations [32] is summarized as follows.

Input vectors: Let $\mathbf{I}^{ck} = (I_1^{ck}, I_2^{ck}, \ldots, I_n^{ck})$ denote the input vector, where $I_i^{ck} \in [0,1]$ indicates the input i to channel ck. With complement coding, the input vector $\mathbf{I}^{ck}$ is augmented with a complement vector $\overline{\mathbf{I}}^{ck}$ such that $\overline{I}_i^{ck} = 1\text{-}I_i^{ck}$.

Activity vectors: Let $\mathbf{x}^{ck} = (x_1^{ck}, x_2^{ck}, \ldots, x_n^{ck})$ denote the F_1^{ck} activity vector. Let $\mathbf{y}$ denote the F_2 activity vector. Upon input pattern presentation, $\mathbf{x}^{ck} = \mathbf{I}^{ck}$.

Weight vectors: Let $\mathbf{w}_j^{ck}$ denote the weight vector associated with the j th node in F_2 for learning the input patterns in F_1^{ck}. Initially, F_2 contains only one *uncommitted* node and its weight vectors contain all 1-s.

Parameters: The fusion ART's dynamics is determined by choice parameters $\alpha^{ck} \geq 0$, learning rate parameters $\beta^{ck} \in [0,1]$, contribution parameters $\gamma^{ck} \in [0,1]$ and vigilance parameters $\rho^{ck} \in [0,1]$ for each of the input field k for $k = 1, \ldots, K$.

As a natural extension of ART, fusion ART responds to incoming patterns in a continuous manner. It is important to note that at any point in time, fusion ART does not require input to be present in all the pattern channels. For those channels not receiving input, the input vectors are initialized to all 1s. The fusion ART pattern processing cycle comprises of five key stages, namely code activation, code competition, activity readout, template matching, and template learning, as described below.

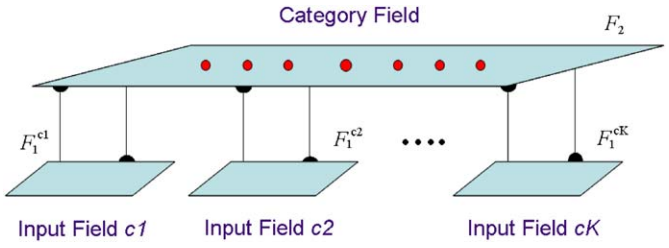

Fig. 1. The fusion ART architecture.

Code activation: Given the activity vectors $\mathbf{x}^{c1}, \ldots, \mathbf{x}^{cK}$, for each F_2 node j, the choice function T_j is computed by

$$T_j = \sum_{k=1}^{K} \gamma^{ck} \frac{|\mathbf{x}^{ck} \wedge \mathbf{w}_j^{ck}|}{\alpha^{ck} + |\mathbf{w}_j^{ck}|}, \tag{1}$$

where the fuzzy AND operation $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv min(p_i, q_i)$, and the norm $|.|$ is defined by $|\mathbf{p}| \equiv \sum_i p_i$ for vectors $\mathbf{p}$ and $\mathbf{q}$.

Code competition: A code competition process follows under which the F_2 node with the highest choice function value is identified. When a category choice is made at node J, $y_J = 1$; and $y_j = 0$ for all $j \neq J$. This indicates a winner-take-all strategy.

Template matching: Before the node J can be used for prediction and learning, a template matching process checks that the weight templates of node J are sufficiently close to their respective input patterns. Specifically, resonance occurs if for each channel k, the *match function* m_J^{ck} of the chosen node J meets its vigilance criterion:

$$m_J^{ck} = \frac{|\mathbf{x}^{ck} \wedge \mathbf{w}_J^{ck}|}{|\mathbf{x}^{ck}|} \geq \rho^{ck}. \tag{2}$$

If any of the vigilance constraints is violated, mismatch reset occurs in which the value of the choice function T_J is set to 0 for the duration of the input presentation. The search process then selects another F_2 node J until a resonance is achieved.

Template learning: Once a resonance occurs, for each channel ck, the weight vector $\mathbf{w}_J^{ck}$ is modified by the following learning rule:

$$\mathbf{w}_J^{ck(\text{new})} = (1\text{-}\beta^{ck})\mathbf{w}_J^{ck(\text{old})} + \beta^{ck}(\mathbf{x}^{ck} \wedge \mathbf{w}_J^{ck(\text{old})}). \tag{3}$$

When an uncommitted node is selected for learning, it becomes *committed* and a new uncommitted node is added to the F_2 field. Fusion ART thus expands its network architecture dynamically in response to the input patterns.

4. The integrated architecture

cBDI-FALCON is a hybrid architecture that integrates components of belief–desire–intension (BDI) architecture, namely the *desire* module and the *intention* module, with a *reactive* module subsuming a simplified version of the belief module. The three key modules, each realized as a fusion ART network, and their interrelation are shown in Fig. 2.

Desire module: The desire module maintains an explicit representation of the agent's goals. By matching the defined goals with the corresponding current state attributes, the desire module computes how well the system has progressed towards the desired goals. The computed degrees of goal attainment in turn serve as reward signals to the feedback field of the reactive module and the evaluation field of the intention module.

Reactive module: The low level reactive learning module is a TD-FALCON model that interacts with the environment through the sensory, motor, and feedback channels. Based on the goals defined in the desire module and the sensory inputs received from the environment, the reactive module performs reinforcement learning so as to acquire a set of action and value policies that enables the agent to achieve its goals.

Intention module: The intention module maintains the plan set and supports the key processes of plan learning, plan selection, plan execution and plan evaluation. Given a set of active goals and the current sensory inputs, the plan selection process identifies the most applicable plan to perform. During the plan execution,

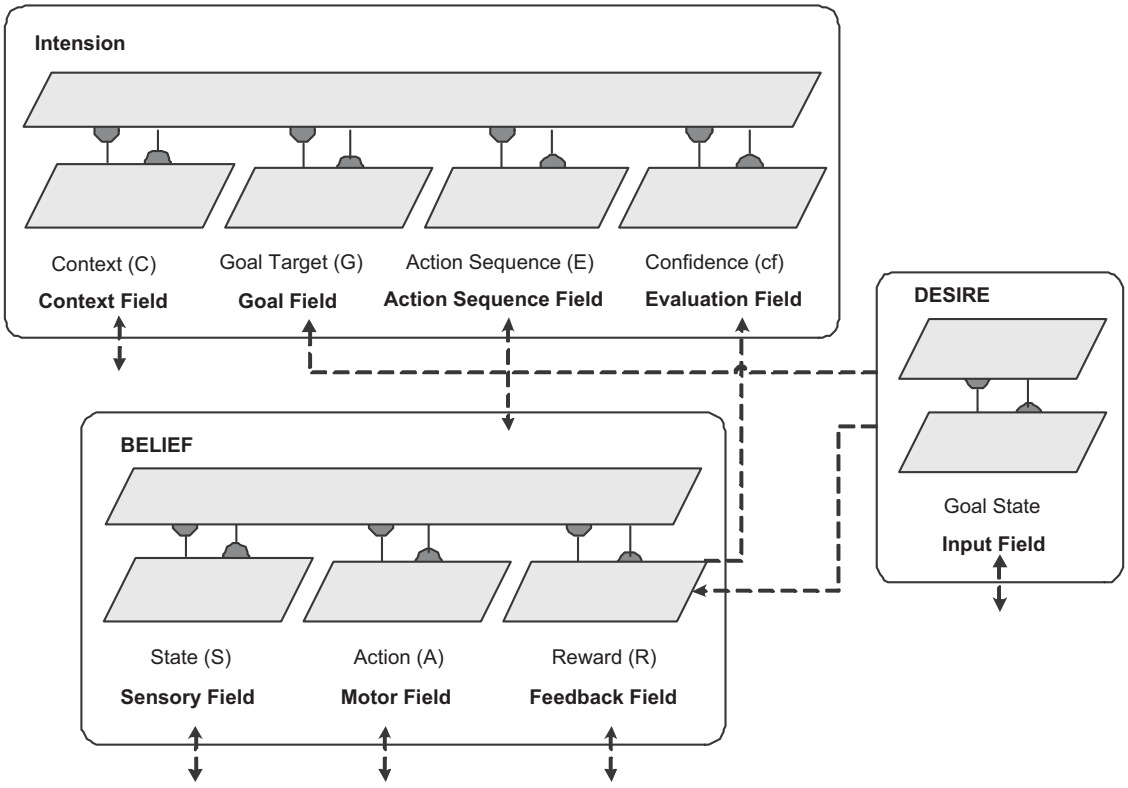

Fig. 2. A schematic architecture of cBDI-FALCON.

the action sequence of the adopted plan is extracted and performed through the motor channel of the reactive module. The execution of plans thus enables an agent to perform a series of actions without the need of going through the typical sense-act-learn cycle for each action. This could potentially lead to saving in computation cost and enable the system to be more resilient in a challenging environment, wherein external signals may not be available all the time. Through a simple form of reinforcement learning, the plan evaluation process adjusts the confidence value of each adopted plan according to the outcome that it leads to.

As mentioned above, each of the three modules can be implemented as a fusion ART network. The detailed algorithms and processes are described in the following sections.

4.1. The desire module

The desire module is a one-channel fusion ART network that consists of an input field F_1^d for representing the current goal state activities, and a cognitive field F_2^d for encoding the goal state vectors of the active goals. The desire module maintains and manages the agent's goals. It also handles the function of goal attainment evaluation. Specifically, each committed node in the F_2^d field is used to encode the target state of an active goal. The goal state vector $\mathbf{x}^d$ is typically a part of or can be derived from the state vector $\mathbf{S}$ represented at the sensory field of the reactive module.

During system initialization, the active goal target vectors are presented to the desire module for encoding. Using a procedure similar to the rule insertion algorithm [33], the goal target vector of each goal j is encoded by a weight vector $\mathbf{w}_j^d$ associated to the F_2^d node j.

During performance, after sensing the environment and extracting the goal state vector $\mathbf{x}^d$ derived from the current state vector $\mathbf{S}$, the goal attainment function A^d computes the match value between the goal state vector $\mathbf{x}^d$ and the goal target vectors $\mathbf{w}_j^d$, where $1 \le j \le \mathcal{N}$ and $\mathcal{N}$ is the number of active goals, with respect to the norm of each individual goal target vector. Specifically, given the current goal state vector $\mathbf{x}^d$, the overall goal attainment can be calculated as

$$A^d = \prod_{j=1}^{\mathcal{N}} \frac{|\mathbf{x}^d \wedge \mathbf{w}_j^d|}{|\mathbf{w}_j^d|}, \tag{4}$$

where $|p| \equiv \sum_{i=1}^{n} p_i$ and $(p \wedge q)_i = \min(p_i, q_i)$. The goal attainment value can then used as the internal reward signal (r) to the reactive module for reinforcement learning and to the intention module for plan evaluation.

4.2. The reactive module

The reactive module is a TD-FALCON model [7,8,14], implemented as a three-channel fusion ART model. It consists of a sensory field F_1^{r1} for representing the current state, a motor field F_1^{r2} for representing the available actions, a feedback field F_1^{r3} for representing the estimated payoff for performing a given action in a particular state, and a cognitive field F_2^r for encoding the relations among the activity patterns across the three input channels. Due to the space constraint, only a brief summary of the TD-FALCON dynamics is given below. Please refer to [8] for a full description of the algorithm.

TD-FALCON incorporates Temporal Difference (TD) methods to estimate and learn value functions of action-state pairs $Q(s,a)$ that indicates the goodness for a learning system to take a certain action a in a given state s. Such value functions are then used in the action selection mechanism, also known as the *policy*, to select an action with the maximal payoff. The TD-FALCON model used in this paper employs a direct code access procedure [33] as shown in Table 1. Given the current state s, TD-FALCON first decides between exploration and exploitation by following an action

Table 1
The reactive module based on TD - FALCON with direct code access.

1. Initialize the FALCON network.
2. Sense the environment and formulate the sensory state s.
3. Following an action selection policy, choose between exploration and exploitation.
   If exploring, take a random action.
   If exploiting, identify the action a with the maximal Q(s,a) value by presenting the state vector **S**, the action vector $\mathbf{A}=(1,\ldots 1)$, and the reward vector $\mathbf{R}=(1,0)$ to FALCON.
4. Perform the action a, observe the next state s', and receive a reward r (if any) from the environment.
5. Estimate the revised value function $Q(s,a)$ following a Temporal Difference rule $\Delta Q(s,a)=\alpha TD_{err}(1\text{-}Q(s,a))$, where $TD_{err}=r+\gamma\max_{a'}Q(s',a')\text{-}Q(s,a)$, of which r is the immediate reward value, $\gamma\in[0,1]$ is the discount parameter, and $\max_{a'}Q(s',a')$ denotes the maximum estimated value of the next state s'.
6. Present the state, action, and reward (Q-value) vectors (**S**, **A**, and **R**) to FALCON for learning.
7. Update the current state by $s=s'$.
8. Repeat from Step 3 until s is a terminal state.

selection policy. For exploration, a random action is picked. For exploitation, TD-FALCON searches for an optimal action through a direct code access procedure. Upon receiving a feedback from the environment after performing the action, a TD formula is used to compute a new estimate of the Q value of performing the chosen action in the current state. The new Q value is then used as the teaching signal for TD-FALCON to learn the association of the current state and the chosen action to the estimated Q value.

4.3. The intention module

In a typical BDI system, plans are stored in a plan library or repository. Each plan p can be represented as a three-tuple of context, goal, and actions as follows:

$$p=\langle \mathcal{C},\mathcal{G},\mathcal{E}\rangle, \tag{5}$$

where $\mathcal{C}$ serves as the context of executing the plan, $\mathcal{G}$ refers to the goal which the plan is supposed to achieve, and $\mathcal{E}$ is the sequences of steps in carrying out the plan.

In our intention module, plans are learned and encoded via a four-channel fusion ART network consisting of a cognitive field F_2^i and four input fields. The input fields are namely the context field F_1^{i1} for representing the current state, the goal field F_1^{i2} for representing the target states, the action sequence field F_1^{i3} for representing the sequence of actions to be performed under a selected plan, and the evaluation field F_1^{i4} for representing the estimated payoff (or confidence) of using the plan. Each node in the cognitive field F_2^i encodes a plan associating a sequence of actions to achieve a given target goal state from the current state.

Input vectors: Let **C** denote the input context vector to F_1^{i1}, where **C** is typically a part of or can be derived from the state vector **S**, acting as a context in which the plan is applicable. Let **G** denote the input goal vector to F_1^{i2}. Let **E** denote the input action sequence vector to F_1^{i3}. Let **cf** denote the input confidence vector to F_1^{i4}.

Activity vectors: Let $\mathbf{x}^{ik}$ denote the F_1^{ik} activity vector. Let $\mathbf{y}^i$ denote the activity vector of F_2^i.

Parameters: The fusion ART's dynamics is determined by choice parameters $\alpha^{ik}\geq 0$, learning rate parameters $\beta^{ik}\in[0,1]$, contribution parameters $\gamma^{ik}\in[0,1]$ and vigilance parameters $\rho^{ik}\in[0,1]$ for each pattern field F_1^{ik}.

The key processes in the intention module include plan selection, plan execution, plan evaluation, and plan learning. The detailed algorithms, as summarized in Table 2 and depicted using the flowchart in Fig. 3, are described as follows.

Table 2
The intention module based on four-channel fusion ART with direct code access.

1. Initialize the 4-channel fusion ART network.
   Formulate goal vector **G** representing the target goal state.
2. Formulate context vector **C**, which can be a part of the sensory vector **S**.
3. Select a plan p with the maximal confidence by presenting the context vector **C**, the goal state vector **G**, the action sequence vector $\mathbf{E}=(1,\ldots 1)$, and the confidence $\mathbf{cf}=(1,0)$ to the 4-channel fusion ART.

4.1. When a plan is selected, carry out plan p according to the action sequence vector.
   Observe the final state s', and receive a reward r (if any) from the environment.
   Estimate the revised confidence of the plan following the equation.
   Present context vector **C**, goal state vector **G**, action sequence vector **E**, and confidence vector **cf** to fusion ART for learning.
   A plan with a lower confidence than the threshold is removed from the repository.

4.2. If no plan is selected, the intention module produces no action output.
   When the reactive module discovers a sequence of actions leading to a goal state, present context vector **C**, goal state vector **G**, action sequence vector **E**, and confidence vector **cf** to the intention module for learning of new plan.

5. Repeat from Step 2 until the goal state is reached.

Plan selection: The plan selection process follows the code selection strategy as used in fusion ART systems. Given the activity vectors initialized by $\mathbf{x}^{i1}=\mathbf{C}$, $\mathbf{x}^{i2}=\mathbf{G}$, $\mathbf{x}^{i3}=\mathbf{E}=(1,\ldots,1)$, and $\mathbf{x}^{i4}=\mathbf{cf}=(1,0)$, for each F_2^i node j, the choice function T_j^i is computed by

$$T_j^i=\sum_{k=1}^{4}\gamma^{ik}\frac{|\mathbf{x}^{ik}\wedge\mathbf{w}_j^{ik}|}{\alpha^{ik}+|\mathbf{w}_j^{ik}|}. \tag{6}$$

A code competition process follows under which the F_2^i node with the highest choice function value is identified.

Plan matching: Before the winning plan can be adopted, a matching process takes place to ensure that the selected plan is a good match according to the template matching criteria. Specifically, resonance occurs if for each channel k, the *match function* m_J^{ik} of the chosen node J meets its vigilance criterion:

$$m_J^{ik}=\frac{|\mathbf{x}^{ik}\wedge\mathbf{w}_J^{ik}|}{|\mathbf{x}^{ik}|}\geq\rho^{ik}. \tag{7}$$

If any of the vigilance constraint is violated, a mismatch reset occurs and the intention module becomes inactive. Otherwise, the system adopts the selected plan and interprets it for execution as described below.

Plan execution: A readout process first takes place, under which the action sequence, denoted by $\mathbf{x}^{i3(\text{new})}$, and the corresponding confidence value, denoted by $\mathbf{x}^{i4(\text{new})}$, of the selected plan J can be obtained by

$$\mathbf{x}^{i3(\text{new})}=\mathbf{x}^{i3(\text{old})}\wedge\mathbf{w}_J^{i3} \tag{8}$$

and

$$\mathbf{x}^{i4(\text{new})}=\mathbf{x}^{i4(\text{old})}\wedge\mathbf{w}_J^{i4}, \tag{9}$$

respectively.

Plan execution then reads out an action at a time from the action sequence field (i.e., $\mathbf{x}^{i3}$). The encoded actions of the adopted plan are then executed through the motor channel of the reactive module.

Plan evaluation: When an adopted plan leads to a positive outcome, the confidence of the plan is increased. On the other hand, if the adopted plan leads to a negative outcome, the confidence is reduced accordingly. Specifically, the confidence of plan j (c_j) is updated using the following formula:

$$\Delta c_j=\delta(1\text{-}c_j)r\text{-}\delta c_j(1\text{-}r), \tag{10}$$

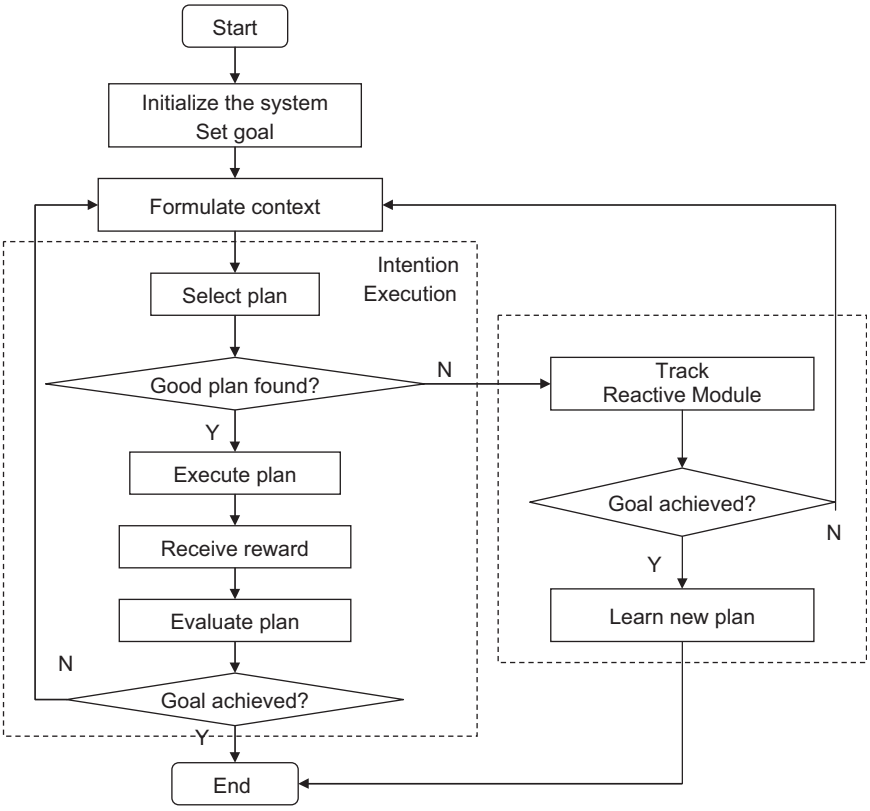

Fig. 3. The system flow of the intention module.

where δ is the plan confidence learning rate and r is the reward signal evaluated by the desire module.

Upon the estimation of the new confidence, the context vector **C**, the goal state vector **G**, the action sequence vector **E**, and the confidence vector $\mathbf{cf} = (c_j, 1\text{-}c_j)$ are presented to fusion ART for learning. When the confidence value of a plan drops below a certain threshold, the plan can be pruned from the intention module.

Plan learning: When the intention module is inactive, the actions performed by the reactive module is recorded by the intention module in the action sequence field. If and when the reactive module discovers a (new) sequence of actions that leads to a target goal state, the input vectors **C**, **G**, **E**, and $\mathbf{cf} = (1, 0)$ are presented to the intention module for learning.

4.4. Integrating intentional and reactive action execution

With the presence of both deliberative and reactive capabilities, a strategy is needed for combining the action sequences and the actions selected by the intention module and the reactive module respectively. We experiment with two strategies, namely the "follow-through strategy" and the "re-evaluation strategy" as follows:

Follow-through strategy: As illustrated in Fig. 3, the simplest strategy is to adopt a follow-through strategy. After a plan is selected for execution, the action sequence of the plan is executed from the beginning to the end. In other words, the agent follows through the entire sequence of actions before it performs another round of sensing again. The behaviour of the overall cBDI-FALCON system with the follow-through strategy can be summarized into the key steps as shown in Table 3. This represents a bold agent.

Table 3
Dynamics of the cBDI-FALCON agent with the follow-through strategy.

1. Initialize the agent's system state in the desire, intention, and reactive modules.
   In particular, initialize the goal state vector **G**.
2. Sense the environment and formulate sensory vector **S** and context vector **C**.
3. Present the input vectors to the intention and reactive modules simultaneously.

4.1. Intention Execution:
   If the intention module finds a good plan, adopt the sequence of actions encoded.
   When the action sequence is completed, obtain feedback, compute goal attainment function, and perform plan learning.
   If not end of trial, go to step 2.

4.2. Reactive Execution:
   When the intention module is inactive, adopt the action selected by the reactive module and record the selected action into the action sequence vector.
   After performing the action, obtain feedback, compute the goal attainment function for reactive learning.
   If end of trial and a goal state is reached, the intention module is activated for learning of new plan.
   If not end of trial, go to step 2.

Re-evaluation strategy: Alternatively, a cautious agent can be modelled using a re-evaluation strategy. This strategy is similar to the follow-through strategy, except that an extra step of sensing is performed half-way through the execution of the plan. This enables the agent to evaluate the current plan for its usability by comparing the next action (from the action sequence) with the action selected by the FALCON's direct access method. The adopted plan continues if the next action of the plan coincides with the action selected by the reactive module.

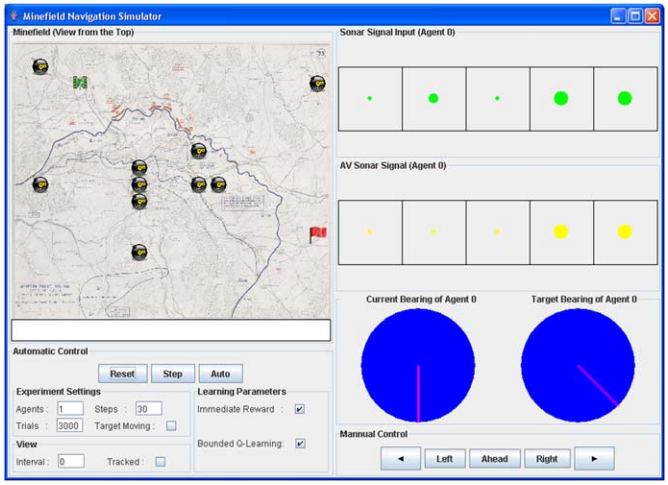

Fig. 4. The minefield navigation simulator.

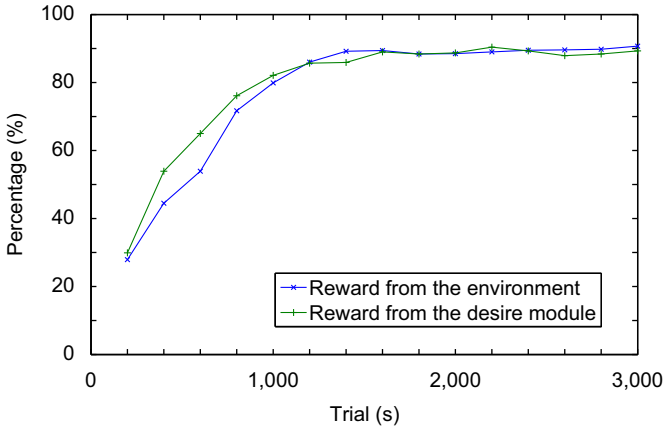

Fig. 5. The success rates of cBDI-FALCON over 3000 trials using reward signal generated internally.

The overall behaviour of the system is similar for the two plan adoption strategies. In contrast to the traditional sense-act-learn cycle, the system now performs a sensing and follows either a plan selected from the intention module or an action selected by the direct code access procedure of the reactive module. The preference of following a plan is higher than that of executing an action selected by the low level reactive module for any trial in the experiments. This principle is consistent with the one used for the behaviour of the Subsumption architecture [34], wherein the outputs of high level modules subsume those of low level modules. After performing a plan, evaluation of plan proceeds within the intention module. When a suitable plan is not identified, the intention module becomes inactive and the system falls back on the reactive module to select the next action. When a new sequence of actions leading to a goal state is discovered by the reactive module, it is learned as a new plan by the intention module.

5. Case study on minefield navigation

The minefield navigation task (Fig. 4) requires an autonomous vehicle (AV) starting at a randomly chosen position in the field to navigate through the minefield to a randomly selected target position in a specified time frame without hitting a mine. A trial ends when the system reaches the target (success), hits a mine (failure), or runs out of time.

Minefield navigation and mine avoidance is a non-trivial task. As the configuration of the minefield is generated randomly and changes over trials, the system needs to learn strategies that can be carried over across experiments. In addition, the system has a rather coarse sensory capability with a 180° forward view based on five sonar sensors. For each direction i, the sonar signal is measured by $s_i = 1/d_i$, where d_i is the distance to an obstacle (that can be a mine or the boundary of the minefield) in the i direction. Other input attributes of the sensory (state) vector include the range and the bearing of the target from the current position. In each step, the system can choose one out of the five possible courses of action, namely MoveLeft, MoveFrontLeft, MoveFront, MoveFrontRight, and MoveRight.

5.1. Goal representation and matching

In contrast to traditional reinforcement learning agents which have no explicit representation of goals, the cBDI-FALCON architecture provides an explicit implementation of goals and goal attainment evaluation. For the minefield problem, we define two main goals as follows.

- Goal 1: Agent reaches the target.
- Goal 2: Agent maintains its life value at the maximum.

Based on the two goals defined, the goal target vectors consist of two key attributes, namely *distance* indicating the remaining distance towards the target and *life* indicating the energy level of the agent. Using complement coding, the goal state vector $\mathbf{x}^d$ can thus be defined as

$$\mathbf{x}^d = (D, \overline{D}, L, \overline{L}),$$

where D and L are the normalized values of the remaining distance and the life value, with $\overline{D}$ and $\overline{L}$ as their complements respectively.

For goal 1, the corresponding goal target vector has a value of 0 for the first element, indicating that the desired remaining distance to the target is zero. The second element, as the complement of the first value, is set to 1. The rest of the elements not relevant to the goal are set to 0. Similarly for goal 2, the goal target vector assigns a value of 1 for the third element, indicating a maximum life value. The other elements are set to 0. The two goal target vectors are thus given by

$$\mathbf{w}_1^d = (0.0, 1.0, 0.0, 0.0)$$

and

$$\mathbf{w}_2^d = (0.0, 0.0, 1.0, 0.0). \quad (11)$$

In each reaction cycle, the states of the goal attributes can be obtained as follows: The distance to the target can be retrieved through the sensory input signals supplied by the maze. The life value is initialized to the user defined value and decremented by a certain value each time the agent runs into a mine. The raw values of distance and life are then normalized to the range of [0,1] before assigning to the goal state vector. By matching the goal state vector and the goal target vectors, the degree of goal attainment is computed using the goal attainment function (Eq. (4)) as described in Section 4.

Based on degree of goal attainment computed internally, we experimented cBDI-FALCON and compared its performance with that of the original TD-FALCON using reward signals provided externally. The experimental results are obtained by averaging over 30 runs consisting of 3000 trials each. Referring to Fig. 5, we can see that cBDI-FALCON using the degree of goal attainment computed by its desire module achieves the same level of

performance as its counterpart using external rewards. More importantly, the desire module has enabled the cBDI-FALCON system a flexible way of defining and switching goals as well as a higher level of self-awareness than a pure reinforcement learner that reacts purely based on external reward signals.

Note that in this work, we do not compare the performance of cBDI-FALCON with other reinforcement learning systems. This is in view that TD-FALCON has been shown to produce superior learning capabilities, compared with many state-of-the art gradient descent based reinforcement learning systems in various benchmark experiments, including minefield navigation [8,14].

5.2. Plan learning and adoption

We set out to study how plans are learnt and used in the minefield navigation task. By varying the values of the plan vigilance parameter ρ^{i1}, we evaluate the system in terms of success rates and the number of plans created and utilized. The experimental results are obtained by averaging over 30 runs of 3000 trials with the value of ρ^{i1} varying from 0.7 to 1.0. In all experiments, the initial plan confidence is set to 1.0. We experimented with cBDI-FALCON with and without plan pruning. With plan pruning, the threshold for removing a plan is exactly the same as the value of plan confidence vigilance ρ^{i4}, which is initialized to be 0.9. The fusion ART parameters in the intention module are further set accordingly as: $\alpha^{i1}=\alpha^{i3}=0.1$, $\alpha^{i2}=\alpha^{i4}=0.001$ and $\gamma^{ik}=1.0$ for $k=1,\ldots,4$.

As shown in Fig. 6, with a plan vigilance of 1.0, the system performance is observed to be almost the same as the original TD-FALCON system (without the use of plans). These results are encouraging considering that prior experiments on the same minefield domain using the PGS system [23] actually found a slight degradation in performance, compared with the original reactive FALCON system. In addition, when the re-evaluation strategy is used with a plan vigilance of 0.99, the system performance is still quite close to that of the original TD-FALCON. In all experiments, we note that the success rates of TD-FALCON and cBDI-FALCON do not improve further significantly beyond 3000 trials. This is largely due to the limitation of the AV's coarse sensory capability and the epsilon-decay schedule used in the action selection policy. The detailed performance of cBDI-FALCON at the 3000th trial, in terms of success rates, hit-mine rates and out of time rates, is summarized in Fig. 7.

As the plan vigilance value decreases, we do notice a gradual drop in the success rates of cBDI-FALCON, although the re-evaluation strategy proves to be more resilient than the follow-through strategy. With the follow-through strategy and a plan vigilance of <0.99, the success rates of cBDI-FALCON drop to be

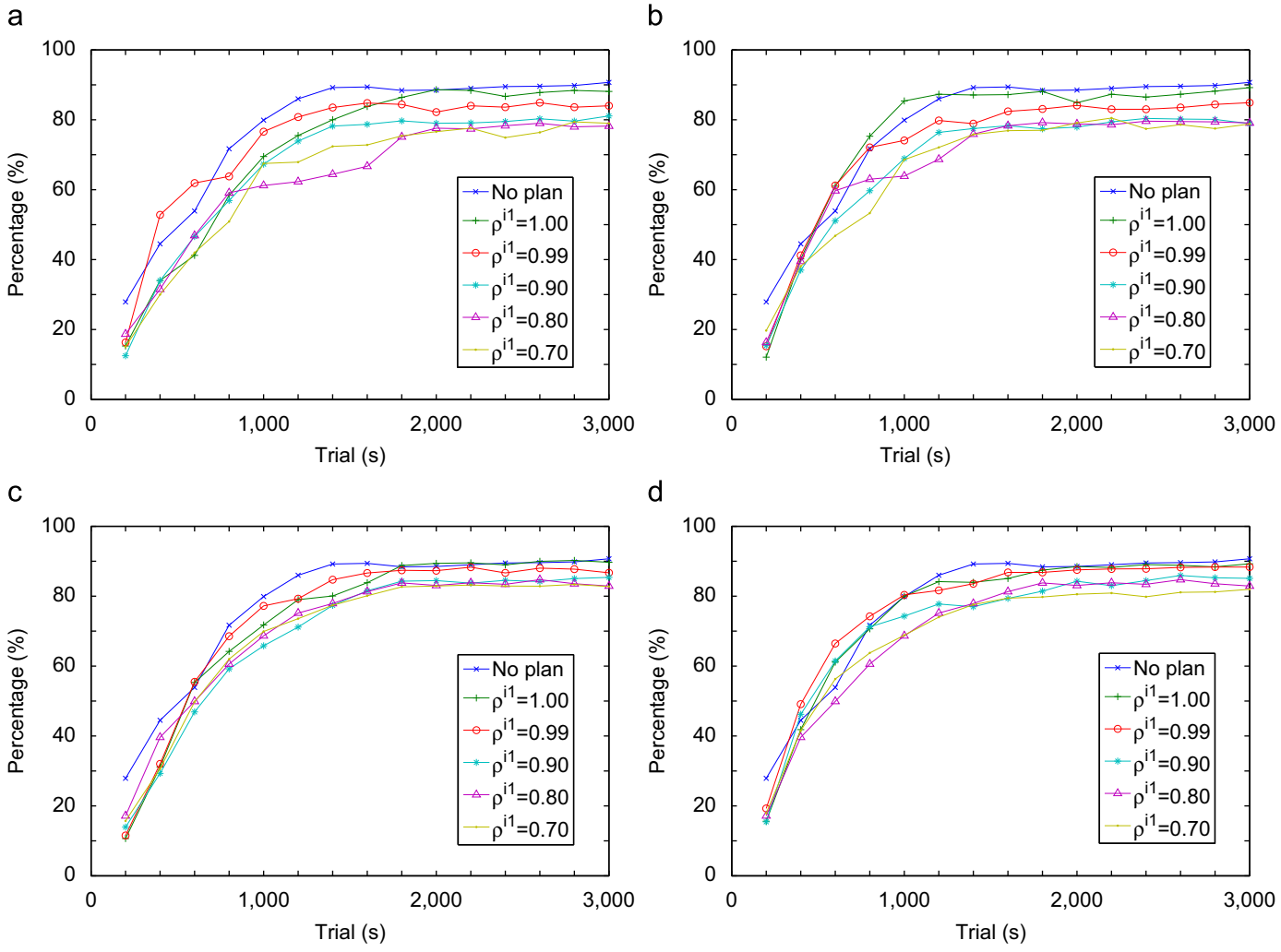

Fig. 6. The success rates of cBDI-FALCON over 3000 trials using different plan vigilance values. (a) Follow-through without plan pruning. (b) Follow-through with plan pruning. (c) Re-evaluation without plan pruning. (d) Re-evaluation with plan pruning.

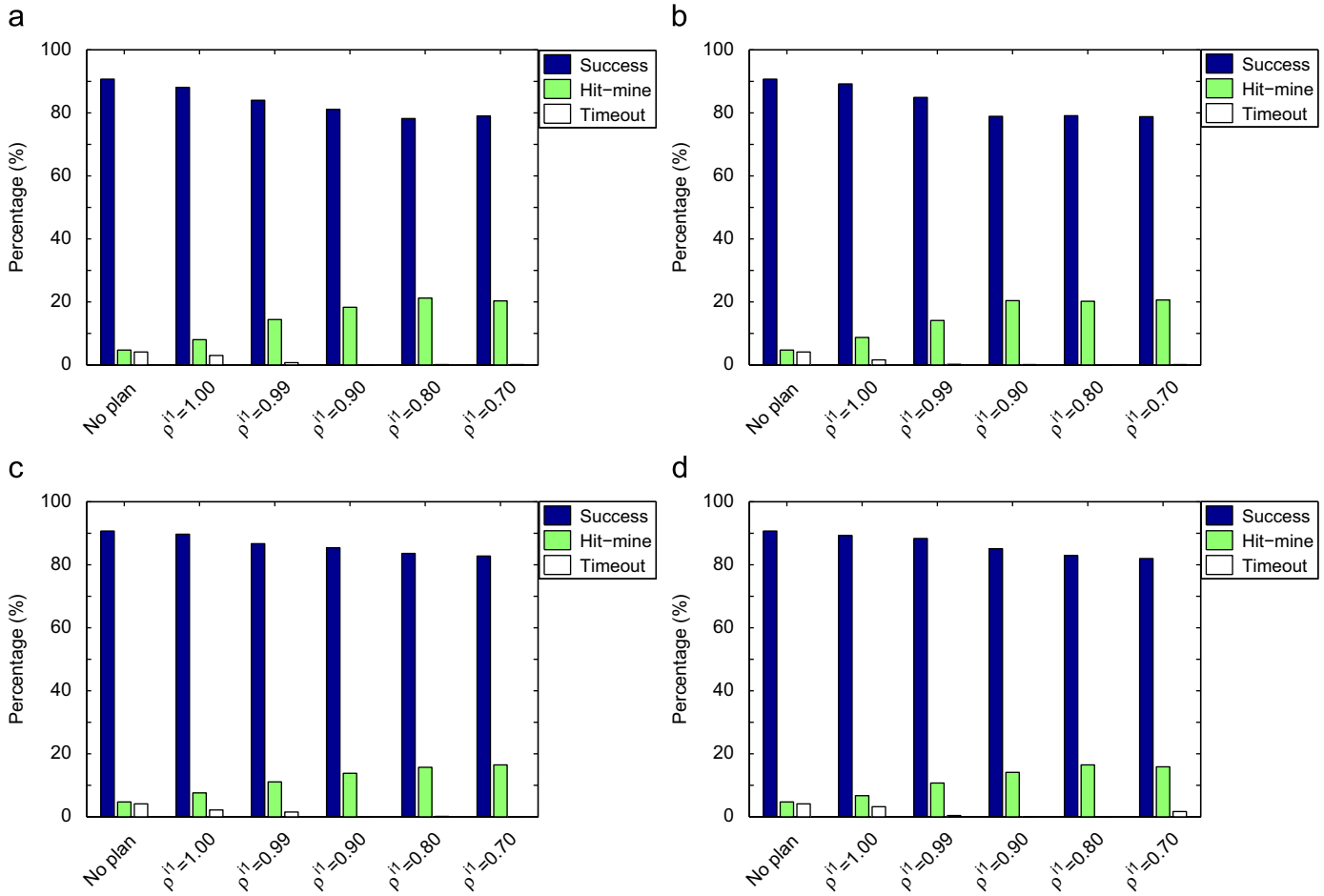

Fig. 7. The performance of cBDI-FALCON after 3000 trials using different plan vigilance values. (a) Follow-through without plan pruning. (b) Follow-through with plan pruning. (c) Re-evaluation without plan pruning. (d) Re-evaluation with plan pruning.

around 80%. In comparison, the success rates are around or above 85% at the 3000th trial for the re-evaluation strategy. The better performance indicates that re-evaluation of adopted plans enables cBDI-FALCON to affirm whether a selected plan can lead to a positive outcome during runtime and helps to maintain a more robust system performance.

Comparing cBDI-FALCON systems with and without the plan pruning mechanism, we do not find a significant difference in terms of the overall performance, although rule pruning seems to help marginally when the plan vigilance is low. In contrast, the effect of rule pruning is much more profound in terms of plan utilization as described below.

To understand the system behaviour in learning and using plans, we examine the number of plans learned in the above experiments . As shown in Fig. 8(a), the number of plans learned by the intention module is proportional to the plan vigilance parameter value. This is not surprising as a high plan vigilance increases the likelihood for existing plans to be rejected, raising the chance of learning a new plan. In addition, the rate of reduction seems to increase when the plan vigilance decreases, though the reduction rate of the re-evaluation strategy is lower than that of the follow-through strategy. For both strategies, rule pruning on the whole helps to reduce the number of learned plans, but the effect is only significant with a low plan vigilance value.

We further study how the learned plans are used in making decision by evaluating plan utilization rate, namely the percentage of the plans used out of the entire population of plans learned by the intention module. It is a measure of how extensive the available plans are tapped. As shown in Fig. 8(b), the plan utilization rate increases as the plan vigilance value decreases to a certain value. This is expected as the plan vigilance parameter determines the level of similarity a selected plan needs to be applicable. A higher vigilance would mean a lower chance for the plan to be used. As the plan vigilance decreases to a certain level, the possibility for a selected plan to lead to failure also increases (see Fig. 6), and thus the confidences for those plans are decreased accordingly. As a consequence, a smaller number of plans would be applicable. However, with the plan pruning strategy, plans with a low confidence are removed from the network. As the number of learned plans drops, the rate of plan utilization thus increases as the plan vigilance decreases.

Based on the above results, the re-evaluation strategy with a plan vigilance of 0.99 appears to produce the most desirable result as it yields a comparable performance with the original TD-FALCON in terms of success rates and at the same time, provides a reasonable level of plan utilization at around 40%.

To examine the effect of realizing the intention module using fusion ART, we compare cBDI-FALCON with the original BDI-FALCON system [6] in terms of the number of the plans learned and adopted. As shown in Fig. 9, for a comparable success rate of around 90%, the BDI-FALCON system with a plan vigilance of 0.8 generated around 2500 plans, which is more than double the 1000 plans generated by cBDI-FALCON, and achieved a much lower plan utilization rate of 10 to 15%. This clearly shows that the full connectionist realization of the integrated architecture

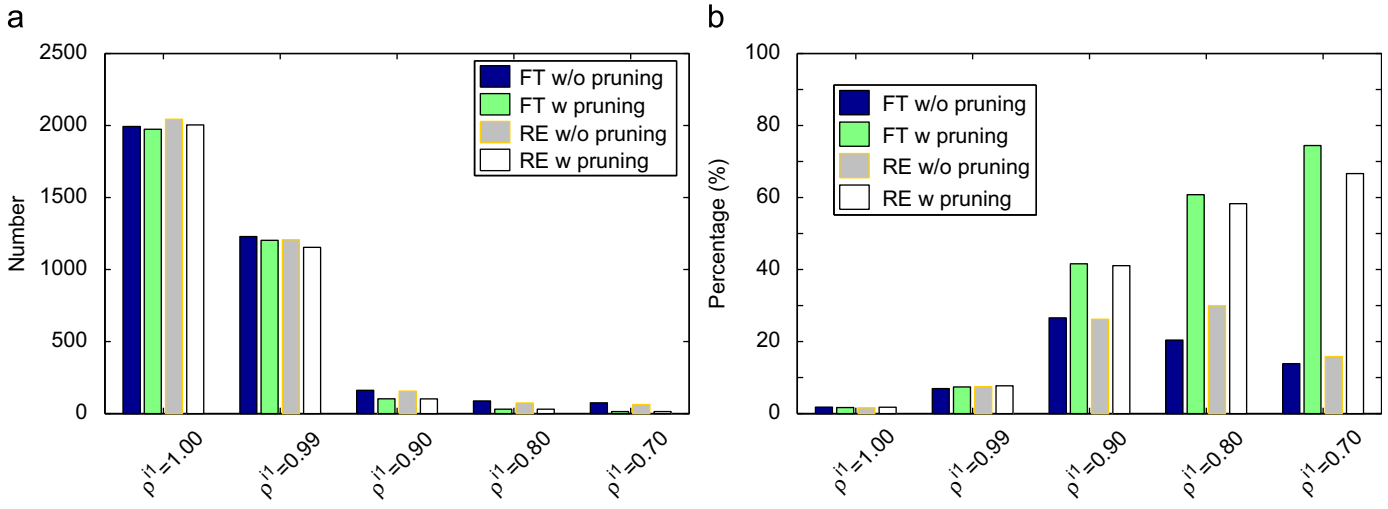

Fig. 8. The number of plans learned and adopted by cBDI-FALCON. (a) Number of learned plans. (b) Plan utilization rate.

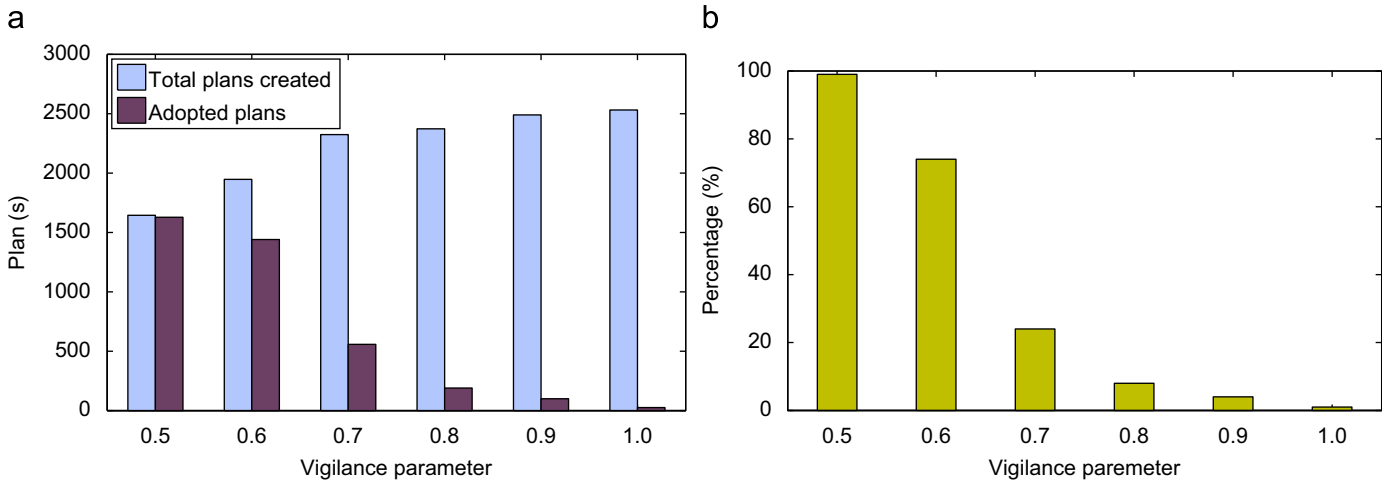

Fig. 9. The number of plans learned and adopted by BDI-FALCON. (a) Number of plans learned and adopted (b) Plan utilization rate.

has improved upon the original system in terms of learning efficiency and plan utilization.

5.3. Plan analysis

Besides success rates and plan utilization, we are interested in how an adopted plan may contribute to the outcome of a trial. At a micro-level, within the set of those plans that lead to successes, we identify how many of them directly lead to the target and how many only contribute partially to a successful path. Consequently, we categorize the types of contribution into four main classes as follows.

1. Plans that directly lead to the target.
2. Plans that contribute partially to the succeeded path.
3. Plans that lead to hit-mine failure.
4. Plans that lead to timeout failure.

As the numbers of plans adopted in the two strategies may not be the same, we compare the plan contribution in the following experiments in terms of the percentages of the plans used. Referring to Fig. 10, within the set of adopted plans, the two strategies have very similar proportions of plans that directly lead to the target but the re-evaluation strategy has a higher percentage of plans with partial contribution towards the target. Correspondingly, among the adopted plans, the re-evaluation strategy has a lower percentage of time-out and hit-mine failures. Therefore, the overall success rates of the re-evaluation strategy are better than those of the follow-through strategy.

5.4. Plan efficiency

An important motivation of using plans is the reduction in the frequency of periodic sensing as in the sense-act-learn cycles. In this section, we compare the two strategies in terms of the number of sensing needed. For the follow-through strategy, there is only one sensing at the beginning of the plan selection process. The sensory inputs are used to choose the most suitable plan for the particular situation represented by those signals. After that, there is no more sensing during plan execution until all actions in the plan have been performed. For the re-evaluation strategy, one sensing at the beginning of the plan selection process is needed in the same way as the first strategy. In addition, another sensing is required half-way through the plan's execution.

As shown in Fig. 11(a), it can be observed that, compared to original FALCON, both strategies introduce more steps leading to success, with the follow-through strategy incurs the most number of steps. However, both strategies reduce the average number of

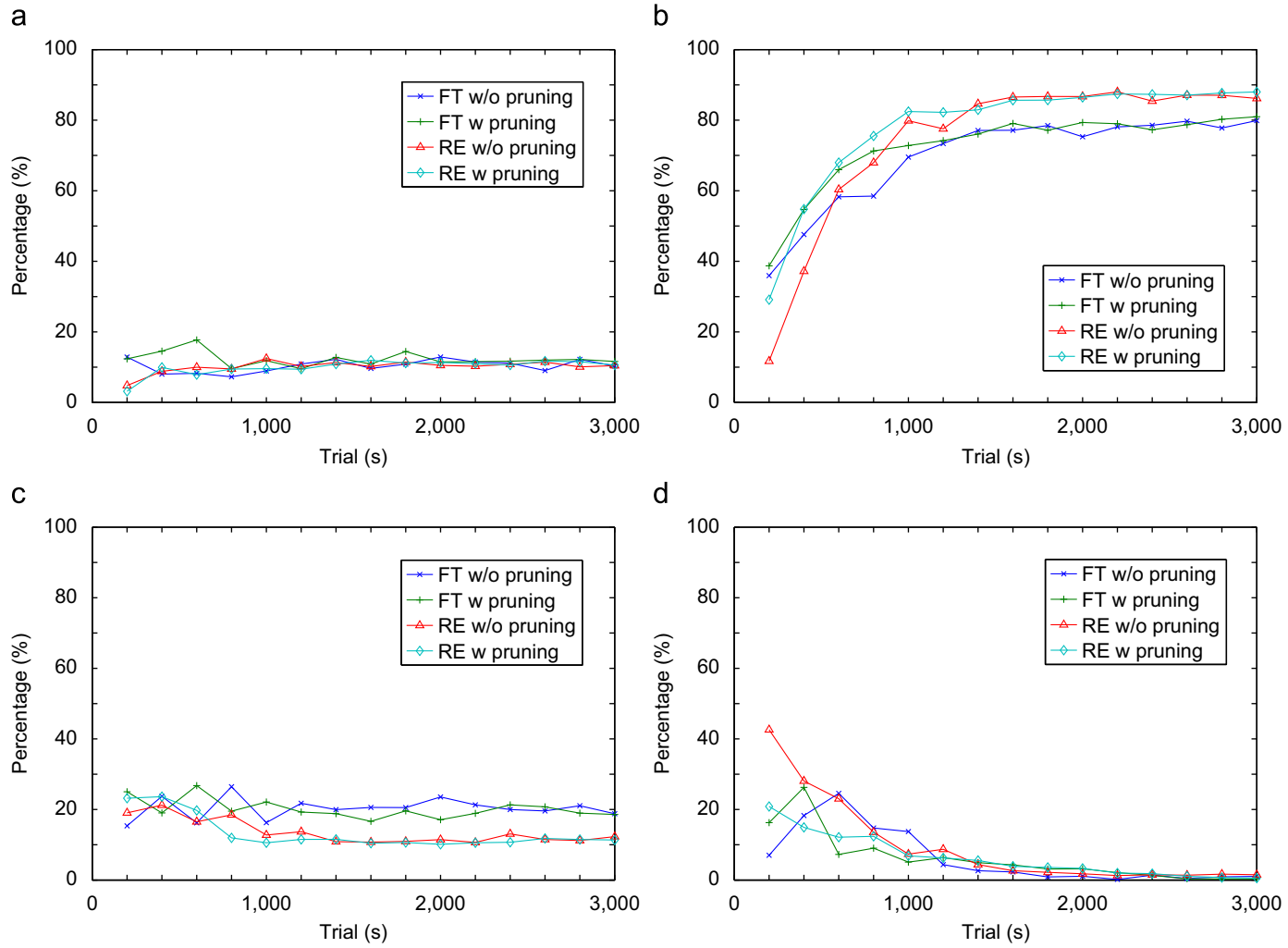

Fig. 10. The proportion of plans leading to various outcomes. (a) Plans leading to success. (b) Plans contributing to success. (c) Plans leading to hit-mine. (d) Plans leading to timeout.

sensing, with the follow-through strategy incurs the least number of sensing, as indicated in Fig. 11(b).

We further explore this aspect by varying the vigilance parameter values. We observe that, a lower plan vigilance can introduce more steps to reach the target. As shown in Fig. 11(c), when the plan vigilance decreases from 0.99 to 0.90, the average number of steps taken by cBDI-FALCON with the follow-through strategy increases from 9.6 to 10.6. However, referring to Fig. 11(d), when the plan vigilance decreases from 0.99 to 0.90, the average number of sensing reduces from 7.3 to 6.5. This implies that, if the agent is meant to be used in a highly noisy environment, where external signals are not reliable or rarely available, lowering the plan vigilance will enable the agent to continue its operation by relying more on the intention module.

6. Conclusion

In view of the complementary strengths of BDI system and machine learning system, hybrid architectures integrating them are getting much research attention. A pure neural network solution, as presented in this paper, has the advantages of parallel processing, unified modelling, and efficient plan learning.

The intention module equips the proposed system with a deliberative planning capability. It thus enables the system to function in a more hostile and dynamic environment, where the sense-act-learn cycle is not always applicable. Following plans also reduces the need for periodic sensing and therefore improves efficiency. We have experimented with two plan adoption strategies, of which the re-evaluation strategy yields better performance, in terms of higher positive contributions among the adopted plans and better overall success rates. It is also superior than the follow-through strategy in reducing the number of sensing cycles.

More significantly, by learning and pruning plans using fusion ART, we show that the number of plans can be significantly reduced especially with a low plan vigilance. With a better plan utilization, our experimental results also show that pruning does not degrade the system's performance in terms of task completion rates.

Moving forward, there remain many challenges in the intention module, especially on the aspects of plan representation and learning. In the current implementation, a plan corresponds to a path towards the target. Learning of exact paths however limits the plan's applicability to very specific situations. Also, due to the space constraint of this paper, we have not worked out the detailed mechanism for encoding temporal sequences using neural activity patterns. Acquiring and executing abstract plan representation through neural processes, as described in STORE working memory model [35], is thus an important research direction. Generalizing a plan into a higher level of abstraction will also make the plans more versatile in handling a wider

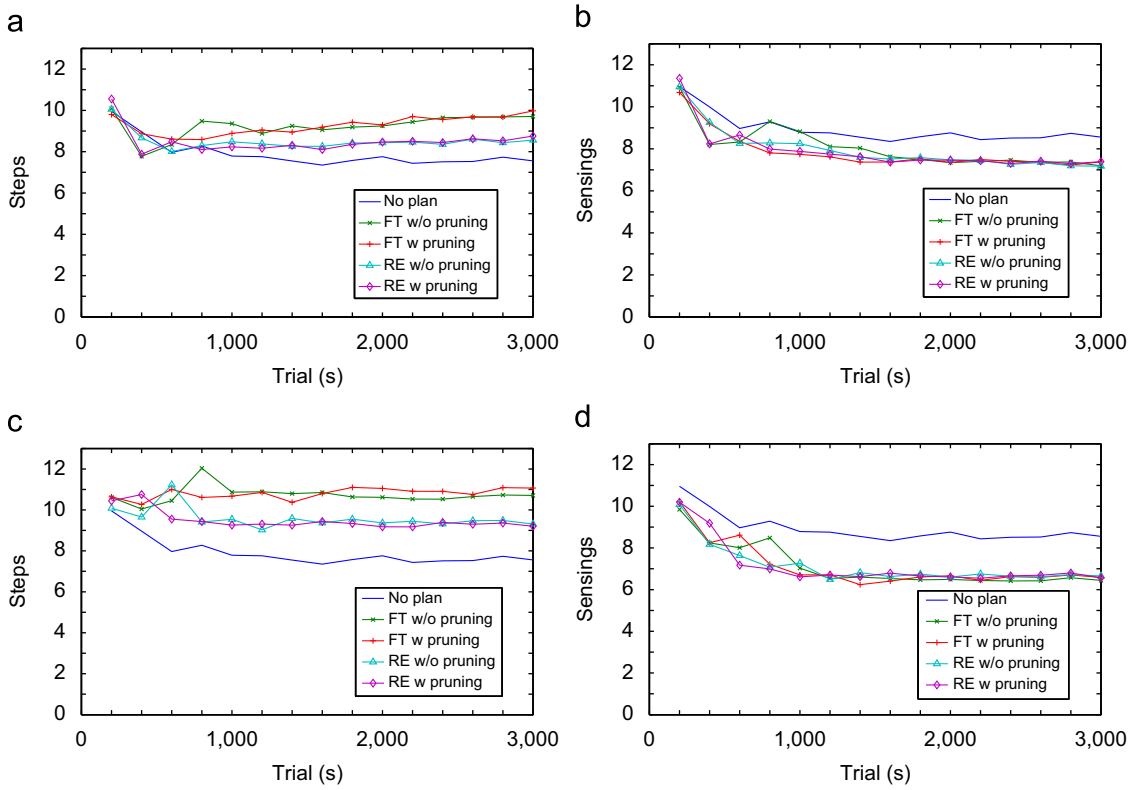

Fig. 11. The number of steps and sensing with $\rho^{i1}=0.99$ and $\rho^{i1}=0.90$. (a) Number of steps with $\rho^{i1}=0.99$. (b) Number of sensing with $\rho^{i1}=0.99$. (c) Number of steps with $\rho^{i1}=0.90$. (d) Number of sensing with $\rho^{i1}=0.90$.

variety of circumstances and keep the size of the plan repository small.

For the desire module, a more sophisticated goal representation will certainly enhance the agent's performance in complex problem solving. Specifically, goal-subgoal hierarchy is one important feature that is missing from our current implementation. By the ability of decomposing a complex goal into simpler subgoals, the latter can be achieved one at a time, laying the path towards the achievement of the primary goal. The concept of goal decomposition will enable the desire module to be more adaptive, in the same way as the reactive and intention modules, by allowing goals to be created, manipulated, and evaluated dynamically.

Working towards self-organizing intentional agents, we have since made progress in the modelling of intentional behaviour, with goal decomposition and abstract plan execution, using self-organizing neural models. Our recent works on intentional planning agents have been reported in [36,37].

Acknowledgements

This work was supported in part by the Singapore National Research Foundation Interactive Digital Media R&D Program, under research Grant NRF2008IDM-IDM004-037. The authors thank Samin Karim, Liz Sonenberg, and Feng Liu for discussion and many invaluable comments to this paper.

References

[1] M. Bratman, D. Israel, M. Pollack, Plans and resource-bounded practical reasoning, Computational Intelligence 4 (4) (1988) 349–355.
[2] R. Sun, Beyond simple rule extraction: the extraction of planning knowledge from reinforcement learners, in: Proceedings of International Joint Conference on Neural Networks, 2000, pp. 24–27.
[3] E. Norling, Folk psychology for human modelling: extending the BDI paradigm, in: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04), 2004, pp. 202–209.
[4] A. Guerra-Hernández, A. E. Fallah-Seghrouchni, H. Soldano, Learning in BDI multi-agent systems, in: Proceedings of the Fourth International Workshop on Computational Logic in Multi-Agent Systems, 2004, pp. 218–233.
[5] C. Lebiere, D. Wallach, Sequence learning in the ACT-R cognitive architecture: empirical analysis of a hybrid model, Proceedings of Sequence Learning—Paradigms, Algorithms, and Applications, Lecture Notes in Computer Science, vol. 1828, Springer, Berlin, 2001, pp. 188–212.
[6] A.-H. Tan, Y.-S. Ong, A. Tapanuj, A hybrid agent architecture integrating desire, intension and reinforcement learning, Technical Report, School of Computer Engineering, Nanyang Technological University, 2008, Submitted for publication.
[7] A.-H. Tan, D. Xiao, Self-organizing cognitive agents and reinforcement learning in a multi-agent environment, in: Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05), 2005, pp. 351–357.
[8] A.-H. Tan, N. Lu, D. Xiao, Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback, IEEE Transactions on Neural Networks 9 (2) (2008) 230–244.
[9] A.-H. Tan, G. Carpenter, S. Grossberg, Intelligence through interaction: towards a unified theory for learning, Proceedings of the Fourth International Symposium on Neural Networks: Advances in Neural Networks, Lecture Notes in Computer Science, vol. 4491, Springer, Berlin, 2007, pp. 1094–1107.
[10] R. Sutton, A. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998.
[11] C. Watkins, P. Dayan, Q-learning, Machine Learning 8 (3–4) (1992) 279–292.
[12] G.A. Carpenter, S. Grossberg, A massively parallel architecture for a self-organizing neural pattern recognition machine, Computer Vision, Graphics, and Image Processing 37 (1987) 54–115.
[13] G.A. Carpenter, S. Grossberg, ART 2: self-organization of stable category recognition codes for analog input patterns, Applied Optics 26 (1987) 4919–4930.
[14] D. Xiao, A.-H. Tan, Self-organizing neural architectures and cooperative learning in multi-agent environment, IEEE Transactions on Systems Man and Cybernetics—Part B 37 (6) (2007) 1567–1580.
[15] K. Schultz, The neural network—teaching computers to think, Computer Power User 2 (1) (2002) 62–63.
[16] D. Gordan, D. Subramanian, A cognitive model of learning to navigate, in: Proceedings of the 19th Annual Conference of the Cognitive Science Society, 1997, pp. 271–276.
[17] R. Sun, C. Sessions, Learning plans without a priori knowledge, Adaptive Behavior 8 (3/4) (2000) 225–254.

[18] P. Wallis, Intention without representation, Journal of Philosophical Psychology 17 (16) (2004) 209–224.
[19] J. Elman, Finding structure in time, Cognitive Science 14 (1990) 179211.
[20] A. Cleeremans, J. McClelland, Learning the structure of event sequences, Journal of Experimental Psychology: General 120 (3) (1991) 235–253.
[21] C. Heinze, S. Goss, A. Pearce, Plan recognition in military simulation: incorporating machine learning with intelligent agents, in: Proceedings of IJCAI Workshop on Team Behaviour and Plan Recognition, 1999, pp. 53–63.
[22] B. Subagdja, L. Sonenberg, Learning plans with patterns of actions in bounded-rational agents, in: Proceedings of International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES 2005), Lecture Notes in Computer Science, vol. 3683, Springer, Berlin, 2005, pp. 30–36.
[23] S. Karim, L. Sonenberg, A.-H. Tan, A hybrid architecture combining reactive plan execution and reactive learning, in: Proceedings of the Ninth Biennial Pacific Rim International Conference on Artificial Intelligence (PRICAI'06), Lecture Notes in Artificial Intelligence, vol. 4099, 2006, pp. 200–211.
[24] A.-H. Tan, FALCON: a fusion architecture for learning, cognition, and navigation, in: Proceedings of the International Joint Conference on Neural Networks, 2004, pp. 3297–3302.
[25] S. Karim, B. Subagdja, L. Sonenberg, Plans as products of learning, in: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'06), 2006, pp. 139–145.
[26] G. Carpenter, S. Grossberg (Eds.), Pattern Recognition by Self-organizing Neural Networks, MIT Press, Cambridge, MA, 1991.
[27] G. Carpenter, S. Grossberg, Adaptive resonance theory, The Handbook of Brain Theory and Neural Networks (2003) 87–90.
[28] J. He, A.-H. Tan, C.-L. Tan, ART-C: a neural architecture for efficient on-line clustering under constraints, in: Proceedings, IJCNN, Honolulu, 2002, pp. 2550–2555.
[29] J. He, A.-H. Tan, C.-L. Tan, Modified ART 2A growing network capable of generating a fixed number of nodes, IEEE Transactions on Neural Networks 5 (3) (2004) 728–737.
[30] A.-H. Tan, Adaptive resonance associative map, Neural Networks 8 (3) (1995) 437–446.
[31] A.-H. Tan, H. Pan, Predictive neural networks for gene expression data analysis, Neural Networks 18 (3) (2005) 297–306.
[32] G.A. Carpenter, S. Grossberg, D.B. Rosen, Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system, Neural Networks 4 (1991) 759–771.
[33] A.-H. Tan, Direct code access in self-organizing neural architectures for reinforcement learning, in: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'07), 2007, pp. 1071–1076.
[34] R. Brooks, Cambrian Intelligence: The Early History of the New AI, MIT Press, Cambridge, MA, 1999 (A Bradford Book).
[35] G. Bradski, G.A. Carpenter, S. Grossberg, Store working memory networks for storage and recall of arbitrary temporal sequences, Biological Cybernetics 71 (1994) 469–480.
[36] B. Subagdja, A.-H. Tan, Planning with iFALCON: towards a neural-network-based BDI agent architecture, in: Proceedings, IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'08), 2008, pp. 231–237.
[37] B. Subagdja, A.-H. Tan, A self-organizing neural network architecture for intentional planning agents, in: Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS'09), 2009, pp. 1081–1088.

Ah-Hwee Tan received his B.S. (First Class Honors) and M.S. degree in Computer and Information Science from the National University of Singapore and Ph.D. in cognitive and neural systems from Boston University. He is currently an Associate Professor and the Head of Division of Information Systems at the School of Computer Engineering, Nanyang Technological University. He was the founding Director of the Emerging Research Laboratory, a research centre for incubating interdisciplinary research initiatives. Prior to joining NTU, he was a Research Manager at the A*STAR Institute for Infocomm Research (I2R), leading the Text Mining and Intelligent Agents research groups. His current research interests include cognitive and neural systems, information mining, machine learning, knowledge discovery, and intelligent agents. He is an editorial board member of Applied Intelligence, a member of ACM and a senior member of IEEE.

Yu-Hong Feng received her B.Sc. and Ph.D. in Computer Science from University of Science and Technology of China and Nanyang Technological University (NTU) respectively. She is now an assistant researcher at the Shenzhen Institute of Advanced Technology (SIAT), CAS, China. She was a post-doctoral fellow in the Hong Kong Polytechnic University and a research assistant in Intelligent Systems Centre (IntelliSys), NTU. Her areas of interest include distributed workflow management, Grid and Cloud computing, machine learning, and pervasive computing.

Yew-Soon Ong received his Bachelors and Masters degrees in Electrical and Electronics Engineering from Nanyang Technology University, Singapore, in 1998 and 1999, respectively. He then joined the Computational Engineering and Design Center, University of Southampton, U.K., where he completed his Ph.D. degree in 2002. Dr. Ong is currently an Associate Professor with the School of Computer Engineering, Nanyang Technological University, Singapore, as well as Director of the Center for Computational Intelligence (C2I). He is co-technical editor-in-chief of Memetic Computing journal, associate editor of IEEE Transactions on Systems, Man and Cybernetics - Part B, International Journal of System Science and Soft Computing Journal. He is also Chair of the Task Force on Memetic Algorithms in the IEEE Computational Intelligence Society Emergent Technology Technical Committee. His current research interests lie in computational intelligence spanning: memetic computing, evolutionary design and grid computing.