This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2664665, IEEE Transactions on Evolutionary Computation

An Evolutionary Transfer Reinforcement Learning Framework for Multi-Agent Systems

Yaqing Hou, Yew-Soon Ong, *Senior Member, IEEE,*, Liang Feng and Jacek M. Zurada, *Life Fellow, IEEE*

***Abstract*—In this paper, we present an evolutionary Transfer reinforcement Learning framework (eTL) for developing intelligent agents capable of adapting to the dynamic environment of multi-agent systems (MAS). Specifically, we take inspiration from Darwin's theory of natural selection and Universal Darwinism as the principal driving forces that govern the evolutionary knowledge transfer process. The essential backbone of our proposed eTL comprises several meme-inspired evolutionary mechanisms, namely *meme representation*, *meme expression*, *meme assimilation*, *meme internal evolution* and *meme external evolution*. Our proposed approach constructs social selection mechanisms that are modeled after the principles of human learning to identify appropriate interacting partners. eTL also models the intrinsic parallelism of natural evolution and errors that are introduced due to the physiological limits of the agents' ability to perceive differences, so as to generate "growth" and "variation" of knowledge that agents have of the world, thus exhibiting higher adaptivity capabilities on solving complex problems. To verify the efficacy of the proposed paradigm, comprehensive investigations of the proposed eTL against existing state-of-the-art TL methods in MAS, are conducted on the "Minefield Navigation Tasks (MNT)" platform and the "Unreal Tournament 2004" first person shooter computer game, in which homogeneous and heterogeneous learning machines are considered.**

***Keywords*—Transfer Learning (TL), Reinforcement Learning (RL), Multi-Agent Systems (MAS), Natural Selection, Memetic Automaton.**

I. Introduction

MUTLI-AGENT systems (MAS) are computerized systems composing of multiple interacting and autonomous agents within an environment of interests for problem-solving. MAS have a wide array of applications in various industrial engineering and scientific fields, such as resource management [1], collaborative decision support systems [2], computer games [3], etc. In conventional MAS, the actions of each agent are usually pre-defined for the states that would be encountered in the environment. However, this approach can be unreliable when the environment becomes complex and changes over time, since the space of environmental states is enormous and hence manually defining these action-state mappings is infeasible. Thus it is necessary to endow the agents with intelligence capable of adapting to the dynamic environment.

In the literature, reinforcement learning (RL) has been proposed as the learning process of agents through trial-and-error interactions in a dynamic environment. RL plays an important role for building intelligent and autonomous agents. In the last decades, RL has attracted significant attentions, and a plethora of RL methodologies have been proposed. Generally, these RL approaches can be divided into two core categories. The first focuses on search in the action-state mapping spaces in order to find the optimum mappings that perform well in the problem of interest. Search methods such as genetic algorithm [4], genetic programming [5] and simulated annealing [6], have been commonly employed in this class of RL approaches. The second category is to estimate the utility function of taking an action in states of the given problem domain via statistical techniques or dynamic programming methods, such as TD(λ) [7] and Q-learning [8]. To date, RL has been successfully applied in many real-world complex applications, including autonomous helicopter [9], humanoid robotics [10], autonomous vehicles control [11] [12], etc.

A more recent machine learning paradigm of transfer learning (TL) has been introduced as an approach of leveraging valuable knowledge from related and well studied problem domains to enhance problem-solving in the target domains of interest. TL has been successfully used for enhancing RL tasks [13]. A variety of transfer learning methodologies, such as instance transfer [14], action-value transfer [15], feature transfer [16] [17] and advice (rule) exchanging [18], have been introduced. Recently, research on TL has also been considered in the multi-agent RL tasks. Boutsioukis *et al.* [19] focused on evaluating the applicability of transfer to multi-agent RL and discussed transferring from single-agent system to multi-agent system and from one MAS to another, in an offline manner. Nevertheless, most of the aforementioned research focuses exclusively on sequential knowledge transfer as highlighted in [13], where learning in the source task must happen prior to learning in the target task. In MAS, on the other hand, since agents interact with one another, the potential for multiple agents to learn from one another in the same environment simultaneously or online should be appropriately exploited. The challenge here is thus to identify suitable source tasks from multiple agents that will contain useful information to transfer [20].

To address the issues of requiring source information and selecting source tasks, Oliveira *et al.* [21] described a study on an interactive Advice-Exchange (AE) mechanism, wherein

Yaqing Hou is with the Interdisciplinary Graduate School, Nanyang Technological University (NTU), Singapore 639798 (e-mail: houy0003@e.ntu.edu.sg).

Yew-Soon Ong is with the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore 639798 (e-mail: asysong@ntu.edu.sg).

Liang Feng is with the College of Computer Science, Chongqing University, China (e-mail: liangf@cqu.edu.cn).

Jacek Zurada is with the Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY 40292, USA (e-mail: jacek.zurada@louisville.edu), and also with the University of Social Science, 90-113, Łódź, Poland.

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2664665, IEEE Transactions on Evolutionary Computation

agents with poor performance seek advice only from the elitist for the next action to take. One major problem observed is that the bad advice, or blind reliance, of the elitist, could hinder the learning process, sometimes even beyond recovery. Similarly, Feng *et al.* [22] proposed to select teacher agents which not only have superior performance but also share similar past experiences. However, as the experience similarity measure is model specific, the approach is likely to fail when agents of diverse models exist. More recently, Taylor *et al.* [20] also discussed some key issues such as when and what to transfer, for conducting TL in MAS while learning progresses online and further proposed a broadcast-based Parallel Transfer learning (PTL) for MAS. In PTL, each agent will broadcast its knowledge to all the other agents while deciding whose knowledge to accept based on the reward received from other agents versus the expected rewards it predicts. Nevertheless, agents in this approach tend to infer incorrect actions on unseen circumstances.

In this paper, we propose an evolutionary Transfer reinforcement Learning framework (eTL) for building intelligent agents in MAS. Specifically, we take inspiration from Darwin's theory of natural selection and Universal Darwinism [23] [24] as the principal driving forces that govern the evolutionary knowledge transfer process. The intrinsic parallelism of natural evolution and the errors which are introduced due to the physiological limits of the agents' ability to perceive differences [25], could generate the "growth" and "variation" of knowledge that agents have of the world [26], thus exhibiting higher adaptivity capabilities on solving complex and non-trivial problems. Particularly, the essential backbone of our proposed framework comprises of memetic automaton[1] [27], which includes meme-inspired[2] evolutionary mechanisms, namely *meme representation*, *meme expression*, *meme assimilation*, *meme internal evolution* and *meme external evolution*. In contrast to existing works on TL in MAS, eTL addresses the aforementioned limitations (e.g., blind reliance) found in the current knowledge transfer process of existing frameworks. In particular, the proposed approach constructs social selection mechanisms that are modeled after the principles of human-learning [28] to effectively identify appropriate interacting partners, thus bringing about improved agent social learning capabilities in dynamic environments [29].

The core contributions of the proposed work are summarized as follows:

1) A novel evolutionary multi-agent Transfer reinforcement Learning framework (eTL) is proposed for modeling intelligent social RL agents. Further, beyond the formalism of memetic automaton, the core aspects of eTL including *meme representation*, *meme expression*, *meme assimilation*, *meme internal evolution* and *meme external evolution* are proposed and discussed comprehensively.
2) To explore the generality of eTL, the Fusion Architecture for Learning and Cognition (FALCON) [11] based on adaptive resonance theory (ART) [30] and the classical multi-layer perceptron (MLP) [8] with gradient descent based back propagation (BP) [31] are investigated as the online learning machines that form the brains of the intelligent agents.
3) To validate the effectiveness of eTL with both homogenous and heterogeneous learning agents, comprehensive empirical studies are conducted on commonly used Minefield Navigation Tasks (MNT) [32]. Experimental results obtained show that eTL which transfers knowledge in an evolutionary manner has significantly better performance than existing state-of-the-art TL approaches including AE and PTL.
4) Last but not least, a well-known first person shooter game, namely "Unreal Tournament 2004" (UT2004) [33] [34], is used to investigate the performance of our proposed eTL under complex problem solving scenarios.

The rest of this paper is organized as follows: Section II begins with an introduction of the proposed eTL. Particularly, details of the evolutionary operators in the eTL framework are presented. Further, study of the eTL with FALCON and MLP-BP are presented in Section III. Section IV presents a comprehensive empirical study of eTL based on the popular MNT and UT2004 as complex problems for verifying the performance of eTL. Finally, we conclude this paper with some brief remarks in Section V.

II. Evolutionary Memetic Multi-Agent Transfer Reinforcement Learning System

This section begins with a brief introduction of memetic science which is followed by an introduction of the proposed evolutionary multi-agent Transfer Learning framework (eTL).

A. Memetic Science

The term "meme" can be traced back to Dawkins in his book "The selfish Gene", where he defined it as "a unit of information residing in the brain and is the replicator in human cultural evolution" [23]. While some believe that memes should materialize as information restricted to the brain, others think that the concept extends to behaviors and artifacts. In the book of "The Meme Machine" [35], Blackmore reaffirmed meme as information copied from one person to another and discussed on the theory of "memetic selection" as the survival of the fittest among competitive ideas down through generations. Other researchers on the other hand have looked upon memes in different ways, from being "contagious information patterns that replicate by parasitically infecting human minds" [36], "constellation of activated neuronal synapses in memory or information encoded in neural structure" [37] to "ideas, the kind of complex idea that forms itself into a distinct memorable unit" [38], and even "genotype as mental representation, and phenotype as implemented behavior or artifact" [39].

For the past few decades, the meme-inspired science of memetics [40] [41] has attracted increasing attention and

[1] A memetic automaton is a software entity that autonomously acquires increasing level of capability and intelligence through embedded memes learnt independently or via interactions.

[2] A meme is defined as the basic unit of cultural information, hold in an individual's memory, which is capable of being transmitted to others [23].

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2664665, IEEE Transactions on Evolutionary Computation

stretched across multi-faceted fields including anthropology, biology, psychology, sociology, etc. In the field of computer science, the meme-inspired computing methodology, or more concisely memetic computation, has also surfaced as an intriguing focus of research. Particularly, one of the most direct and simplest applications of meme-inspired technologies in problem solving has been the memetic algorithm [42] [43], which has been established as a key methodology in complex optimization. Beyond the simple and adaptive hybrids of memetic algorithms, the research of memetic computation further culminates into a meme-centric "memetic automaton" [27] [44] that seamlessly integrates memes into units of domain information useful for problem-solving. Recently, memes have been defined as the transformation matrixes that can be reused across different problem domains for enhanced evolutionary search [45].

As with genes in genetics that serve as "instructions for building proteins", memes are synonymous to memetics as the instructions for carrying out behavior, constructing artifact for problem solving. In this paper, we consider a manifestation of neuronal memes as memory items or portion of an organism's neurally-stored information that fill the agents' mind universe. Further, beyond the formalism of memetic automaton and taking memes as focal point of interest in the context of MAS, we propose an evolutionary multi-agent Transfer reinforcement Learning framework (see Fig. 1) to develop the evolutionary learning process, particularly on agents' social interactions, for more efficient problem solving. In what follows, we direct our attention on the memetic representation and mechanisms pertaining to eTL.

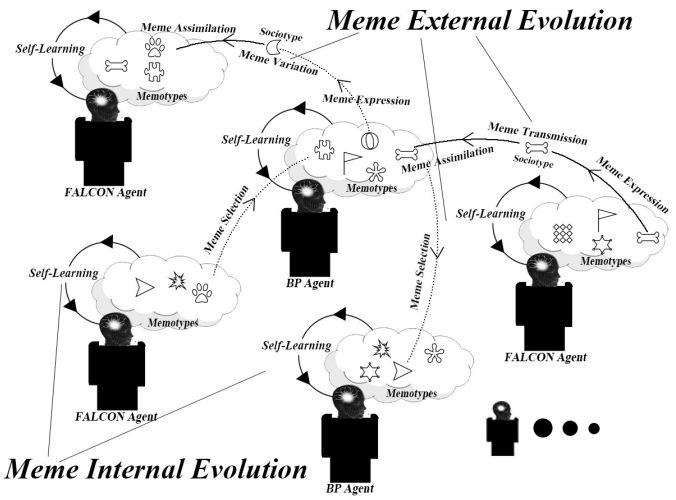


Fig. 1. Illustration of the evolutionary multi-agent Transfer reinforcement Learning framework.

B. Proposed eTL Model

In the design of eTL, memes represent the building blocks of the agents' mind universe, which are categorized into memotypes and sociotypes. Internally, memotypes (depicted by the different LEGO-like block objects that lies in the agents mind universe of Fig. 1) are described as the agents' ideas or knowledge captured as memory items or generalized abstractions inside the agents' mind universe. Externally, sociotypes are defined as the manifested behaviors that could be observed by the others. In particular, in the case that differing agents have distinct memotype structures, sociotypes in the proposed eTL offer a channel for agents to transfer knowledge across the whole population.

Meme representation and *meme evolution* form the two core aspects of eTL. It then undergoes *meme expression* and *meme assimilation*. *Meme expression* is defined for an individual to express their stored neuronal memes as behavioral actions, while *meme assimilation* captures new memes by translating corresponding behaviors into knowledge that blends into the individual's mind universe. In other words, *meme representation* pertains to what is a meme, while *meme expression/assimilation* activates/updates the meme during the learning process.

The *meme evolution* processes, namely *meme internal evolution* and *meme external evolution*, comprise the main behavioral learning aspects of eTL. To be specific, *meme internal evolution* denotes the process for agents to update their mind universe via self learning or personal grooming. In eTL, all agents undergo *meme internal evolution* by exploring the common environment simultaneously. During *meme internal evolution*, *meme external evolution* might happen to model the social interaction among agents mainly via imitation, which takes place when memes are transmitted. Specifically, *meme external evolution* happens whenever the current agent identifies an appropriate teacher agent via a *meme selection* process. Once the teacher agent is selected, *meme transmission* occurs to instruct how the agent imitates from others. During this process, *meme variation* then facilitates the innovative characteristics of knowledge transfer among agents. Upon receiving the feedback from the environment after performing an action, the agent then proceeds to update its mind universe accordingly.

III. Realization of eTL with Learning Agents

In this section, we present two realizations of learning agents that take the form of neurally-inspired learning structures, namely a Temporal Difference-Fusion Architecture for Learning and Cognition (FALCON) and a Back Propagation (BP) multi-layer neural network, respectively. To be more specific, FALCON is a natural extension of self-organizing neural models proposed in [11] for real-time reinforcement learning, while BP is a classical multi-layer network that has been widely used in various learning systems. As we discussed in Section II, eTL comprises several evolutionary operators including *meme representation*, *meme expression*, *meme assimilation*, *meme internal evolution* and *meme external evolution*. In what follows, we present the detailed realization for each of these operators.

A. FALCON Learning Agent

A FALCON learning agent under the proposed meme-inspired evolutionary mechanism is composed of FALCON

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2664665, IEEE Transactions on Evolutionary Computation

meme representation, meme activation, meme competition, sociotype readout, memotype matching and memotype learning.

1) FALCON Meme Representation: The mind universe of a FALCON agent (depicted in Fig. 2) employs a three-channel neural network architecture which consists of a category field F_2 and three input fields, namely, a sensory field F_1^{c1} for representing current states, a motor field F_1^{c2} for representing actions, and a feedback field F_1^{c3} for representing reward values.

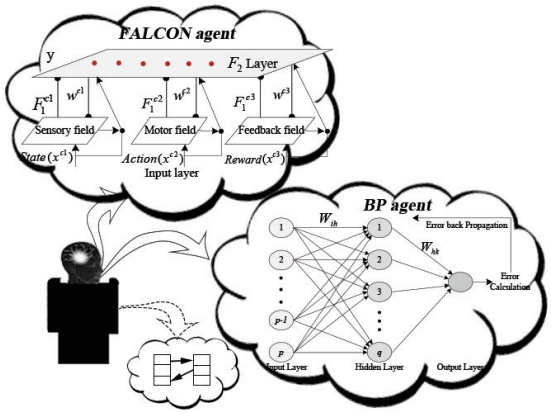


Fig. 2. Illustration of meme representation taking the form of FALCON and BP.

Input vectors: $\mathbf{I} = (\mathbf{S}, \mathbf{A}, \mathbf{R})$ denotes the input vectors where $\mathbf{S} = (s_1, s_2, ..., s_n)$ denotes the state vector, and s_n indicates the value of the input sensory n; $\mathbf{A} = (a_1, a_2, ..., a_m)$ denotes the action vector, and a_m indicates the m^{th} possible action; $\mathbf{R} = (r, 1-r)$ denotes the reward vector, and $r \in [0, 1]$.
Activity vectors: $\mathbf{x}^{ck}$ denotes the F_1^{ck} activity vector for $k = 1, 2, 3$ where $\mathbf{x}^{c1}, \mathbf{x}^{c2}, \mathbf{x}^{c3}$ correspond to the input state $\mathbf{S}$, action $\mathbf{A}$, and reward $\mathbf{R}$, respectively.
Weight vectors: $\mathbf{W} = (\mathbf{w}_1^{ck}, \mathbf{w}_2^{ck}, ..., \mathbf{w}_j^{ck})$ denotes the weight vector wherein $\mathbf{w}_j^{ck}$ indicates the weight associated with the j^{th} neuron in layer F_2 for learning the input activity vector F_1^{ck}. At the beginning, F_2 only has one uncommitted neuron whose weight is initialized by all 1's. When the uncommitted neuron is selected to learn the association, it becomes committed and another uncommitted neuron will be initialized.

Internally, a memotype is defined as the meme inhabiting inside the mind universe of a FALCON agent, which is created and stored in the cognitive field of F_2. All of these memotypes (neurons) form the knowledge of the agent which models the association of the input states and action, thus essentially providing instruction for selecting the appropriate actions. Externally, the sociotype meme of an agent refers to its expressed actions or behaviors, which can be observed and imitated by other agents.

2) FALCON Meme Expression:

- **Meme activation:** Firstly, a bottom-up propagation process occurs such that the activities of the memotype in the F_2 field are computed. Specifically, given the activity vectors $\mathbf{x}^{ck}$, the value T_j that represents the similarity of activity vectors with their respective weight vectors for each meme j of F_2 is computed as follows:

$$T_j = \sum_{k=1}^{3} \gamma^{ck} \frac{|\mathbf{x}^{ck} \wedge \mathbf{w}_j^{ck}|}{\alpha^{ck} + |\mathbf{w}_j^{ck}|}, \tag{1}$$

where the fuzzy AND operation $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv min(p_i, q_i)$ for vectors $\mathbf{p}$ and $\mathbf{q}$, and the norm $|\cdot|$ is defined by $|\mathbf{p}| \equiv \sum_i p_i$. Choice parameters α^{ck} and contribution parameters γ^{ck} are predefined, and $\alpha^{ck} > 0, \gamma^{ck} \in [0, 1]$.
- **Meme competition:** *Meme competition* process takes place right after meme activation, in which the F_2 meme with the highest activation value is identified. Let $\mathbf{y}$ denotes the F_2 activity vector. The winner is indexed at J where

$$T_J = max\{T_j : \text{for all } F_2 \text{ node } j\}, \tag{2}$$

$y_J = 1$ and $y_j = 0$ for all $j \neq J$. This indicates a winner-take-all strategy.
- **Sociotype readout:** Upon selecting a winning F_2 meme J, the chosen meme performs a readout of its weight vectors into the input field F_1^{ck} such that

$$\mathbf{x}^{ck(new)} = \mathbf{x}^{ck(old)} \wedge \mathbf{w}_J^{ck}. \tag{3}$$

Consequently, the $\mathbf{x}^{ck(new)}$ of F_1^{ck} are the result of fuzzy AND of $\mathbf{x}^{ck(old)}$ and $\mathbf{w}_J^{ck}$.

3) FALCON Meme Assimilation:

- **Memotype matching:** Before meme J selected in the *meme competition* can be used for learning, a *memotype matching* process is conducted to access if the weight templates of meme J are sufficiently similar to their respective activity patterns. Specifically, resonance occurs only if the match function m_J^{ck} for each channel k meets the vigilance criterion:

$$m_J^{ck} = \frac{|\mathbf{x}^{ck} \wedge \mathbf{w}_j^{ck}|}{|\mathbf{x}^{ck}|} \geq \rho^{ck}, \tag{4}$$

where vigilance parameter ρ^{ck} is pre-defined and $\rho^{ck} \in [0, 1]$ for $k = 1, 2, 3$. If any of the vigilance constraints is violated, mismatch reset occurs in which the value T_J is set to 0 for the duration of the input presentation and another meme will be selected for the matching.
- **Memotype learning:** If resonance occurs successfully, the weight vector $\mathbf{w}_J^{ck}$ in each channel k is updated as follows:

$$\mathbf{w}_J^{ck(new)} = (1 - \beta^{ck})\mathbf{w}_J^{ck(old)} + \beta^{ck}(\mathbf{x}^{ck} \wedge \mathbf{w}_J^{ck(old)}), \tag{5}$$

where $\beta^{ck} \in [0, 1]$ denotes the learning rate parameters. The rationale is to learn by encoding the common attribute values of both the input vectors and weight vectors.

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2664665, IEEE Transactions on Evolutionary Computation

B. BP Learning Agent

A BP learning agent under the proposed meme-inspired evolutionary mechanism is composed of BP meme representation, neuronal meme (memotype) forward propagation, neuronal meme competition and sociotype readout, memotype backward error estimation and memotype learning.

1) BP Meme Representation: The mind universe of a BP agent has a three-layer architecture (as shown in Fig. 2), consisting of an input layer I for representing current states $\mathbf{S}$ and available actions $\mathbf{A}$, an output layer O consisting of only one neuron for evaluating the association between selected action and a particular state, a hidden layer H for increasing the expressiveness of the network, and a reward signal $\mathbf{R}$ for estimating the errors of output neurons. All hidden and output neurons receive inputs from the initial inputs or the interconnections and produce outputs by transformation using a symmetrical sigmoid function. In contrast to the FALCON agent whose memotypes are generated dynamically during the learning process, the internal memotypes housed in a BP agent are modeled as fixed neurons in the hidden layer.

2) BP Meme Expression:

- **Neuronal meme (memotype) forward propagation:** A forward propagation process first takes place in which the reward signals of performing each possible action are computed. Specifically, given the state vector $\mathbf{S}$, for each possible action a_m in $\mathbf{A}$, the outputs of the hidden memotypes are firstly calculated as

$$H_h = \frac{1}{1 + e^{(-Gain \times \sum_{i=1}^{p} I_i \times W_{ih})}}, h = 1, ..., q, \quad (6)$$

where $Gain$ is the gain of sigmoid function, I_i indicates the value of i^{th} output in layer I, W_{ih} indicates the connection weight from i^{th} neuron in layer I to the h^{th} memotype in layer H, p and q are the number of neurons in the layer I and H, respectively. Then, the output reward value $(O_k)_m$ of performing each action a_m from A is further computed as

$$(O_k)_m = \frac{1}{1 + e^{(-Gain \times \sum_{h=1}^{q} H_h \times W_{hk})}}, k = 1, \quad (7)$$

where H_h denotes the output of the h^{th} meme in the hidden layer, W_{hk} is the connective weight between h^{th} memotype in the hidden layer and the node in the output layer, q is the total number of memotypes in the hidden layer and $k = 1$ means the index of the only node in the output layer.

- **Neuronal meme competition and sociotype readout:** After *meme forward propagation*, the reward value $(O_k)_m$ is then used to identify the sociotype meme with the highest reward value. The winner of the winner-take-all competitive strategy is indexed as M where

$$O_M = max\{(O_k)_m : for\ all\ actions\ in\ \mathbf{A}\}. \quad (8)$$

The corresponding action a_M is thus read out as the identified sociotype meme.

3) BP Meme Assimilation:

- **Memotype backward error estimation:** After performing the identified sociotype meme a_M, a memotype backward error estimation process checks the difference between the actual output O_k and expected output T_k of the network. If neuron b is the node of output layer, the error ε_b is estimated via

$$\varepsilon_b^l = (T_b - O_b) \cdot O_b \cdot (1 - O_b), \quad (9)$$

where l is the index number of the layer. If neuron b is the memotype of the hidden layer, the error signal ε_b is then back propagated from the output layer:

$$\varepsilon_b^l = \left[\sum_k \varepsilon_k^{l+1} \cdot W_{bk}\right] \cdot H_b \cdot (1 - H_b), \quad (10)$$

where H_b denotes the output of the b^{th} memotype in the hidden layer.

- **Memotype learning:** With the error signal term, the weight vector of each meme in the hidden layer is updated using the generalized learning rule:

$$W_{ab}^{new} = W_{ab}^{old} + \eta \cdot \varepsilon_b^l \cdot O_a^{l-1} + \tau \cdot (\eta \cdot \varepsilon_b^l \cdot O_a^{l-1})^{old}, \quad (11)$$

in which $\eta \in [0, 1]$ is the constant learning rate, $\tau \in [0, 1]$ is the pre-defined positive momentum accelerating the convergence rate of error propagation process, O_a^{l-1} denotes the output value of the sublayer related to the connective weight and superscripts new and old represent the current and most recent training steps, respectively.

C. Meme Internal Evolution

A general illustration of the *meme internal evolution* and *meme external evolution* processes in the eTL framework is depicted in Fig. 1. More specifically, the *meme internal evolution*, as summarized in Algorithm 1, governs the growth of an individual's mind universe via self learning. It is obvious that the internal evolutionary process is made up of a sequence of learning trials, and the trials continue until the ultimate conditions, such as mission numbers and fitness levels, are satisfied (see Line 3). During the learning process, given current state *s* and a set of possible actions $\mathbf{A}$, an agent firstly predicts the Q-values by performing each possible action through a *meme expression* process (see Lines 5-6). Furthermore, the received Q-values are used to select an appropriate action based on an action selection strategy (see Line 7). 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 for performing the chosen action in the current state. The new Q-value is then used to learn the association from the chosen action and the current state to the new Q-value (see Lines 9-19).

In eTL, an ϵ-greedy action selection scheme is used to balance exploration and exploitation in the self-learning process (see Line 7). Compared to other more complex methods, ϵ-greedy requires no memorization of exploration specific data and is reported to be often the method of first choice as stated

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2664665, IEEE Transactions on Evolutionary Computation

Algorithm 1: Meme Internal Evolution Process

Input: Current state s, a set of possible actions $\mathbf{A} = (a_1, a_2, ..., a_m)$

1 **Initialization:** Generate the initial agents
2 **Begin**:
3 **while** *stop conditions are not satisfied* **do**
4 **for** *each current agent* $agt(c)$ **do**
5 **for** *each action* $a_m \in \mathbf{A}$ **do**
6 $Q(s, a_m) = Predict(s, a_m)$
7 **Select** an action a_M from $\mathbf{A}$: $a_M = Scheme(Q(s, a))$.
8 **Perform** selected a_M: $\{s', r\}$ =*Perform*(a_M) where s' is the resultant state and r is the immediate reward from the environment.
9 **Estimate** the reward $Q^{new}(s, a_M)$ by: $Q^{(new)}(s, a_M) = Q(s, a_M) + \vartheta\delta(1 - Q(s, a_M))$
10 **Do:**
11 **if** *agent* $\in$ *FALCON agents* **then**
12 *Meme activation* with vector {**S**,**A**,**R**}
13 *Meme competition* with vector {**S**,**A**,**R**}
14 *Meme matching* with vector {**S**,**A**,**R**}
15 *Memotype learning* with vector {**S**,**A**,**R**}
16 **else**
17 *Memotype forward propagation* with vector {**S**,**A**,**R**}
18 *Memotype backward error estimation* with vector {**S**,**A**,**R**}
19 *Memotype learning* with vector {**S**,**A**,**R**}
20 **End**

by Sutton [46]. This strategy selects an action of the highest $Q(s, a)$ value with probability $1 - \epsilon$ $(0 \leq \epsilon \leq 1)$, or takes a random action otherwise.

In addition, in case of the multiple-step prediction problems, the action selection should not only depend on the current states, since the agent can only know the merit of an action beyond several steps in the future. Therefore, the Temporal Difference (TD) method, such as Q-learning, is employed to estimate the value function of action-state pairs $Q(s, a)$, which indicates the goodness of a learning system to perform an action a given state s (see Line 9). Specifically, the iterative updating rules of Q-values is defined by:

$$Q^{(new)}(s, a) \leftarrow Q(s, a) + \vartheta\delta(1 - Q(s, a)), \qquad (12)$$

where $\vartheta \in [0, 1]$ is the learning parameter and δ is the temporal-difference error defined as:

$$\delta = r + \gamma max_{a'} Q(s', a') - Q(s, a), \qquad (13)$$

where $r \in [0, 1]$ is the immediate reward value, $\gamma \in [0, 1]$ is the discount parameter, a' is the predicted action under the next state s', and $max_{a'} Q(s', a')$ is the maximum estimated Q-value of the next state s'. The scale term $(1 - Q(s, a))$ can guarantee that the Q-values will remain to be bounded between 0 and 1, which will be consistent with the input values of the system. Further, the Q-value newly computed by the TD formula is encoded into the reward for 1) FALCON agents: $\mathbf{R} = (Q^{(new)}(s, a), 1 - Q^{(new)}(s, a))$ and 2) BP agents $\mathbf{R} = (T_k) = (Q^{(new)}(s, a))$.

D. Meme External Evolution

Meme external evolution, which serves to model the social interaction among the learning agents, is governed by four evolutionary factors, namely *meme selection*, *meme expression*, *meme imitation* and *meme variation*.

1) Meme Selection: Meme selection is inspired by Darwin's notion of natural selection, with the goal of identifying the teacher agent with the highest yield predicted. Through this mechanism, agents' knowledge (or memes) that are more beneficial for problem solving are duplicated exponentially in the knowledge universe pool, while those that are less helpful rarely get replicated. In the past few years, most researchers have concentrated on exploring "how an agent can learn from other agents".

Among them, uniform random selection is the simplest and most commonly used scheme for choosing a representative agent from the crowd, where each individual has equal chance of being chosen. Nevertheless, more efficient selection approaches are possible if useful information about the individuals is available. A well-known selection strategy is the "imitate-from-elitist", in which agents learn through a reward mechanism. Based on this elitism selection mechanism, Oliveira *et al.* [21] presented a study on interactive advice-exchange mechanism where agents share episodes of given states by seeking advice at critical times, or by requesting the demonstration of a solution to a specific problem from agents whose performance is currently better than all the other agents.

However, the advice exchange mechanism is likely to suffer from blind reliance problem since advisees have no knowledge of the holdings of the reliance and always seek the advice from the teacher agent with the best performance. It was reported that the best teacher agent may not always be the agent with the best performance. This might happen for several reasons. But the most common of them is that each agent is unique both genetically and memetically. Hence, the responding of each agent with a common action for the given states may not suffice. In such cases, an agent also needs to learn which of its partners it can trust and which has the most similar experience in dealing with the given circumstance. To tackle this blind reliance problem, the notion of "like-attracts-like" [47] has been introduced wherein agents learn from others not only based on the updated performance, but also on the similarities of the best matched knowledge under the given circumstance.

In our design, the decision of which peer an agent should learn from depends on both the agents' superior historical success, namely *elitist* and the agents' confidence in solving the request for a particular given situation, namely *similarity*. This *meme selection* process in eTL is thus a fusion of

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2664665, IEEE Transactions on Evolutionary Computation

the "imitate-from-elitist" and "like-attracts-like" principles. On one hand, the *elitist* is maintained as a centralized resource to express the judgement of agents' behaviors in accordance with the past knowledge. It is the simplest and most direct way to provide recommendations since agents do not need to compute the reputation while being able to identify the elitist agent directly. However, when the agents communicate with others, the interaction process might be very labour-intensive and the required information could be inefficient and not always desirable. The reason is that although the elitist has a higher performance than the partners, it might not be the most suitable candidate in solving the given unseen circumstance. On the other hand, *similarity* does not solely focus on the knowledge of the past, but the prediction on future reward under similar situations. This scheme does not impose higher performances in the partners but rather focuses on the sharing of relevant experiences in dealing with similar previously encountered situations. As a consequence, the *similarity* scheme narrows down the relevant *elitist* to learn from. Nevertheless, in the event that agents possess differing prior knowledge (in the form of memotypes) for evaluating a common experience, the *similarity*-based selection scheme would be futile since it is difficult for agents to judge whether the knowledge from the teacher is beneficial or detrimental to itself.

Algorithm 2: Meme Selection Process

Input: Current state s, a set of possible actions $\mathbf{A} = (a_1, a_2, ..., a_m)$

1 **Begin**:
2 **Get** current agent $agt(c) \in P$
3 **for** *each agent* $agt(v)$ *where* $v \neq c$ **do**
4 **Get** the current state of agent $agt(c)$ as s
5 **Get** the Si value of $agt(v)$ under the given s: $Si(agt(v)) = max\{Q(s, \mathbf{A})\}/Q_{best}$
6 **if** $(El(agt(v)) \geq El(agt(c)) \& Si(agt(v)) \geq Si(agt(c)))$ **then**
7 **Put** $agt(v)$ into set B
8 **for** *each agent* $agt(v)$ *in* B **do**
9 $Sc(agt(v)) = El(agt(v)) \times Si(agt(v))$
10 **Select** the teacher (or source) agent with highest $Sc(agt(v))$
11 **End**

In *meme selection*, the trustworthiness of performance is estimated in accordance with the positive and (or) negative outcomes produced by the agents in the past. The detailed outline of the *meme selection* process is given in Algorithm 2. Particularly, we define $El(agt(v))$ as the relative fitness of agent $agt(v)$ and $Si(agt(v))$ as the relative maximum predicted award to perform an action under the given state s:

$$\begin{cases} El(agt(v)) = Fitness(agt(v))/Fitness_{best}, \\ Si(agt(v)) = max\{Q(s, \mathbf{A})\}/Q_{best}, \end{cases} \tag{14}$$

where $Fitness(agt(v))$ denotes the fitness of an agent $agt(v)$, v in $agt(v)$ is the index of the v^{th} agent, $Fitness_{best}$ is the reputation of the elite or the best performing agent, and $Fitness = Fitness + 1$ if the agent completes a mission successfully; $Q(s, \mathbf{A})$ indicates the maximum predicted Q-value of agent $agt(v)$ for given state s, and the Q_{best} is the maximum historical Q value of agent $agt(v)$. In the learning process, agents that have been found to have both higher *elitist* and *similarity* values than the current agent $agt(c)$ (also known as target agent) are identified to form the set B (see Lines 6-7):

$$\begin{aligned} B = \{agt(v) \in P | (El(agt(v) \geq El(agt(c))) \\ \& Si(agt(v)) \geq Si(agt(c))\}, \end{aligned} \tag{15}$$

where P is the population of all agents in the environment. Subsequently, the agent with the highest Sc value shall serve as the teacher agent $agt(s)$ (or source agent) in the selection set for *meme transmission* (see Lines 8-10):

$$Sc(agt(v)) = El(agt(v)) \times Si(agt(v)). \tag{16}$$

2) Meme Transmission: Once a selection is made, *meme transmission* between agents with unique learning capabilities happens by means of imitation. In our problems, each agent learns in the same environment and has the same act abilities. Therefore, imitation could offer the advantage for the imitating agent to behave at approximately the same performance level as its imitator. In addition, if a human player is imitated, the autonomous agents could exhibit more believable and complex human-like behaviors, seamlessly.

Algorithm 3: Meme Transmission Process

Input: Current state s, a set of possible actions $\mathbf{A} = (a_1, a_2, ..., a_m)$

1 **Begin**:
2 **Get** current agent $agt(c) \in P$
3 **Get** current state $s(agt(c))$ of $agt(c)$
4 **Pass** $s(agt(c))$ to the teacher agent $agt(s)$
5 **Get** action $a(agt(s))$ of $agt(s)$ through Steps 5-7 in Algorithm 1
6 **Perform** variation on $a(agt(s))$ with probability ν
7 **Perform** $a(agt(s))$: $\{s', r\}$ =*Perform*$(a(agt(s)))$ where s' is the resultant state and r is the immediate reward from the environment.
8 The remaining steps are the same as those in Algorithm 1
9 **End**

The *meme transmission* process in the proposed eTL is summarized in Algorithm 3. More specifically, the current agent $agt(c)$ first passes its state to the selected teacher agent $agt(s)$ (see Lines 3-4). Subsequently, agent $agt(s)$ expresses its corresponding action $a(agt(s))$ by *meme expression*, which can be observed by agent $agt(c)$ (see Line 5). Then, agent $agt(c)$ assimilates the received action into its mind universe by means of *meme assimilation* (see Lines 7-8). In addition,

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2664665, IEEE Transactions on Evolutionary Computation

the detailed variation process which may occur in Algorithm 3 (see Line 6) is described in *meme variation* as follows.

3) Meme Variation: Meme variation serves to give the intrinsic innovation tendency of selected sociotypes for *meme transmission* in the evolutionary knowledge transfer process. More specifically, for knowledge transmission without variation, agents might believe that the knowledge as represented by sociotypes of an elite agent is always good based on its particular demonstration of performance at the given circumstance. Due to the nonlinearity of the knowledge transfer process, this bias could spread and spiral out of control since it infects any other agents which come into contact with [22]. This would suppress the agents' ability to explore the environment. Therefore, *meme variation* is considerably essential for retaining diversity in agents' attitude towards the innovative learning process.

In the proposed eTL framework, *meme variation* occurs at the expression and imitation stages, in which a probabilistic interference cost is added to the estimated action's Q-value to allow different actions to be selected for expression:

$$Q^t = \lambda \times Rand + (1 - \lambda) \times Q, \tag{17}$$

where Q^t is the mutated Q, $\lambda \in [0, 1]$ is the control parameter of the degree of randomness, and $Rand \in [0, 1]$ is a random value with uniform distribution. The pre-defined $\nu \in [0, 1]$ is the probability to control the frequency of the variation process.

E. eTL with FALCON and BP

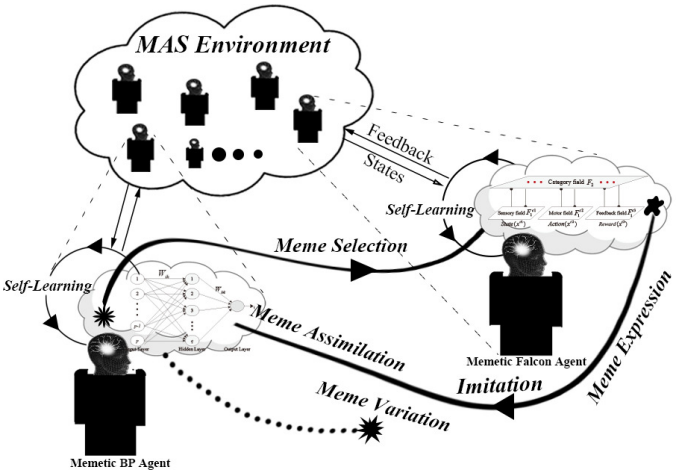


Fig. 3. The fully general multi-agent scenario with the proposed eTL paradigms. Agents employ FALCON or BP as the learning machine in their mind universes and learn the domain knowledge for performing appropriate actions given environmental states. They may also interact directly as indicated by the arrows between agents.

The proposed eTL is illustrated in Fig. 3 where a population of agents learn and evolve in a single MAS environment. Particularly, both the aforementioned FALCON and BP agents denote the two distinct connectionist learners that coexist in the environment simultaneously. In the design of eTL, each individual can interact with the environment or one another by undergoing the *meme internal evolution* process and/or the *meme external evolution* process, where the former denotes the learning process that updates an individual's internal knowledge via personal grooming. The latter process which is central to the behavioral aspects of imitation, models the interactions among multiple agents.

Algorithm 4: Basic eTL Framework

```
1  Begin:
2  Initialization: Generate the initial N agents
3  while stop conditions are not satisfied do
4      for each current agent agt(c) do
5          /*Perform meme internal evolution*/
6          See Algorithm 1, Line 5-7
7          if Identify the teacher agent via meme selection
             then
8              /*Perform meme external evolution*/
9              Perform meme expression with agt(s) under
                 the state of agt(c)
10             Perform meme variation with probability ν
11             /*ν is the frequency probability of variation
                 process*/
12             Perform meme transmission for agt(c) to
                 assimilate agt(s)'s action
13             /*End meme external evolution*/
14         Evaluate the probability C_p of each agent
15         /*Perform the action from the meme external
             evolution with probability C_p, else perform
             action from the meme internal evolution*/
16         The remaining steps are the same as those in
             Algorithm 1.
17         /*End meme internal evolution*/
18 End
```

The basic steps of the eTL are outlined in Algorithm 4. To be specific, a population of N agents is first initialized with BP or FALCON architectures. Furthermore, each of the current agents undergoes *meme internal evolution*. Meanwhile, *meme external evolution* proceeds if an agent $agt(c)$ identifies the teacher agent $agt(s)$ via *meme selection*. Then, agent $agt(c)$ shall undergo the action predicted using *meme internal evolution* or *meme external evolution* according to probability C_p:

$$C_p(agt(c)) = 1 - \frac{Fitness(agt(c))}{Fitness(agt(s))} \times \frac{Si(agt(c))}{Si(agt(s))}, \tag{18}$$

where $Fitness(agt(c))$ is the fitness of agent $agt(c)$. $Fitness(agt(s))$, on the other hand, denotes the fitness of the selected teacher agent, $Si(agt(c))$ is the similarity value of current agent $agt(c)$ for the given states, $Si(agt(s))$ is

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2664665, IEEE Transactions on Evolutionary Computation

the similarity value of the teacher agent. Thus, each agent undergoes the action provided by *meme external evolution* with probability C_p. During *meme external evolution*, *meme variation* occurs at a probability of v to maintain the diversity towards agents' selection for *meme transmission*.

IV. Empirical Study

In order to study the effectiveness and efficiency of the proposed evolutionary transfer learning approach, we firstly validate eTL on a commonly used Minefield Navigation Tasks (MNT) [32]. Subsequently, a well-known commercial first person shooter game, namely "Unreal Tournament 2004" (UT2004) [33], is used to investigate the performance of eTL under complex game scenarios.

A. Experimental Platform - Minefield Navigation Task

In MNT, the goal of each unmanned tank (or agent) is to navigate across a minefield comprising of randomly positioned mines, other tanks and target within a given stipulated time frame successfully. Fig. 4 depicts an overall view of the MNT environment. The unmanned tanks that spawn randomly within the field at the beginning of each mission are equipped with sonar sensors so that they can get access to a set of detections, including mine detection, agent detection and target bearing. Further, each tank in MNT possesses sonar sensors that have a 180° forward view, including left, left oblique, front, right oblique and right.

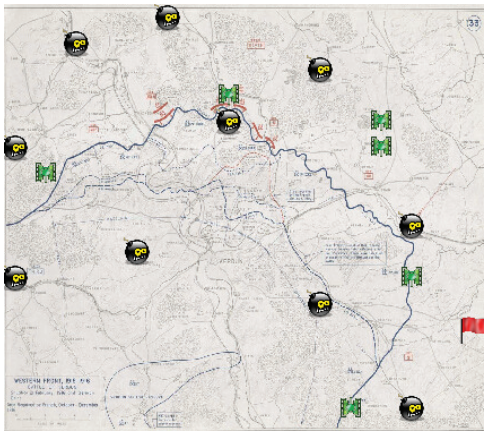


Fig. 4. A snapshot of the Minefield Navigation Task.

According to the input state, each unmanned tank performs one of the five possible actions at each step: $\mathbf{A}$ = {*LEFT, LEFT FRONT, FRONT, RIGHT FRONT, RIGHT*}. Particularly, an unmanned tank is rewarded with a positive value of 1 when the tank arrives at the target. Or, if the tank hits a mine, collides with other tanks or does not reach the target within the given time frame, the tank will be assigned with zero reward. In accordance with the Q-learning lemma, negative feedback signal is not used in this reward schema to ensure the Q value is always bounded within the desired $[0, 1]$ interval. In the experimental study, there is a total of 10 mines and 1 target (known as the red flag) that are randomly generated over missions in a 16×16 minefield. A mission completes when all tanks reach the target (success), hit a mine or collide with another tank (failure), or exceed 30 time steps (out of time), as considered in [11] [22].

TABLE I. Summary of the parameter setting in the proposed eTL

FALCON Parameters		
Choice Parameters $(\alpha^{c1}, \alpha^{c2}, \alpha^{c3})$	$(0.1, 0.1, 0.1)$	
Learning Rates $(\beta^{c1}, \beta^{c2}, \beta^{c3})$	$(1.0, 1.0, 1.0)$	
Contribution Parameters $(\gamma^{c1}, \gamma^{c2}, \gamma^{c3})$	$(0.5, 0.5, 0)$	
Baseline Vigilance Parameters $(\rho^{c1}, \rho^{c2}, \rho^{c3})$	$(0.2, 0.2, 0.5)$	
BP Parameters		
Learning Rate η	0.25	
Momentum Factor τ	0.5	
Gain of Sigmoid Function $Gain$	1.0	
Temporal Difference Learning Parameters		
TD learning rate ϑ	0.5	
Discount Factor γ	0.1	
Initial Q-value	0.5	
ϵ-greedy Action Policy Parameters	FALCON	BP
Initial ϵ value	0.5	0.5
ϵ decay rate	0.0005	0.00025
Transfer Variation Parameters		
Degree of Randomness λ	0.1	
Frequency of Variation ν	0.1	

B. Experimental Configuration

The parameter settings of FALCON, BP, TD methods and the eTL configured in the present experimental study are summarized in Table I. Notably, the configurations on our experiments in MNT are considered to be consistent with the previous study in [11] for the purpose of fair comparisons. The results with respect to the following metrics are then reported:

- SR denotes the average success rate of the agents on completing the missions.
- GM denotes the average number of memotypes in agents. In particular, as discussed in Section III, the memotypes of FALCON and BP are defined as the number of memes generated in the cognitive field F_2 and the number of memes pre-configured in the hidden layer, respectively. In general, for the same level of success rate, agents with lower number of memotypes are preferred, since it implies better generalization of knowledge.
- KN denotes the average number of knowledge transfer events happening between agents. In general, a higher number of knowledge transfer implies higher computational costs incurred to perform transfer learning among agents.

C. Performance of eTL using FALCON or BP as Learning Agents.

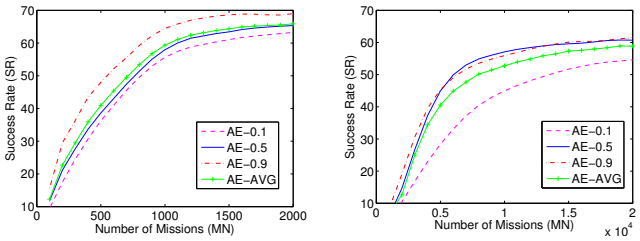
In this subsection, the objective of the present experimental study is to investigate the performance of the proposed eTL

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2664665, IEEE Transactions on Evolutionary Computation

with FALCON or BP learning agents on completing the Minefield Navigation Tasks. Specifically, the MAS is composed of six FALCON or BP agents in the 16x16 size map of the Minefield Navigation Task platform where the learning architecture of MAS is discussed in Section II-B. Further, the interaction and implementation of learning agents are presented in Section III. In this study, 60 memotypes are configured in BP to achieve the best success rate while maintaining a low space complexity [11]. To maintain consistency with previous studies [11] [22] [47], we conduct 30 sets of independent experiments and report the average performances of *1)* FALCON agents at 100-mission intervals over a total of 2000 missions and *2)* BP agents at 1000-mission intervals over a total of 20000 missions, respectively.

Moreover, two state-of-the-art TL approaches for MAS, namely Advice Exchange (AE) model [21] and Parallel Transfer Learning (PTL) [20], are considered here for the purpose of comparison. In AE, the agents, or advisees, learn from better performed advisors based on the "imitate-from-elitist" selection strategy. More specifically, an agent $agt(c)$ with poor performance will seek advice from the best performing agent $agt(b)$ if $(q_c < dq_b)$, where q_c denotes the average quality of agent $agt(c)$, q_b is the average quality of agent $agt(b)$ and d is a user defined discount parameter in the interval of $[0, 1]$. For fair comparison, in our experiment, we investigate various values of the discount factor, particularly, for i) a small value 0.1 (AE-0.1), ii) a medium value 0.5 (AE-0.5) and iii) a large value 0.9 (AE-0.9), respectively, and report the average success rates of the agents (AE-AVG). Fig. 5 depicts the average success rates of agents under differing AE models. Notably, AE models with different discount factors reported distinct success rates. Further, PTL is a TL method where the current agent leverages from the knowledge broadcasted by the others in the environment. In particular, if an agent has useful information to share, it shall broadcast the information to all other agents. Meanwhile, the current agent will also check its communication buffer to determine if any information has been received and then decides whether to accept the information or discard it.



(a) FALCON agents. (b) BP agents.

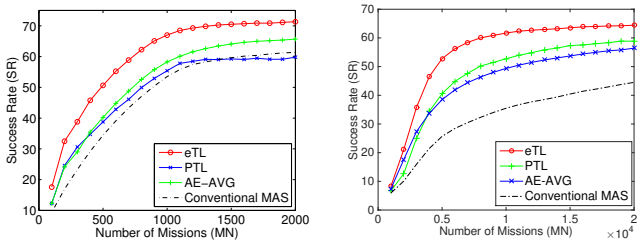
Fig. 5. Success rates of agents under AE models with discount factor 0.1, 0.5 and 0.9 on completing the missions in Minefield Navigation Tasks.

The complete results pertaining to the success rate (SR), generated memotypes (GM) and interval-unit knowledge transfer numbers (KN) of both BP and FALCON agents under the conventional MAS (Conv. M), Advice Exchange model (AE), Parallel Transfer Learning (PTL) and the proposed eTL are summarized in Table II and Figs. 6-7. The detailed analyses of the obtained results shall be discussed comprehensively in what follows.

TABLE II. PERFORMANCE COMPARISON AMONG ETL, EXISTING TL APPROACHES AND CONVENTIONAL MAS (SR: SUCCESS RATE, GM: GENERATED MEMOTYPES, KN: KNOWLEDGE TRANSFER NUMBERS).

		FALCON agents			BP agents		
#	Scenarios	SR	KN	GM	SR	KN	GM
1	eTL	71.3	324	188	64.2	1715	60
2	PTL	59.8	131	191	56.4	178	60
3	AE-0.1	63.3	24	152	54.6	59	60
4	AE-0.5	65.4	58	158	60.7	396	60
5	AE-0.9	69.0	428	168	61.4	4959	60
6	AE-AVG	65.8	170	159	58.8	1797	60
7	Conv. M	61.3	0	144	44.5	0	60

1) Comparison of MASs with and without TL: The success rates obtained by the MAS with and without TL approaches are given in the SR columns of Table III. Their corresponding learning trends are also depicted in Fig. 6. It it notable that most TL approaches outperform the conventional MAS. This is attributed to the TL approaches which endow agents with capacities to benefit from the knowledge transferred from the better performing agents, thus accelerating the learning rate of the agents in solving the complex task more efficiently and effectively. Agents in the conventional MAS, on the other hand could only undergo learning via personal grooming. To illustrate the efficacy of agents in the eTL framework in solving the minefield navigation problem, Fig. 7 depicts the sample snapshots of the navigated routes made by the FALCON agents. As can be observed, at the early learning stages, such as 1 and 500 learning missions, the FALCON agents have high probability to hit the mines or collide with other agents. While learning progresses, these agents tend to exhibit higher success rates.



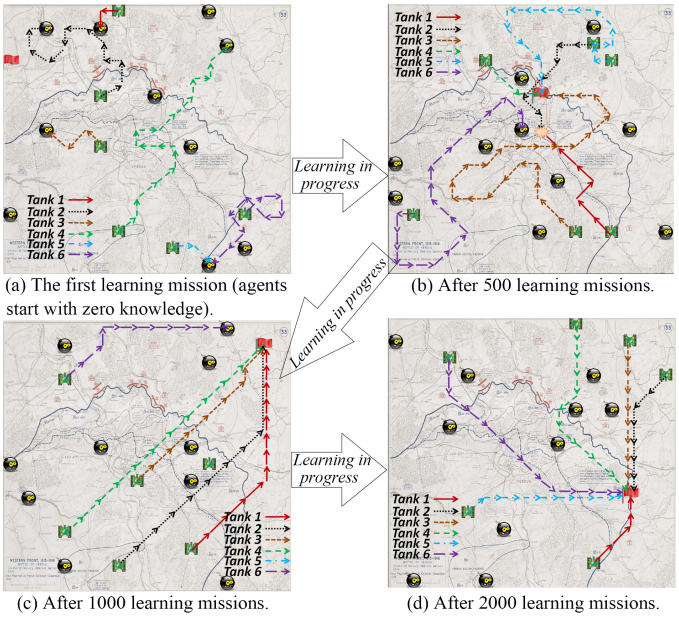
(a) FALCON agents. (b) BP agents.

Fig. 6. Success rates of FALCON or BP agents under eTL, PTL, AE-AVG and conventional MAS on completing the missions in MNT.

2) Comparison of eTL against other state-of-the-art TL approaches: When compared to the state-of-the-art TL approaches, our proposed eTL is shown to achieve superior performance in terms of success rate throughout the learning process. In particular, FALCON and BP agents with the proposed eTL reported approximately 11.5% and 7.8% improvements in

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2664665, IEEE Transactions on Evolutionary Computation



(a) The first learning mission (agents start with zero knowledge).
(b) After 500 learning missions.
(c) After 1000 learning missions.
(d) After 2000 learning missions.

Fig. 7. Snapshots of the FALCON agents' navigation routes on completing 1, 500, 1000 and 2000 learning missions in Minefield Navigation Task. The objective of each FALCON agent (denoted as a tank in the figure) in a learning mission is to arrive at the target successfully by navigating across the minefield safely and within the allocated time span. Thus a learning mission completes when all tanks reach the target, hit a mine or collide with another tank, or exceed the given time steps. All tanks and mines are randomly generated for each learning mission hence different learning missions have unique navigational routes.

success rate, respectively, over PTL at the end of missions (see Table III, column SRs). This is due to the reason that, when deciding whether to accept the information broadcasted by the others, agents in PTL tend to make incorrect predictions on previously unseen circumstances.

Further, the proposed eTL also demonstrated superiority in attaining higher success rates than all AE models. As discussed in Section III-D, this can be attributed to the *meme selection* operator of the proposed eTL, which considers a fusion of the "imitate-from-elitist" and "like-attracts-like" principles so as to give agents the option of choosing more reliable teacher agents over the AE model. Moreover, it is worthy noting that the AE model with a discount factor of 0.9 (labeled as AE-0.9) attained a success rate that is the closest to that of the proposed eTL. However, AE-0.9 incurred a higher knowledge transfer numbers (i.e., computational efforts) during the learning process (see Table III, column KNs).

Moreover, in order to quantitatively evaluate how effective the memotypes are generated in FALCON agents under different TL approaches, the *effectiveness ratio* is defined as follows:

$$Ratio(TL) = GM/(SR(TL) - SR(Conv.M)), \qquad (19)$$

where $(SR(TL) - SR(Conv.M))$ denotes the improvements on success rate of the learning agents exhibited by the TL approaches against the conventional MAS. In general, a smaller positive $Ratio$ value is preferable, which implies a higher knowledge generalization performance.

TABLE III. THE EFFECTIVENESS OF GENERATED MEMOTYPES IN FALCON AGENTS UNDER THE ETL AND OTHER EXISTING TL APPROACHES.

		FALCON agents		
#	Metrics	eTL	PTL	AE-AVG
1	Generated Memotypes	188	191	159
2	Success Rate Improvement	10.0	-1.5(1)	4.5
3	Memotype Effectiveness	18.8	191	35.3

Table III summarizes the effectiveness ratio of the memotypes generated in FALCON agents under different TL approaches at the end of the learning missions. Notably, the proposed eTL reported the most effective memotypes in FALCON agents since it attained the smallest effectiveness ratio among all the three TL approaches.

D. Performance of Proposed eTL using Heterogeneous Agents

In the previous subsection, a study on MAS with a homogenous setting has been considered, i.e., all agents in the system are assumed to be uniform and bear the same learning machine. In this subsection, we further consider the complex realistic scenario involving a diversity of heterogeneous agents in a MAS. In particular, a mixture of three FALCON agents and three BP agents are employed in the heterogeneous MAS for solving the Minefield Navigation Tasks. Notably, the architecture of MAS and the interaction among heterogeneous learning agents are discussed in Section III.

In the heterogeneous MAS, knowledge transfer among agents is more complex than that in the homogenous MAS. This is because FALCON and BP agents possess unique learning structures, hence they have different learning capabilities, including specialized action prediction rules, learning speed, etc. As such, the knowledge identified by TL approaches working well in homogenous MAS might become futile and even detrimental to the agents with dissimilar learning structures. For instance, in the heterogeneous MAS where both FALCON and BP agents are configured using the PTL as the TL approach, BP agents will broadcast the useful knowledge to all other agents. Considering that the FALCON agents learn much faster than BP agents (as discussed in Section IV-C), these knowledge broadcasted from BP agents could always lag behind that of FALCON agents, and hence deteriorate the performance of FALCON agents as depicted in Fig. 8(a). Thus, in this subsection, we further conduct experiments to investigate the performance of our proposed eTL in the heterogeneous MAS setting.

The success rate (SR), knowledge transfer numbers (KN) and generated memotypes (GM) of both FALCON and BP agents obtained under the conventional MAS and TL approaches are presented in Table IV. Moreover, the learning trends in terms of success rate are depicted in Fig. 8.

Notably, FALCON agents under AE and PTL learning approaches often succumbed to poor success rate in the long run. According to the average success rate attained by AE models with discount factor of 0.1, 0.5 and 0.9 (see Figure

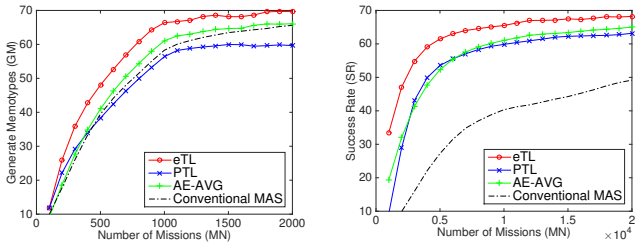
1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2664665, IEEE Transactions on Evolutionary Computation

TABLE IV. COMPARISON AMONG ETL, EXISTING TL APPROACHES AND CONVENTIONAL MAS IN HETEROGENEOUS MAS (SR: SUCCESS RATE, GM: GENERATED MEMOTYPES, KN: KNOWLEDGE TRANSFER NUMBERS).

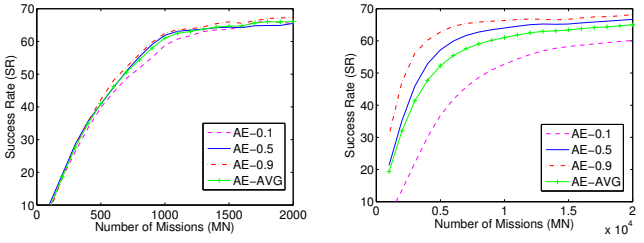
		FALCON agents			BP agents		
#	Scenarios	SR	KN	GM	SR	KN	GM
1	eTL	69.7	310	168	68.2	1189	60
2	PTL	59.7	93	167	63.1	267	60
3	AE-0.1	65.5	27	144	60.1	33	60
4	AE-0.5	65.5	62	154	66.7	266	60
5	AE-0.9	67.3	423	158	68.2	4700	60
6	AE-AVG	66.1	171	152	65.0	1667	60
7	Conv. M	65.2	0	145	49.1	0	60

9), competitive success rates of FALCON agents to that of the conventional MAS is reported. On the other hand, under the PTL scheme, a significant drop of 5.5% in success rate as compared to the conventional MAS is observed at the end of the learning process (see Table IV, column SRs). As aforementioned, this is a result of the blind reliance on the knowledge transferred from the BP agents that misled the FALCON agents into performing erroneous acts under the heterogeneous MAS environment.



(a) FALCON agents. (b) BP agents.

Fig. 8. Success rates of FALCON and BP agents under eTL, PTL, AE-AVG and conventional MAS on completing the missions in the heterogeneous MAS.



(a) FALCON agents. (b) BP agents.

Fig. 9. Success rates of FALCON and BP agents under AE models with discount factor 0.1, 0.5 and 0.9 on completing the missions in the heterogeneous MAS.

On the other hand, the proposed eTL significantly outperforms the counterparts, for both FALCON and BP agents, throughout the entire learning process. Specifically, at the end of the learning process, it attained the highest success rate of 69.7% and 68.2% for FALCON and BP agents, respectively (see Table IV, column SRs). Particularly, according to the result in Fig. 8, FALCON agents are noted to learn much faster than BP agents. Therefore, the knowledge transferred from FALCON agents is helpful for complementing the learning ability of BP agents, thus enhancing the SRs of BP agents significantly on the MNT problem. The result thus further highlights the efficacy of the *meme selection* strategies in the eTL in the choice of reliable and efficient information while reducing blind reliance in transfer learning, specifically in the context of the heterogeneous MAS.

Moreover, we investigated on KNs and GMs of FALCON and BP agents in both homogenous and heterogeneous MNTs. As can be observed in Tables II and IV, PTL reports the least KNs, significantly less than those of eTL and AE-AVG. However, although PTL generates far less KNs, it generates almost the most number of memotypes in FALCON agents among all TL approaches. This indicates that the transferred knowledge in PTL tends to generate too much redundant and possibly detrimental memotypes, hence leading to the poor success rates of FALCON agents. On the other hand, AE-AVG reports the smallest number of memotypes among all TL approaches for FALCON agents. Particularly, according to Table IV, the AE approach with a discount factor of 0.9 (labeled as AE-0.9) generates around 158 memotypes and incurs a higher knowledge transfer number (423). eTL, on the contrary, generates lower KNs, while still obtaining both higher numbers of memotypes and success rates in FALCON agents. The obtained result thus highlights the efficiency of eTL in making use of the transferred knowledge, while generating useful memotypes in FALCON agents.

In summary, the performance of the proposed eTL in the heterogeneous MAS is consistent to that of the homogenous settings. This demonstrates the superiority of the proposed eTL in producing higher success rate against existing TL approaches in both homogenous and heterogeneous MASs.

E. Performance of Proposed eTL in Unreal Tournament 2004

Further, we investigate the performance of the proposed eTL in the popular but complex commercial first-person shooter computer game - “Unreal Tournament 2004” (UT2004). Over the past decades, first-person shooter computer games have attracted dramatic attentions and garnered a large part of the multibillion dollars computer game industries. Notably, the modeling of autonomous non-player characters is of great importance for the commercial success since it makes the game more playable, challenging and more importantly, it has been deemed to be useful in improving the players degree of satisfaction. Therefore, we further consider a practical game problem where eTL is applied to model the non-player characters in UT2004.

Fig. 10 depicts a snapshot of the running game taken from the judge view, which showcases the shootout features in UT2004, such as weapons, armour or medical kits, etc. In UT2004, all autonomous robots that spawn randomly are equipped with a set of sensors. Taking the information captured by sensors as the input state, autonomous robots could learn and utilize the association of their current states, behavior

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2664665, IEEE Transactions on Evolutionary Computation

Fig. 10. A snapshot of the "Unreal Tournament 2004".

selections and rewards. Specifically, the state vector of a robot comprises variables encoded by boolean or discretized numbers within [0,1]. The base status of an autonomous robot includes the health level, being damaged or not, whether enemies are spotted, ammo level, and variables that indicate the current action of a robot by means of archiving previous states and actions. The reward vector only comprises two variables, namely the reward and its complement. Defining 0.5 as a successful hit and 1 as a successful kill, the reward system is simple for fast learning. Table V illustrates a sample knowledge of the autonomous robot during training process. For each state, the robot performs a set of pre-defined combat actions based on some rule-based heuristics. The performing reward is then used to update the agents' mind universe.

TABLE V. SAMPLE KNOWLEDGE OF THE AUTONOMOUS ROBOTS.

STATE	$S = (1.0, 0.0, 0, 1, 1, 0, 0.4, 0.6, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1)$
IF	health is 1.0, and not being damaged, and opponent is in sight, and has 40 percent of ammo, and currently in collecting item state.
THEN	go into engaging fire state
WITH	reward of 0.768

In our design, we firstly initialize two groups of six armed robots in the region of interest. Each armed robot is equipped with a set of sonar sensors and weapons. One group of armed robots are labeled as *Hunters* and each of them has a mind universe that is governed by human defined rules [33]. The second group comprises autonomous armed robots with FALCON as the online learning machine. Under the simple "DeathMatch" scenario, two groups of armed robots fight against each other by making use of the available resources, including weapons, medical kits, armors, etc. When the armed robots are killed by the enemies, a new one will spawn randomly in their corresponding base. The battle repeats until each group reaches a minimum missions of 200, wherein one mission completes only if an armed robot in the group is killed by the enemies. To evaluate the performance of our proposed eTL in UT2004 more quantitatively, the kill rate (KR) of game characters has been defined by the percentage of enemy kill numbers over agent death numbers [33] [34]. Typically, a higher KR is preferred since it indicates that a game character could kill more enemies while is less killed by the others.

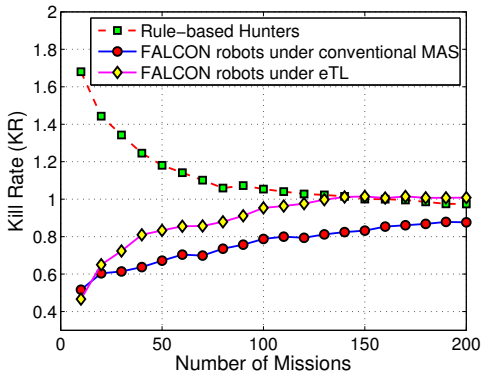


Fig. 11. *Kill Rate* of FALCON robots under the conventional MAS and the proposed eTL fighting against the rule-based *Hunters*.

Fig. 11 summarizes the fighting performance of rule-based *Hunters* and FALCON robots under both the conventional MAS and proposed eTL, in terms of *Kill Rate* (defined as KR in Section IV-B). Notably, at the early learning stage, the rule-based *Hunters* have significant higher KR than FALCON robots since the rules are well-designed for *Hunters* while FALCON robots suppose to randomly explore the environment at the beginning of the learning process. However, the KR of *Hunters* continues to decrease as the FALCON robots learn during combats. After 200 missions, the KR of the FALCON team under the conventional MAS is approximately 10% lower than that of the *Hunters*. FALCON robots under the proposed eTL, on the other hand, report significant higher KR than those under the conventional MAS throughout the learning process. Particularly, FALCON robots with eTL attained KR of 1.0 within 200 missions. The improved performance obtained by the proposed eTL demonstrates its efficacy in improving the combat performance of FALCON robots via the evolutionary knowledge transfer in the first-person shooter computer game - "Unreal Tournament 2004".

V. CONCLUSION

In this paper, we propose an evolutionary Transfer reinforcement Learning framework (eTL), which is governed by several meme-inspired evolutionary operators, namely *meme representation*, *meme expression*, *meme assimilation*, *meme internal evolution* and *meme external evolution*. In particular, detailed designs of the meme-inspired operators and two realizations of the learning agents in eTL that take the form of the Temporal Difference-Fusion Architecture for Learning and Cognition (FALCON) and the classical Back Propagation (BP) multi-layer neural network, respectively, are presented. The performance efficacy of our proposed eTL is investigated via comprehensive empirical studies and benchmarked against both the conventional MAS without knowledge transfer and two state-of-the-art MAS transfer learning approaches (i.e., Advice Exchange Model (AE) and Parallel Transfer Learning (PTL)), on the widely used Minefield Navigation Task

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2664665, IEEE Transactions on Evolutionary Computation

platform, under homogenous as well as heterogeneous MASs settings. The superior performance achieved by eTL confirms its efficacy in conducting evolutionary knowledge transfer in MAS, hence suggests the basis for greater research in developing evolutionary transfer learning strategies. Also, a well-known first person shooter game, namely "Unreal Tournament 2004" (UT2004), is used to demonstrate the effectiveness of the proposed eTL under the complex problem solving scenarios.

As discussed, memes encoded in computational representations form the underlying building blocks (knowledge, belief, emotion, etc.) of the mind-universe of an individual. In this work, memes generally take the form of knowledge encoded in neural structures such as FALCON and BP. For future work, we would like to explore the generality and adaptivity of the proposed framework in addressing the increasing complexity and dynamic nature of problem-solving, by focusing on novel representations of memes as well as the corresponding realizations of the evolutionary mechanisms in eTL, i.e., *meme external evolution*, namely *meme selection*, *meme transmission*, *meme imitation* and *meme variation*.

ACKNOWLEDGMENT

This research is supported by the National Research Foundation Singapore under its Interactive Digital Media (IDM) Strategic Research Programme.

REFERENCES

[1] G. Conte, G. Morganti, A. M. Perdon, and D. Scaradozzi, "Multi-agent system theory for resource management in home automation systems," *Journal of Physical Agents*, vol. 3, no. 2, pp. 15–19, 2009.

[2] D. Dasgupta, "An artificial immune system as a multi-agent decision support system," in *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, vol. 4. IEEE, 1998, pp. 3816–3820.

[3] R. Adobbati, A. N. Marshall, A. Scholer, S. Tejada, G. A. Kaminka, S. Schaffer, and C. Sollitto, "Gamebots: A 3d virtual world test-bed for multi-agent research," in *Proceedings of the second international workshop on Infrastructure for Agents, MAS, and Scalable MAS*, vol. 5. Montreal, Canada, 2001.

[4] D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette, "Evolutionary algorithms for reinforcement learning," *J. Artif. Intell. Res. (JAIR)*, vol. 11, pp. 241–276, 1999.

[5] D. C. Dracopoulos, D. Effraimidis, and B. D. Nichols, "Genetic programming as a solver to challenging reinforcement learning problems," *International Journal of Computer Research*, vol. 20, no. 3, p. 351, 2013.

[6] A. Gosavi, *Simulation-based optimization: parametric optimization techniques and reinforcement learning*. Springer, 2014, vol. 55.

[7] S. P. Singh and R. S. Sutton, "Reinforcement learning with replacing eligibility traces," *Machine learning*, vol. 22, no. 1-3, pp. 123–158, 1996.

[8] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, U.K, May 1989.

[9] J. A. Bagnell and J. G. Schneider, "Autonomous helicopter control using reinforcement learning policy search methods," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 1615–1620.

[10] J. Peters, S. Vijayakumar, and S. Schaal, "Reinforcement learning for humanoid robotics," in *Proceedings of the third IEEE-RAS international conference on humanoid robots*, 2003, pp. 1–20.

[11] A.-H. Tan, N. Lu, and D. Xiao, "Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback," *Neural Networks, IEEE Transactions on*, vol. 19, no. 2, pp. 230–244, 2008.

[12] T.-H. Teng, A.-H. Tan, and J. M. Zurada, "Self-organizing neural networks integrating domain knowledge and reinforcement learning," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 26, pp. 889–902, 2014.

[13] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *The Journal of Machine Learning Research*, vol. 10, pp. 1633–1685, 2009.

[14] M. E. Taylor, N. K. Jong, and P. Stone, "Transferring instances for model-based reinforcement learning," in *Machine learning and knowledge discovery in databases*. Springer, 2008, pp. 488–505.

[15] M. E. Taylor and P. Stone, "Representation transfer for reinforcement learning," in *AAAI 2007 Fall Symposium on Computational Approaches to Representation Change during Learning and Development*, 2007.

[16] B. Banerjee and P. Stone, "General game learning using knowledge transfer." in *IJCAI*, 2007, pp. 672–677.

[17] T. J. Walsh, L. Li, and M. L. Littman, "Transferring state abstractions between mdps," in *ICML Workshop on Structural Knowledge Transfer for Machine Learning*, 2006.

[18] M. E. Taylor and P. Stone, "Cross-domain transfer for reinforcement learning," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 879–886.

[19] G. Boutsioukis, I. Partalas, and I. Vlahavas, "Transfer learning in multi-agent reinforcement learning domains," in *Recent Advances in Reinforcement Learning*. Springer, 2012, pp. 249–260.

[20] A. Taylor, I. Dusparic, E. Galván-López, S. Clarke, and V. Cahill, "Transfer learning in multi-agent systems through parallel transfer," in *Theoretically Grounded Transfer Learning at the 30th International Conference on Machine Learning (ICML). Omnipress*, 2013.

[21] E. Oliveira and L. Nunes, "Learning by exchanging advice," *In R.Khosla, N. Ichalkaranje, and L. Jain, editors, Design of Intelligent Multi-Agent Systems, chapter 9. Spring, New York, NY, USA*, 2004.

[22] L. Feng, Y.-S. Ong, A.-H. Tan, and X.-S. Chen, "Towards human-like social multi-agents with memetic automaton," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE, 2011, pp. 1092–1099.

[23] R. Dawkins, "The selfish gene," *Oxford: Oxford University Press*, 1976.

[24] T. Bäck, U. Hammel, and H.-P. Schwefel, "Evolutionary computation: Comments on the history and current state," *Evolutionary computation, IEEE Transactions on*, vol. 1, no. 1, pp. 3–17, 1997.

[25] J. W. Eerkens and C. P. Lipo, "Cultural transmission, copying errors, and the generation of variation in material culture and the archaeological record," *Journal of Anthropological Archaeology*, vol. 24, no. 4, pp. 316–334, 2005.

[26] M. A. Runco and S. R. Pritzker, *Encyclopedia of creativity*. Elsevier, 1999, vol. 2.

[27] X. Chen, Y.-S. Ong, M.-H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 591–607, 2011.

[28] T. L. Huston and G. Levinger, "Interpersonal attraction and relationships," *Annual review of psychology*, vol. 29, no. 1, pp. 115–156, 1978.

[29] F. Bousquet and C. Le Page, "Multi-agent simulations and ecosystem management: a review," *Ecological modelling*, vol. 176, no. 3, pp. 313–332, 2004.

[30] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer vision, graphics, and image processing*, vol. 37, no. 1, pp. 54–115, 1987.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning representations by back-propagating errors*. MIT Press, Cambridge, MA, USA, 1988.

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2664665, IEEE Transactions on Evolutionary Computation

[32] D. Gordon and D. Subramanian, "A cognitive model of learning to navigate," in *Proc. 19th Conf. of the Cognitive Science Society*, vol. 25, 1997, p. 271.

[33] D. Wang and A. Tan, "Creating autonomous adaptive agents in a real-time first-person shooter computer game," *Computational Intelligence and AI in Games, IEEE Transactions on*, 2014.

[34] Y. Hou, L. Feng, and Y.-S. Ong, "Creating human-like non-player game characters using a memetic multi-agent system," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 177–184.

[35] S. Blackmore, *The meme machine*. Oxford University Press, 2000, vol. 25.

[36] G. Grant, "Memetic lexicon," *Principia Cybernetica Web*, 1990.

[37] J. Delius, "Of mind memes and brain bugs; a natural history of culture," 1989.

[38] D. C. Dennett, *Consciousness explained*. Penguin UK, 1993.

[39] L. Gabora, "The origin and evolution of culture and creativity," *Journal of Memetics: Evolutionary Models of Information Transmission*, vol. 1, no. 1, pp. 1–28, 1997.

[40] Y.-S. Ong, M. H. Lim, and X. Chen, "Research frontier-memetic computation - past, present & future," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, p. 24, 2010.

[41] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," *Evolutionary Computation, IEEE Transactions on*, vol. 9, no. 5, pp. 474–488, 2005.

[42] Y. S. Ong and A. J. Keane, "Meta-lamarckian learning in memetic algorithms," *Evolutionary Computation, IEEE Transactions on*, vol. 8, no. 2, pp. 99–110, 2004.

[43] Y.-S. Ong, M.-H. Lim, F. Neri, and H. Ishibuchi, "Special issue on emerging trends in soft computing: memetic algorithms," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 13, no. 8, pp. 739–740, 2009.

[44] Y. Hou, Y. Zeng, and Y. S. Ong, "A memetic multi-agent demonstration learning approach with behavior prediction," in *International Conference on Autonomous Agents & Multiagent Systems*, 2016, pp. 539–547.

[45] L. Feng, Y.-S. Ong, M.-H. Lim, and I. W. Tsang, "Memetic search with interdomain learning: A realization between cvrp and carp," *Evolutionary Computation, IEEE Transactions on*, vol. 19, no. 5, pp. 644–658, 2015.

[46] Heidrich-Meisner, "Interview with richard s. sutton," *Knstliche Intelligenz*, vol. 3, no. 1, pp. 41–43, 2009.

[47] X. Chen, Y. Zeng, Y.-S. Ong, C. S. Ho, and Y. Xiang, "A study on like-attracts-like versus elitist selection criterion for human-like social behavior of memetic mulitagent systems," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 1635–1642.

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.