

Structured Memetic Automation for Online Human-like Social Behavior Learning

Yifeng Zeng, Xuefeng Chen, Yew-Soon Ong, Jing Tang and Yanping Xiang

Abstract—Meme automaton is an adaptive entity that autonomously acquires increasing level of capability and intelligence through embedded memes evolving independently or via social interactions. This paper embarks a study on memetic multiagent system (MeMAS) towards human-like social agents with memetic automaton. We introduce a potentially rich meme-inspired design and operational model, with Darwin’s theory of natural selection and Dawkins’ notion of a meme as the principal driving forces behind interactions among agents, whereby memes form the fundamental building blocks of the agents’ mind universe. To improve the efficiency and scalability of MeMAS, we propose memetic agents with structured memes in the present paper. Particularly, we focus on meme selection design where the commonly used *elitist* strategy is further improved by assimilating the notion of *like-attracts-like* in the human learning. We conduct experimental study on multiple problem domains and show performance of the proposed MeMAS on human-like social behavior.

Index Terms—Memetic Automaton, Multiagent Systems, Structured Memes, Human-like Behavior

I. INTRODUCTION

MEME has been an important concept in the line of research on evolutionary algorithm as it is coined as an adaptive individual learning procedure that improves local search operators in population based search algorithms. By integrating meme and canonical evolutionary algorithm, memetic algorithm (MA) has attracted increasing attention and exhibited appealing performance on solving a broad set of real world problems, ranging from continuous optimization [1], [2], combinatorial optimization [3], [4], constrained optimization [5] to image processing [6], etc.

Beyond the formalism of simple and adaptive hybrids in MA, Situngkir [7] presents a structured analysis of culture by means of memetics, where meme is considered as the smallest unit of information. Heylighen *et al.* [8] discuss the replication, spread and reproduction operators of memes in cultural evolution. Nguyen *et al.* [9] study the notion of “Universal Darwinism” and social memetics in search, and investigate on the transmission of memetic material via non-genetic means. Meuth *et al.* [10] show the potential of meme learning and high-order memes for more efficient problem

solving while Acampora *et al.* [11] introduce memetic agents as intelligent explorers to create “in time” and personalized experiences for e-Learning. In contrast to memetic algorithms, less study on other manifestations of memes for effective problem solving has been explored, making it a fertile area for further research investigation. This paper thus presents an attempt to reduce this gap.

Recently a comprehensive MA study [12] defines *memetic automaton* as an adaptive entity or agent that is self-contained and uses memes as the building blocks of information that facilitates problem-solving. Conceptualization of memetic automaton unleashes the significant number of potentially rich meme-inspired designs, operational models, and algorithm frameworks that could form the cornerstones of memetic computation as tools for effective problem-solving. In the present study, a memetic multiagent system (MeMAS) [13], [14] is developed and the infrastructure development of memetic agents is based on the *temporal difference - fusion architecture for learning and cognition* (TD-FALCON) [15]. The MeMAS development has shown much benefit on solving complex problems like the navigation problem in the minefield domain [14], and so on. The benefit lies in the capability of memetic agents on acquiring proper memes and learning from each other in a complex setting.

To achieve adaptation to the ever changing environments, memetic agents need to act promptly upon the relevant knowledge that is encapsulated into meme blocks. Hence the meme search is critical to facilitate the actions of memetic agents. Since the number of meme blocks grows significantly in a complex problem domain, the meme search becomes inefficient on providing decision support. Following the line of biologically inspired representation [16], we propose memetic agents with structured memes to speed up decision making of the memetic automatons. The representation implies that humans learn a complex concept by first identifying simple and abstract ones and then composing them together. Similarly we organize memes in a hierarchical way and maintain the memes in different abstraction levels. An abstract meme is placed at the top level while a specific level is in the bottom level. The hierarchical memes provide one efficient way to locating proper memes once agents conduct the search in the stored meme blocks.

With an efficient search strategy, memetic agents can quickly exchange information across individuals and learn to enhance their capabilities. We make a further step to investigate the learning mechanism in the MeMAS especially on whom shall be selected for the learner memetic agents. Following the commonly used elitist principle, the MeMAS

Yifeng Zeng is with Department of Automation, Xiamen University, China; School of Computing, Teesside University, UK; Email: Y.Zeng@tees.ac.uk.

Xuefeng Chen and Yanping Xiang are with School of Computer Science and Engineering, University of Electronic Science and Technology of China, China; Email: {cxflvechina, xiangyanping}@gmail.com.

Yew-Soon Ong is with School of Computer Engineering, Nanyang Technological University, Singapore; Email: asysong@ntu.edu.sg.

Jing Tang is with Industry Technology Systems Ltd., UK; Email: amysurfing@gmail.com.

drives agents' learning through the rewarding mechanism - agents always learn from a successful teacher agent of highest rewards. However, we notice that in practice, rewarding is not the sole element to drive the learning of memetic agents. As one part of social interactions, the notion of *like-attracts-like* has a significant influence on interpersonal attraction which is one of the core principles behind the human learning [17]. With this observation, we propose the *like-attracts-like* based interactions and develop human-like social behavior of memetic agents. The proposed technique is geared towards the increasingly popular design of human-like agents in the agent-related research [18]. We finally evaluate the improved MeMAS in the minefield navigation domain [14] as well as one 3D interactive computer game.

The core contributions of the present paper is summarized as follows: In section II, the meme representation for a basic data structure or building block and evolutionary mechanisms including meme expression, meme assimilation, meme internal/external evolution, of MeMAS are described. The concept of structured memes in MeMAS which form the mind universe of an agent and its motivation are then presented in section III. In contrast to the previous works [19], [20], the organization of memes in a hierarchical way helps elevate the increasing complexity of growing memes in the mind universe. Special attention on the study of different selection mechanism on the evolution structured meme is studied in section IV. Last but not least, a comprehensive study of the proposed MeMAS with structured memes is presented to showcase its potentials for online human-like social behavior learning in multiple problem domains.

II. MEMETIC MULTIAGENT SYSTEM

As memetic multiagent systems (MeMAS) are the basic framework in our work, we elaborate relevant concepts in this section. More details could be found in [14]. The MeMAS architecture is depicted in Fig. 1, where a population of memetic agents learn and evolve in a dynamic environment.

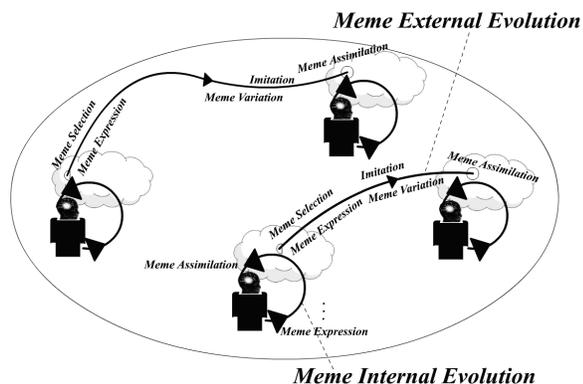


Fig. 1. Illustration of Memetic Multiagent Systems

MeMAS contains a basic data structure, namely *meme representation*, and four functions, namely *meme expression*, *meme assimilation*, *meme internal evolution* and *meme external evolution*, that control knowledge exchange and evolution of individual agents. *Meme representation* uses meme to

represent internal knowledge of individual agents and defines mind-universe of the agents [21]. *Meme expression* translates the knowledge into a set of behaviors that are (partially) in an environment. Meanwhile, *Meme assimilation* converts observed behaviors into internal knowledge blended into individual mind-universe.

Meme evolution is central to behavioral aspects of memetic automaton. It includes *meme internal evolution* and *meme external evolution*. *Meme internal evolution* is a process where individual agents grow their mind-universe through self-learning. *Meme external evolution* models agents' interaction, which is primarily driven by imitation [22]. In *meme external evolution*, *meme selection* determines an appropriate teacher agent to learn from, while *meme transmission and variation* relates to how one imitates and what is imitated. Note that *meme variation* process models innovative characteristics of interactions between agents.

A. Meme Representation

Meme representation is the first step in memetic computation. Internally, the meme (memotype) defines the building blocks of the cognitive space in an agent's mind universe. Externally, the meme (sociotype) manifests as the agent's expressed or observed behavior [8]. In the subsequent sections, we first define the model of the memetic agent, and then investigate the manifestations of memotype as neural meme embodiments and sociotype as behavior exhibited by the agents.

The TD-FALCON [15] models the mind universe of a memetic agent in the form of three-channel neural network architecture. As depicted in Fig. 2, the TD-FALCON model is consisted of four components: a sensory field F_1^c1 for representing current states, a action field F_1^c2 for representing available actions, a reward field F_1^c3 for representing the feedback received from the environment, a cognitive field F_2 for the acquisition and storage of memes, which encodes a relation among the patterns in the three input channels. One node in the field represents a specific meme in the mind universe.

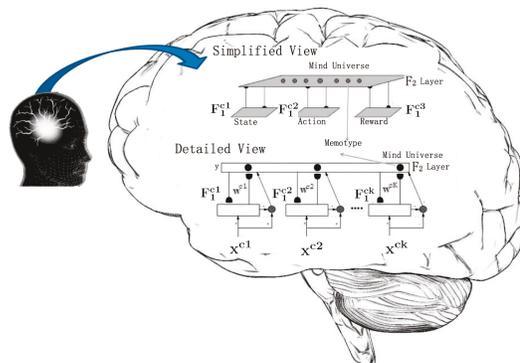


Fig. 2. Neural Network Architecture of Each Memetic Agent in Figure 1

Input vector: Let $IV = \{S, A, R\}$ be the input vector, where $S = (s_1, s_2, \dots, s_n)$ is the state input, and s_i indicates the value of sensory input i ; $A = (a_1, a_2, \dots, a_m)$ is the action

vector, and a_i indicates a possible action i ; $\mathbf{R} = (r, \bar{r})$ is the reward vector, and r is the reward signal value and $\bar{r} = 1 - r$ (the complement of r).

Activity vector: Let \mathbf{x}^{ck} be the \mathbf{F}_1^{ck} activity vector for $k = 1, 2, 3$, while \mathbf{y} is the output of \mathbf{F}_2 layer activity neurons. Activity vectors \mathbf{x}^{c1} , \mathbf{x}^{c2} and \mathbf{x}^{c3} are the input state \mathbf{S} , action \mathbf{A} and reward \mathbf{R} , respectively.

Cognitive weight: Let \mathbf{w}_j^{ck} be the cognitive weight associated with the j^{th} neuron in layer \mathbf{F}_2 for learning the input patterns of \mathbf{F}_1^{ck} . Initially, each agent is in the blank state and \mathbf{F}_2 contains only one uncommitted neuron. An uncommitted neuron is able to encode any meme, with the initial weight vector to 1s.

1) *Memotype*: In MeMAS, the memotype is defined as the meme inhabiting within the mind universe of a memetic agent and encoding learned semantic rule mappings between world states and actions. It is stored in the cognitive field \mathbf{F}_2 , and forms the knowledge of the agent denoting the associated patterns of the input channels.

2) *Sociotype*: The sociotype meme of an agent refers to its expressed action or behavior, which can be observed and imitated by other agents. Typically, when an agent observes or acquires a meme in its sociotype representation, it will assimilate the sociotype meme for updating the existing memotype memes or creating new memotype memes by an inference derivation approach.

B. Meme Expression

Meme (memotype) activation: A bottom-up propagation process first takes place in which the activities of all the memes (memotype) in the \mathbf{F}_2 field are computed. Specifically, given the activity vectors \mathbf{x}^{c1} , \mathbf{x}^{c2} , \mathbf{x}^{c3} , for each meme j in \mathbf{F}_2 , the activation value T_j is computed below.

$$T_j = \sum_{k=1}^3 \gamma^{\text{ck}} \frac{\mathbf{x}^{\text{ck}} \{ \mathbf{w}_j^{\text{ck}} \}}{\alpha^{\text{ck}} + \mathbf{w}_j^{\text{ck}}} \quad (1)$$

where the fuzzy AND operation $\{ \}$ is defined by $(\mathbf{p} \{ \mathbf{q} \} \leq \min(p_i, q_i)$, and the norm \times is defined by $\mathbf{p} \times \mathbf{q} = \sum_i p_i q_i$ for vector \mathbf{p} and \mathbf{q} . Parameter α^{ck} and γ^{ck} are predefined by users, k denotes the index of input channel.

Meme (memotype) competition: Meme competition identifies the \mathbf{F}_2 layer neuron or the encoded memotype with the highest activation value after meme activation. The system is said to make a choice when at most one \mathbf{F}_2 memotype is active. The winner is indexed at J , where

$$T_J = \max\{T_j : \text{for all node } j \text{ in } \mathbf{F}_2\} \quad (2)$$

when a category choice is made at meme J , y_J is equal to 1; otherwise, y_i is 0 for all $i \neq J$.

Sociotype readout: The chosen neuron J in layer \mathbf{F}_2 performs a readout of its action into the input fields \mathbf{F}_1^{ck} below.

$$\mathbf{x}^{\text{ck}(\text{new})} = \mathbf{x}^{\text{ck}(\text{old})} \{ \mathbf{w}_J^{\text{ck}} \} \quad (3)$$

The resulting \mathbf{F}_1^{ck} activity pattern or sociotype is thus the fuzzy AND (as defined in Eq. 1) of $\mathbf{x}^{\text{ck}(\text{old})}$ and \mathbf{w}_J^{ck} .

C. Meme Assimilation

Memotype matching: Before the agent uses the meme J for the learning purpose, it checks whether the weights of meme J are sufficiently close to their respective input patterns by a memotype matching process. Specifically, a resonance occurs if, for a channel k , the matching function m_J^{ck} of the chosen meme J meets the vigilance criterion ρ^{ck} :

$$m_J^{\text{ck}} = \frac{\mathbf{x}^{\text{ck}} \{ \mathbf{w}_J^{\text{ck}} \}}{\mathbf{x}^{\text{ck}}} \simeq \rho^{\text{ck}} \quad (4)$$

where ρ^{ck} is the vigilance parameters and k is the index of input channel.

When a resonance occurs, the memotype is updated below. If any of the vigilance constraints is violated, a mismatch reset occurs in which the activation value T_J is set to 0 for the duration of the input presentation. The search process then selects another meme J in \mathbf{F}_2 until a resonance is achieved.

Memotype update: Once a proper meme J is selected for firing, for each channel ck , the weight vector or memotype \mathbf{w}_J^{ck} is updated by the following learning rule.

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

The learning rate parameters β^{ck} is typically set to 1 when an uncommitted meme is selected; otherwise, it can remain as 1 for fast learning or below 1 for slow learning in a noisy environment. An uncommitted meme will be committed when it is selected for learning, and then a new uncommitted neuron is added to the \mathbf{F}_2 field. Hence the memetic agent can expand its network architecture dynamically in response to the input patterns.

D. Meme Internal Evolution

Meme internal evolution is the self-learning process of memetic agents and consists of a sequence of trials and learning. The evolution is summarized in Algorithm 1. In a sense-act-learn cycle, an agent first predicts the reward values through *meme activation*, *meme competition* and *sociotype readout* with the current state \mathbf{s} . Subsequently, the agent uses the predicted reward values to select an action a_I according to an ϵ -greedy strategy. Next, the agent performs a_I and receives a feedback from environment. Based on the feedback, the agent uses a temporal difference (TD) formulation (i.e., Q-Learning) [15] to estimate reward values. Finally, the agent does meme assimilation with the current state, the performed action and the estimated reward values, and then updates the current state for the next sense-act-learn cycle.

To balance exploration and exploitation in meme internal evolution, an ϵ -greedy action selection scheme is used in MeMAS (in Step 1(b)). It selects an action of the highest $Q(\mathbf{s}, a)$ value with probability $1 - \epsilon$ ($0 \geq \epsilon \geq 1$), or chooses a random action otherwise. In addition, the value of ϵ gradually decays with time.

Meanwhile, the new reward $Q^{\text{new}}(\mathbf{s}, a)$ is estimated by means of a temporal difference (bounded Q-learning) in an iterative process.

$$Q^{\text{new}}(\mathbf{s}, a_I) = Q(\mathbf{s}, a_I) + \Delta Q \quad (6)$$

Algorithm 1 Meme Internal Evolution

- 1: **Do** action selection:
 - a. Given the current state s , predict the reward for each possible action by

$$Q(s, \mathbf{a}_i) = \text{Predict}(s, \mathbf{a}_i)$$
 - b. Select an action a_I following a ϵ -greedy action selection scheme

$$a_I = \epsilon\text{-greedy}(Q(s, \mathbf{a}))$$
- 2: **Perform** a_I :

$$\{s', r\} = \text{Perform}(\mathbf{a}_I)$$

where s' is the resultant state, r is the reward (if any) from the environment.
- 3: **Do** meme assimilation:
 - a. Estimate the reward $Q^{new}(s, \mathbf{a}_I)$ by Eq. 6
 - b. Do *meme activation* with vector $\{\mathbf{S}, \mathbf{A}, \mathbf{R}\}$
 - c. Do *meme competition* with vector $\{\mathbf{S}, \mathbf{A}, \mathbf{R}\}$
 - d. Do *memotype matching* with vector $\{\mathbf{S}, \mathbf{A}, \mathbf{R}\}$
 - e. Do *memotype update* with vector $\{\mathbf{S}, \mathbf{A}, \mathbf{R}\}$

$/*\{\mathbf{S}, \mathbf{A}, \mathbf{R}\}$ consists of state (s), action (a_I), and reward ($Q^{new}(s, \mathbf{a}_I)*/$
- 4: **Update** the current state by $s = s'$.
- 5: **Repeat** from Step 1 until s is a terminal state.

where the temporal-difference ΔQ is computed by:

$$\Delta Q(s, a_I) = \alpha TD_{err}(1 - Q(s, a_I)) \quad (7)$$

where $\alpha / [0, 1]$ is the learning parameter, and the temporal-error TD_{err} is calculated by the Q-learning rule below.

$$TD_{err} = r + \gamma \max_{a'} Q(s', a') - Q(s, a_I) \quad (8)$$

where r is the immediate-reward value, $\gamma / [0, 1]$ is the discount parameter, and $\max_{a'} Q(s', a')$ denotes the maximum predicted value of the next state s' .

Hence the reward for memetic agent is in Eq. 9.

$$\mathbf{R} = (Q^{(new)}(s, a_I), 1 - Q^{(new)}(s, a_I)) \quad (9)$$

E. Meme External Evolution

In MeMAS, the meme external evolution is the process that agents interact with each other through the imitation. It is governed by the three principles of the Universal Darwinism [23], namely *meme selection*, *meme transmission* and *meme variation*.

1) *Meme (Societal) Selection*: Inspired by Darwin's principle of natural selection, meme selection serves to identify the teacher agents that an agent shall learn from. Through this mechanism, memes that are helpful for problem solving are replicated exponentially, while less helpful memes are rarely replicated. In particular, *imitate-the-elite* [24] is one of the more popular strategies for sociotype meme selection. The selection follows the computation below.

$$Agt_j = \text{argmax}_j F(Agt_j) \quad (10)$$

where $F(Agt_j)$ is the fitness of the agent Agt_j .

Notice that knowledge of the elite agent may not be useful to other agents since agents may have different attributes. Taking this cue, we propose a novel yet natural mechanism called *like-attracts-like* to improve sociotype meme selection in section IV.

2) *Meme Transmission via Imitation*: When agents are not familiar with each other, they communicate socially by imitation. As illustrated in Fig. 3, meme transmission is the process that an agent observes the sociotype that is expressed by its teacher agent. In this way, the agent is able to imitate behavior of its teacher agent. Meanwhile, variation may occur during the imitation process, which will be detailed next.

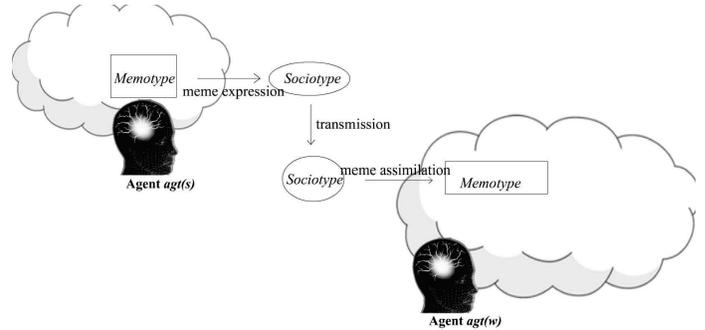


Fig. 3. Example of Imitation between Two Agents

3) *Meme Variation*: Meme variation forms the intrinsic innovation tendency in the mind universe of an agent during cultural evolution, and retains more diversity in the agents' attitudes towards learning and innovation. For knowledge (meme) transmission without variation, bias will be introduced since a deterministic approach may cause every agent to believe that a piece of knowledge (meme) is good based on its particular demonstration of success at a given time instance. Due to nonlinear interactions among agents, an initial bias can quickly spread out of control as it infects any agent it comes into contact with. This will suppress the agents' ability in the learning and innovation. Hence, meme variation plays a key role in reflecting human-like interactions among agents.

Meme variation can occur during the *meme transmission* and *meme assimilation* stages. For simplicity, we only consider the variation at the *meme assimilation* stage in our memetic model. In particular, meme variation is implemented by means of perturbation, i.e., a random value is added to the action's Q value. This would lead to different actions being selected in the state transmission.

$$Q^t = \eta \equiv Rand + (1 - \eta) \equiv Q \quad (11)$$

where Q^t is the mutated Q value, η is the parameter controlling the degree of randomness, and $Rand$ is a random with uniform distribution in the range $[0, 1]$. A predefined probability τ is used to control the variation frequency.

Based on the three principles, the process of meme external evolution is depicted in Algorithm 2.

F. Memetic Multiagent System

Algorithm 3 outlines the basic steps in MeMAS. First, a memetic agent team is initialized. Subsequently, each agent of the team performs *meme external evolution* with a probability of C_p or *meme internal evolution* with a probability of $1 - C_p$.

Algorithm 2 Meme External Evolution

- 1: **Do** meme selection by agent $agt(stu)$ to pick out the teacher agent $agt(tea)$.
- 2: **Pass** $agt(stu)$'s current state s to $agt(tea)$ for getting the imitated action $a(tea)$ of $agt(tea)$ through:
 - a. Do Meme (memotype) activation with s
 - b. Do Meme (memotype) competition with s
 - c. Do Sociotype readout
- 3: **Do** meme transmission with $a(tea)$
- 4: **Imitate** $a(tea)$ by $agt(stu)$:

$$\{s', r\} = \text{Perform}(a(tea))$$
 where s' is the resultant state, r is the reward (if any) from the environment.
- 5: **Do** meme assimilation by $agt(stu)$:
 - a. Estimate the reward $Q^{new}(s, a(tea))$ based on Eq. 6
 - b. Do meme activation with vector $\{S, A, R\}$
 - c. Do meme competition with vector $\{S, A, R\}$
 - d. Do memotype matching with vector $\{S, A, R\}$
 - e. Do memotype update with vector $\{S, A, R\}$
$$/*\{S, A, R\} \text{ consists of state } (s), \text{ action } (a(tea)), \text{ and reward } (Q^{new}(s, a(tea)))*/$$
- 6: **Update** $agt(stu)$'s current state by $s = s'$.
- 7: **Repeat** from Step 1 until s is a terminal state.

until the termination conditions are satisfied. C_p is computed based on the agent's performance in Eq. 12.

$$C_p(agt(w)) = 1 - \frac{F(agt(w))}{F_{best}} \quad (12)$$

where $F(agt(w))$ is the fitness of agent $agt(w)$ and is defined as the number of times that agent $agt(w)$ accomplishes a task successfully, and F_{best} is the fitness of the elite agent. Thus an agent $agt(w)$ shall perform imitation with a probability of $C_p(agt(w))$.

Algorithm 3 Implementation of Memetic Multiagent System

- 1: **Initialization:** Generate a memetic agent team
- 2: **While** (Stopping conditions are not satisfied)
- 3: **For** each agent
- 4: Compute the probability C_p in Eq. 12
- 5: **If** ($rand < C_p$)
- 6: Perform meme external evolution
- 7: **Else**
- 8: Perform meme internal evolution
- 9: **End If**
- 10: **End For**
- 11: **End While**

III. MEMETIC AGENTS WITH STRUCTURED MEMES

As an information and knowledge building block in MeMAS, memes are maintained and selected while memetic agents interact with others in the changing world. In general, the memetic agents need to search the entire meme space in order to identify a proper meme for decision support. The search becomes inefficient when the space grows due to the problem complexity. To alleviate the problem, we propose to structure the memes in a hierarchical way. The memes are organized in different levels: the top level maintains the most abstraction memes while the bottom one stores concrete memes. The development is motivated by the biologically inspired representations and aligned with the practical ways

of humans on organizing and retrieving the knowledge in the brain.

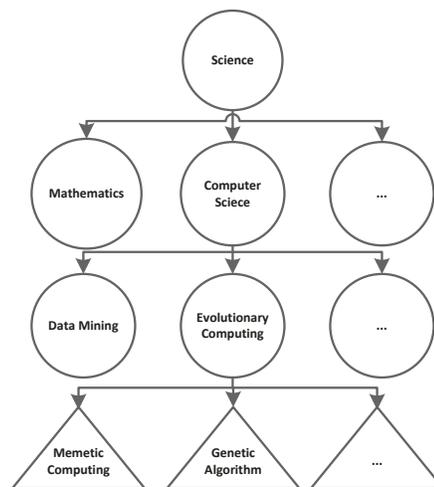


Fig. 4. Structured knowledge in the brain. Internal nodes (circles) represent memes of abstraction concepts whereas leaves (triangles) are memes of specific concepts.

A. Meme Assimilation with Structured Memes

Fig. 4 shows one classical way that humans structure knowledge (in terms of science study) in the brain. Abstract knowledge is organized in a high level while specific one is placed in a bottom level. Inspired by the structured knowledge, memetic agents build the structure of memes by classifying the memes into different classes in a hierarchical way. The basic data structure of the structured memes is a tree in which internal nodes are the tags of classes and leaves are memes. In internal nodes, their children are the subclasses, and information of one internal node contains its children's common characteristics.

As memes are organized in a tree structure, we need to adapt the meme assimilation process in a hierarchical way. To improve the efficiency of meme assimilation, a three-phases assimilation method is proposed according to the thinking process of humans (Algorithm 4). At the first phase (rough search), the search process quickly finds the best and smallest active subclass $BSSubclass$ by selecting the best active subclass to iterate the search further (lines 1-6). At the second phase (careful search), meme activation, meme competition and meme matching are used to find the best matching meme from the best and smallest active subclass (lines 7-9). At the third phase (memotype update), the memetic agent updates its structured memes based on the matching result (lines 10-16). If the current vector $\{S, A, R\}$ matches an existing meme successfully, memotype update is done for the best matching meme. Otherwise, a new meme is created and inserted into the structured memes. Analogous to that humans reorganize a large class of knowledge including a large amount of knowledge, memetic agents split their big classes containing more than $NumClassSplit$ memes. Finally, the best matching (or new) meme's ancestors are updated by memotype update (line 17).

Algorithm 4 Memotype Assimilation with Structured Memes

```

1:  $BSSubclass \leftarrow$  Root of StructuredMeme
2: While ( $BSSubclass.children$  are not leaves)
3:   Do Meme activation for all  $BSSubclass.children$ 
   with vector  $\{S, A, R\}$ 
4:   Do Meme competition for all  $BSSubclass.children$ 
   with vector  $\{S, A, R\}$ 
5:    $BSSubclass \leftarrow$  the best active subclass of
    $BSSubclass$ 
6: End While
7: Do Meme activation for all  $BSSubclass.children$  with vector
 $\{S, A, R\}$ 
8: Do Meme competition for all  $BSSubclass.children$  with vector
 $\{S, A, R\}$ 
9: Do Memotype matching for all  $BSSubclass.children$  with
vector  $\{S, A, R\}$ 
10: If (an existing meme is matched successfully)
11:   Do memotype update for the best matching meme
   with vector  $\{S, A, R\}$ 
12: Else
13:   Create a new meme for vector  $\{S, A, R\}$  and insert
   it into  $BSSubclass$ 
14:   If ( $BSSubclass.size > NumClassSplit$ )
15:     Do class splitting for  $BSSubclass$ 
16: End If
17: Update the best matching (or new) memes ancestors with mem-
otype update

```

In lines 13-14 of Algorithm 4, a memetic agent does *class splitting* to re-organize the memes in a large class. As shown in Algorithm 5, *class splitting* divides the large class *SplitClass* into some small ones by rebuilding it with an adaptive vigilance criterion ρ_{clasp}^{ck} . Normally, ρ_{clasp}^{ck} should increase with the depth d in *StructuredMeme* and range from 0 to 1, thus *StructuredMeme* is to be built in a hierarchical way (line 2). In addition, ρ_{clasp}^{ck} should be less than the baseline vigilance parameter ρ^{ck} . Hence, ρ_{clasp}^{ck} is computed below.

$$\rho_{clasp}^{ck} = \min[\rho^{ck}, 1 - (\rho_{iniclasp}^{ck})^d] \quad (13)$$

where $\rho_{iniclasp}^{ck} / [1 - \rho^{ck}, 1)$ is the parameter for adjusting ρ_{clasp}^{ck} . Based on ρ_{clasp}^{ck} , *meme activation*, *meme competition* and *memotype matching* are used to find the best matching subclass for meme J in *NewSplitClass* (lines 4-6), if the best matching subclass is found successfully, meme J is inserted into it; otherwise, a new class containing meme J is created and inserted into *NewSplitClass* (lines 7-11). At the end, *SplitClass* is replaced by *NewSplitClass* (line 14) and the *StructuredMeme* is re-organized.

B. Meme Expression with Structured Memes

Similar to *meme assimilation*, *meme expression* adopts a hierarchical approach in the structured memes (in Algorithm 6). To search for the best matching meme, the search process starts from the root of *StructuredMeme* (line 1). At every level, the search process utilizes *meme activation* and *meme competition* to choose the best matching node *BestMatchMeme* until the best matching meme is found (lines 2-6). After that, *sociotype* is got by *sociotype readout* with the best matching meme (line 7).

Algorithm 5 Class Splitting

```

1: Initialize a structured memes NewSplitClass with a root node
   in which the date is same to SplitClass's root
2: Compute  $\rho_{clasp}^{ck}$  based on Eq. 13 for each input channel
3: For each meme  $J$  in SplitClass
4:   Do Meme activation for all SplitClass.children
   with meme  $J$ 
5:   Do Meme competition for all SplitClass.children
   with meme  $J$ 
6:   Do Memotype matching for all SplitClass.children
   with meme  $J$ 
7:   If (an existing subclass is matched successfully)
8:     Insert meme  $J$  into the best matching subclass
9:     Update meme  $J$ 's ancestors by memotype update
10:  Else
11:    Create a new class which contain meme  $J$  and
    insert it into NewSplitClass
12:  End If
13: End For
14: Use NewSplitClass to replace SplitClass in
   StructuredMeme

```

Algorithm 6 Memotype Expression with Structured Memes

```

1: BestMatchMeme  $\leftarrow$  Root of StructuredMeme
2: While (BestMatchMeme is not a leaf)
3:   Do Meme activation for all children of
   BestMatchMeme with vector  $\{S, A, R\}$ 
4:   Do Meme competition for all children of
   BestMatchMeme with vector  $\{S, A, R\}$ 
5:   BestMatchMeme  $\leftarrow$  the best active subclass of
   BestMatchMeme
6: End While
7: Get sociotype by do sociotype readout with
   BestMatchMeme

```

IV. LIKE-ATTRACTS-LIKE VERSUS ELITISM PRINCIPLE AS SELECTION CRITERION

With structured memes in MeMAS, we proceed to improve *meme selection* in order to stimulate human-like social behavior of memetic agents. One central issue in the *meme external evolution* is *meme selection* that picks out a teacher agent for the imitation purpose. As one of the most popular strategies for selecting a teacher agent, imitate-the-elite is adopted in MeMAS (Eq. 10). However, with this scheme, the agents focus on the elite pool and measure the experience in terms of fitness values, which has the tendency of a biased selection towards only the elite agents.

As humans are prone to imitate others of a similar type, a memetic agent shall also manifest such human-like social behavior when it is selecting a teacher agent. The selection shall consider not only solution performance in terms of agents' fitness, but also the evolution origin in terms of agents' personal attributes. Thus, in contrast to using an imitate-the-elite scheme, we adopt the like-attracts-like principle of agents' experiences in MeMAS. For this purpose, we first define the similarity measurement of agents' memotype, and then implement the selection setting.

A. Selection Criterion

As defined in the TD-FALCON model, a memotype, denoted by $Q=(S, A, R)$, essentially encodes a mapping between

input states, \mathbf{S} , and actions, \mathbf{A} , through the reward measurement, \mathbf{R} . It models how a memetic agent responds to a sensory input. The basic definition loses an important connection between a meme and its genetic origin in the concept of meme automaton. Analogous to that a meme is in part regulated by its gene, behavior of memetic agents may be often determined by their original properties. Accordingly, we expand a memotype with agents' attributes denoted by Θ . Formally, the augmented memotype is defined as: $MT = \langle \Theta, Q \rangle$. The next issue is on how to measure the similarity among agents' memotypes.

As agents' attributes, Θ , normally have a numerical scale, we use normalized Euclidean distance to measure the similarity between two attribute sets Θ and Θ' .

$$S_{ED}[\Theta, \Theta'] = 1 - \frac{\sum_{\theta \in \Theta, \theta' \in \Theta'} \sqrt{\sum_{\theta_{dim} \in \Theta, \theta'_{dim} \in \Theta'} (\theta_{dim} - \theta'_{dim})^2}}{\Theta} \quad (14)$$

where θ_{dim} (or θ'_{dim}) is normalized value whose range is $[0,1]$, for a single attribute, θ (or θ'), in the sets, and Θ is the cardinality of Θ . Note that Eq. 14 computes the common set of Θ and Θ' .

Considering the bounded rationality of an agent, we use probabilistic models to define its behavior. Formally, let $Pr(A|S)$ be a set of probability distributions over actions given the input of world states. To measure the distance between Q and Q' , we resort to Kullback-Leibler (KL) divergence [25] in Eq. 15.

$$D_{KL}[Q||Q'] = Pr_Q(A|S) \ln \frac{Pr_Q(A|S)}{Pr_{Q'}(A|S)} \quad (15)$$

We further adapt Eq. 15 and define a symmetric measurement of similarity between Q and Q' in Eq. 16. And the similarity value is scaled within $[0,1]$.

$$SD_{KL}[Q, Q'] = e^{-\frac{1}{2}\{D_{KL}[Q||Q'] + D_{KL}[Q'||Q]\}} \quad (16)$$

As a memetic agent may be featured by both attributes and behavior, the similarity between agents, Agt_i and Agt_j , is subsequently computed in Eq. 17.

$$SIM(Agt_i, Agt_j) = \frac{S_{ED}[\Theta, \Theta'] + SD_{KL}[Q, Q']}{2} \quad (17)$$

Driven by the like-attracts-like principle, Agt_i may select Agt_j that has the largest $SIM(Agt_i, Agt_j)$ value. On the other hand, Agt_i may also expect to learn from an elite agent that has more sophisticated skills. We use the fitness ratio, $\frac{F(Agt_j)}{F_{best}}$, to measure how well Agt_j approaches the best one. Consequently, Agt_i will select an agent as the teacher that has the largest value for the combined measurements of $SIM(Agt_i, Agt_j)$ and $\frac{F(Agt_j)}{F_{best}}$. Formally, the selection criterion is defined in Eq. 18.

$$Agt_j = \operatorname{argmax}_j \quad K_1 \equiv SIM(Agt_i, Agt_j) + K_2 \equiv \frac{F(Agt_j)}{F_{best}} \quad (18)$$

where K_1 and K_2 are parameters balancing the similarity and elitist factors.

Note that if the selection is solely driven by the like-attracts-like principle, Agt_i loses chance to explore the entire solution space. More importantly, by learning from a distinct type of agents, Agt_i may update the memotype particularly on speeding up the co-evolution by recognizing its genetic origin and relating it to other types.

B. Parameter Settings

As an agent does not act individually in the environment, its selection on a teacher agent may be influenced by evolutions of other agents in MeMAS. Particularly, the dynamic properties of MeMAS may impact the trade-off between the aforementioned two principles: like-attracts-like and elitist. We make a further step to illustrate the settings of K_1 and K_2 .

Intuitively, the similarity factor may place an important role in the agent's selection if there are dominating groups of similar agents in the MeMAS. Otherwise, the factor may become weak if all agents are equally similar. Under the thoughts of this vein, we may specify K_1 as the diversity value of the MeMAS that measures the uncertainty of different groups of agents for a population of agents.

Resorting to regular clustering techniques like k -means [26], we group N agents into l clusters ($\langle C_1, \dots, C_l \rangle$) in terms of similarity measurements. Each group contains a number of agents that have similar memotypes. To compute the diversity of agent groups in the MeMAS, we use normalized information entropy as defined below. K_1 is proportional to the entropy value in Eq. 19.

$$K_1 \in \frac{\sum_l \frac{|C_l|}{N} \ln\left(\frac{|C_l|}{N}\right)}{\ln C_l} \quad (19)$$

where $\frac{|C_l|}{N}$ is the ratio of the size of cluster C_l to the MeMAS space.

We perceive that the setting of K_2 depends on distributions of agents' skills in MeMAS. Naturally an agent may pick out a teacher agent depending on the similarity of candidate agents if all of them are elite. In other words, the elitist principle may have a small impact on the selection when there is little divergence of the skill levels for all agents. We compute the variance of all agents' fitness values, and let K_2 be the proportion of the standard deviation in Eq. 20.

$$K_2 \in \operatorname{stdev}[F(Agt_j \quad j = 1, \dots, N)] \quad (20)$$

Example 1 (Parameter Setting): In Fig. 5, the MeMAS contains a set of agents that have different distributions of fitness values and types in terms of similarity. Fig. 5(b) shows that the agents are grouped into three clusters in terms of the similarity. In addition, the fitness values are distributed over an entire scale. Hence, both the similarity and elitist factors have a competitive impact on the selection. As there is only one group of agents in Fig. 5(a), the diversity of MeMAS approaches zero. Consequently, the single factor of the elite is counted in the selection. Similarly, in Fig. 5(c), most agents

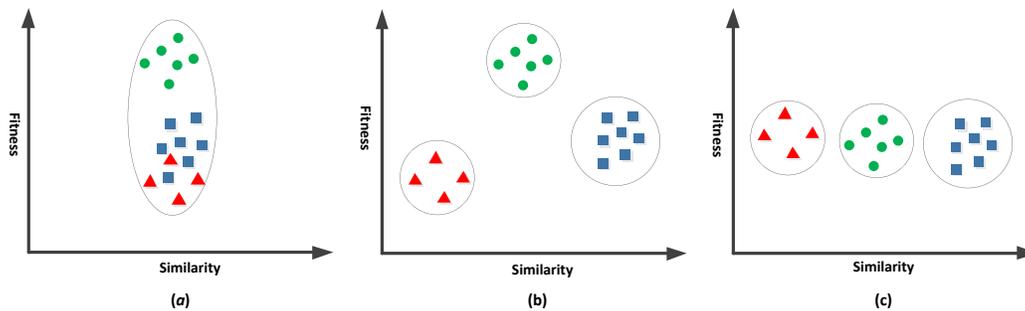


Fig. 5. Distributions of fitness values and types of agents impact the settings of K_1 and K_2 : (a) $K_1 \approx 0$, $K_2 > 0$; (b) $K_1 > 0$, $K_2 > 0$; (c) $K_1 > 0$, $K_2 \approx 0$.

have similar fitness values, selecting a teacher agent mainly depends on the similarity factor.

Given no prior knowledge on the MeMAS, we normally initialize K_1 and K_2 as equal in the initial phase. After each evolution, we compute K_1 and K_2 online, and the new values manifest the updated MeMAS state. When K_1 approaches 0, the new selection criterion completely follows the imitate-the-elite strategy as the memotype similarity no long plays any role in the selection.

V. EXPERIMENTAL RESULTS

We first evaluate the effectiveness and efficiency of the proposed memetic agents with structured memes in the minefield navigation task (MNT) [15]. Subsequently, to show the social behavior of memetic agents, we develop two domains: one is the adapted version of minefield navigation task (AMNT) and the other is a 3D interactive game on the homeland defense. With the two domains, we study the new selection criterion with the comparison to the conventional elitism selection criterion in the MeMAS framework. To conduct the subject study on both MeMAS frameworks, we invite human players to play with the memetic agents in the games and evaluate several human-like properties of the memetic agents.

A. Minefield Navigation Domain and Structured MeMAS

As a motivating domain, the minefield navigation task is well studied in the previous articles [15], [19], [20]. We start to briefly describe the problem domain, and then investigate parameter settings of MeMAS by balancing the trade-off between solution quality and efficiency. Subsequently, we examine performance of the MeMAS framework compared to a state-of-art multiagent learning approach. This justifies the MeMAS use in this work. Finally, we show benefits of MeMAS with structured memes on improving the solution efficiency.

1) *Domain Description*: In the minefield navigation task, a number of autonomous vehicles aim to navigate through a minefield to reach a target safely within a specified time period (in Fig. 6). In each trial, the starting points of vehicles, the target and the mines are randomly generated in the map, and the target and the mines remain stationary during a trial, thus the vehicles should repeat the cycles of sense, act, and learn

to arrive at the target safely. A trial is terminated once either all the vehicles reach the target (success), hit a mine (failure), collide with another one (failure), or 30 sense-act-learn cycles are run out.

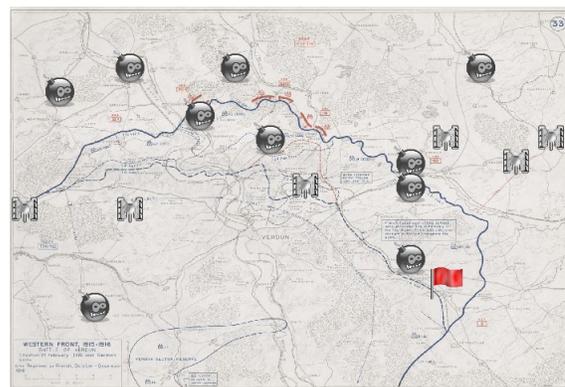


Fig. 6. Minefield Navigation Task

Since the vehicles do not have any priori knowledge on the location of the mines, the target and their companions, they need to use their equipped sonar sensors to detect the environment. The sonar sensors have a rather coarse sensory capability with a 180° forward view and can detect the positions of mines, the target and other vehicles as agents' input (state). For each direction i of a sonar sensor, the sonar signal is measured by $s_i = \frac{1}{d_i}$, where d_i is the distance to an obstacle (that can be either a mine or the boundary of the minefield) in the i th direction. To the signal of mines and other vehicles, s_i will be set as 0 if s_i is smaller than 1. The vehicles move from one cell to another by selecting the five possible actions, namely, proceed left, diagonally left, forward move, diagonally right, and right at each discrete time step. After taking an action, vehicles will receive an evaluation feedback (reward). The reward scheme is described as follows: in the case that a trial of the vehicle ends, if the vehicle reaches the target, a reward of 1 is assigned; otherwise, a reward of -1 is given; in other cases, $reward_t = \frac{1}{r_{t+1}} - \frac{1}{r_t}$ where $reward_t$ is the reward of the agent's action and r_t is the distance between the vehicle and the target at t^{th} step. Since Eqs. 1 and 4 can not compute the positive number and negative number at the same time, the reward should be normalized

TABLE I
SUMMARY OF THE PARAMETER SETTING IN MEMETIC AGENT

TD-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 ($\gamma^{c1}, \gamma^{c2}, \gamma^{c3}$)	1.0, 1.0, 1.0
Baseline vigilance parameters ($\rho^{c1}, \rho^{c2}, \rho^{c3}$)	0.2, 0.2, 0.5
Temporal Difference Learning Parameters	
TD learning rate α	0.5
Discount factor γ	0.1
Initial Q-value	0.5
ϵ -greedy Action Policy Parameters	
Initial ϵ value	0.5
ϵ decay rate	0.0005

prior to the calculation of Eqs. 1 and 4. The size of minefield is set to be $16 \equiv 16$.

2) *Parameter Settings*: For a fair comparison, the parameter settings of the memetic agent TD-FALCON are kept the same in all the following experiments. As suggested in [15], the parameter settings of TD-FALCON are summarized in Table I.

In MeMAS, the meme variation process includes two new parameters (τ and η) and both of them are set to 0.1 for generating moderate innovative characteristics of interactions between agents.

In memetic agents with structured memes, two parameters are introduced, namely $\rho_{iniclasp}^{ck}$ and $NewSplitClass$ of the *class splitting* process. In Eq. 13, $\rho_{iniclasp}^{ck} / [1 - \rho^{ck}, 1)$ is the parameter for adjusting ρ_{clasp}^{ck} . Meanwhile, a relationship between $\rho_{iniclasp}^{ck}$ and the max depth of *StructuredMeme* in an agent (denoted as $maxDep$) can be infer from Eq. 13.

$$maxDep \geq \max_{k \in \{1,2,3\}} \log_{\rho_{iniclasp}^{ck}} (1 - \rho^{ck}) \quad (21)$$

Given the values of ρ^{ck} , a large $\rho_{iniclasp}^{ck}$ leads to a large $maxDep$. To get a proper *StructuredMeme* which can be built and retrieved in a short time, $maxDep$ is set to 3, thus the values of $\rho_{iniclasp}^{ck}$ are set to (0.9, 0.9, 0.8).

The parameter $NewSplitClass$ controls the size of class in *StructuredMeme* and its value selection impacts the MeMASS performance. We take a further step to investigate such impact in the MNT domain and show the performance over 300 runs for both the single agent and multiagent cases in Fig. 7. As indicated in Fig. 7(a), a larger $NewSplitClass$ value, which indicates many classes in each layer, generally requires more time on the meme search. However, the run time does not significantly increase since a proper meme can be easily located. In contrast, a small value of $NewSplitClass$ leads to a large amount of time on the meme search. For example, it demands around 90s and 225s respectively for both the cases when $NewSplitClass$ is set to 1 in MeMASS (not shown in the figure due to the large scale). On the other hand, as shown in Fig. 7(b), the success rate, which shows the number of successful navigations over the total number of simulations, grows with the increasing of the $NewSplitClass$ value. This is because memes are clearly differentiated and a similar meme is precisely identified in the search. For a

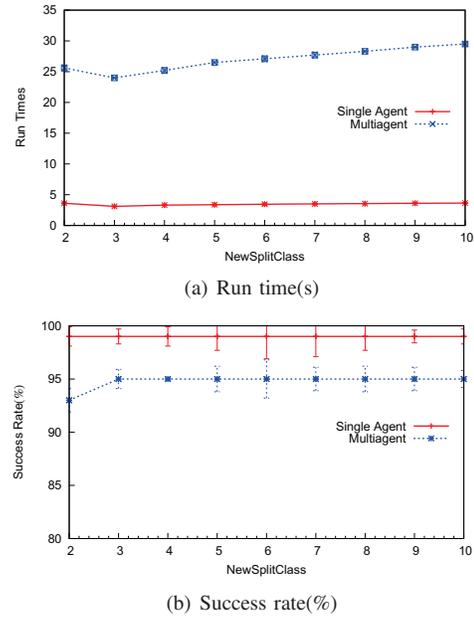


Fig. 7. Impact of the parameter $NewSplitClass$ on the MeMASS performance in the MNT domain.

performance tradeoff, we let $NewSplitClass$ be equal to 3 in MNT.

We further observe that the increasing value of $NewSplitClass$ does not substantially improve the success rate if a sufficient number of classes are developed to differentiate the memes in the search. In general, the $NewSplitClass$ value is not necessary to be rather large. For example, it is set to be 3 in the MNT domain and the success rate does not have much change even if we increase the $NewSplitClass$ value. This also occurs when the impact of $NewSplitClass$ is studied on the run times. Accordingly we have the uniform setting of $NewSplitClass$ (equal to 3) in all the three domains and obtain expected performance.

3) *Utility of MeMAS*: MeMAS emerges as an important framework for multiagent learning particularly in the line of multiagent demonstration learning research. Before we examine the improved MeMAS, we first show the performance of the plain MeMAS compared to the state-of-art multiagent demonstrate learning method - advice exchange model (AE) [27]. In the AE model, agents seek advice only from the elitist for the next action to take. However, the blind reliance, of the elitist, could hinder the learning process. Agents in MeMAS can achieve better learning performance through the meme selection and meme variation operators in the system. Fig. 8 confirms that MeMAS outperforms AE in the tests. Hence, we follow the MeMAS framework and aim to improve its performance in solving multiagent learning problems.

4) *MeMAS with Structured Memes*: To evaluate the effectiveness of structured memes, we compare the MeMAS consisting of memetic agents with structured memes (MeMASS) and the conventional MeMAS (CMeMAS) [19] on the average 100-trial intervals success rate. We first investigate the single agent case by letting one vehicle execute the task for a total

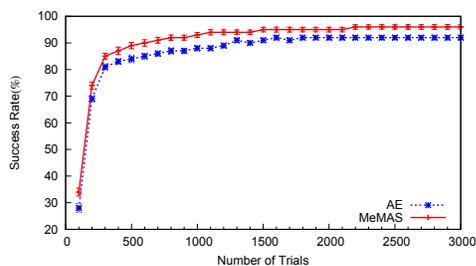
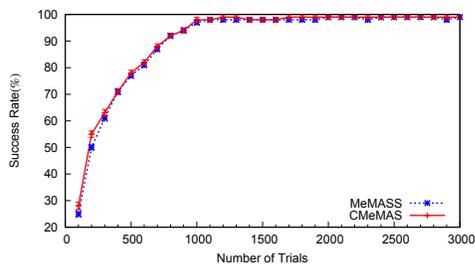
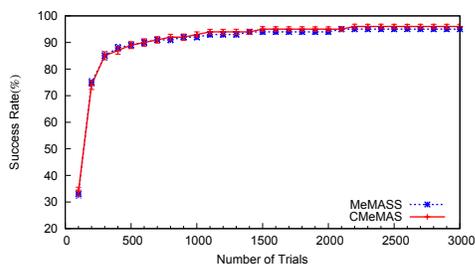


Fig. 8. MeMAS outperforms the state-of-art multiagent demonstrate learning method - AE - in the MNT domain .



(a) Single Agent Case



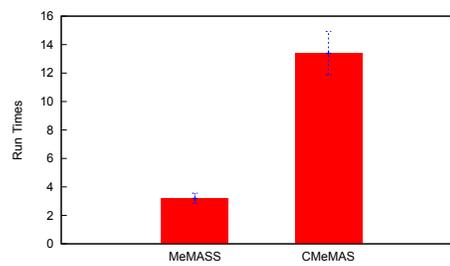
(b) Multiagent Case

Fig. 9. Average Success Rate of MeMASS and CMeMAS in MNT.

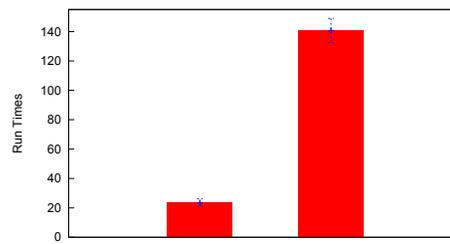
of 3000 trials, and then we consider the multiagent case by using six vehicles to do the same experiments. All simulations are repeated 300 times.

Fig. 9 shows the average success rates (as well as the variances, which are however quite small in most the cases and can't be easily spotted in the figures) of MeMASS and CMeMAS on completing the missions in MNT. In both cases, their performance gap is extremely small in every interval. Hence MeMASS maintains the high effectiveness of CMeMAS.

In addition, we use the average run time of a simulation to evaluate the efficiency of MeMASS and CMeMAS. As shown in Fig. 10, MeMASS is approximately 4 times faster than CMeMAS in the single agent case and 7 times faster than CMeMAS in the multiagent case. Note that the multiagent case is more complex than the single agent case. Hence MeMASS is more efficient than CMeMAS, especially on the complex problem. Overall, the MeMAS with structured memes significantly improves the efficiency of MeMAS while maintaining the high effectiveness. Subsequently we will use MeMASS in the rest of experiments.



(a) Single Agent Case



(b) Multiagent Case

Fig. 10. Average Run Time (s) of MeMASS and CMeMAS in MNT. All experiments are run on Intel i7-3770 CPU (4 cores), 3.4 GHz, and 8G memory. The vertical lines with bars denote the variance values.

B. Adapted Minefield Navigation Domain

To investigate the performance of the improved selection strategy in MeMAS, we adapt the basic MNT domain by adding more types of tank. We first briefly describe the new problem domain and then present the performance of the new selection strategy in MeMAS. Finally, we analyze the scalability of the improved MeMAS in more complicated scenarios.

1) *Domain Description*: To investigate the performance of the improved selection strategy in MeMAS, we adapt the basic MNT domain by adding more types of tank. The basic rules of the adapted minefield navigation task are the same as those of the previous domain. In addition, we include two types of vehicles as well as mines in the field. One type of vehicle wears a thin armor, denoted by *Veh1*, and can be easily eliminated by any mine type; while the other possesses a thick armor, as denoted by *Veh2*, and can only be destroyed by the highly explosive mines (in red in Figs. 11). A screenshot of this domain is shown in Fig. 11.

In this domain, we have a total of 10 tanks (divided equally for two types) and 40 mines (also divided equally for two types) in a 32×32 field. The vehicle is allowed to move at most 60 steps in each trial (run out of time). To study the performance of the new selection criterion (denoted as MeMAS-E) and the conventional elitism selection criterion (denoted as MeMAS-C) in the MeMAS framework, we let the vehicle agents execute the task every 100 trials of training and continuously perform this for a total of 3000 trials. The simulations are repeated for 30 times.

2) *Performance of the New Selection Mechanism*: Fig. 12 depicts the average success rates of both types of vehicles on completing the missions in AMNT. The *Veh2* agents perform competitively in both MeMAS-E and MeMAS-C,

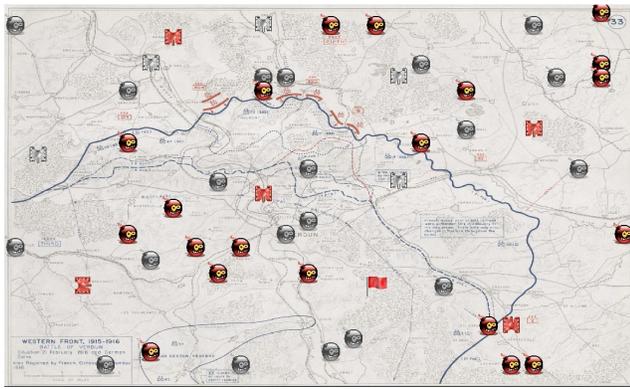


Fig. 11. A scenario of the adapted minefield navigation task. *Veh1* is represented by gray rectangle while *Veh2* is represented by red rectangle.

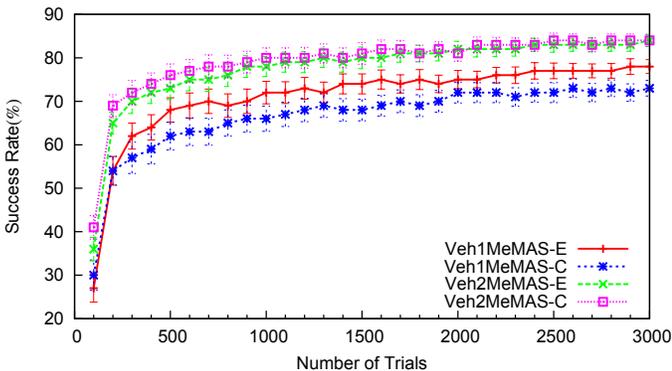


Fig. 12. Average success rate performance of tank agents in both MeMAS-E and MeMAS-C on completing the missions in AMNT.

while *Veh1* agents in the MeMAS-E outperform their counterparts in MeMAS-C. In particular, in MeMAS-E, vehicles with a thin armor (*Veh1*) have a higher success rate than those in MeMAS-C. The results show the benefits of the like-attracts-like selection criterion over an elitism based scheme. In MeMAS-C, we observe that *Veh1* prefers to learn from *Veh2* because *Veh2* having a thick armor often succeeds in achieving the goals and becomes the elitist agent. However, *Veh1* is often destroyed by the explosive mines when it follows *Veh2* and moves across the minefield. This leads to the downfall of many *Veh1*s so that the *Veh1* fails to repeat *Veh2*'s successful experience. On the other hand, by selecting a similar type of tanks as the teacher agent within the environment, as in MeMAS-E, *Veh1* is able to truly imitate the appropriate skill of similar counterparts in achieving the robust performances observed by both *Veh1* and *Veh2*.

To further compare MeMAS-E with MeMAS-C, we also compute the diversity of memetic agents' behavior using information entropy based on the type of tanks that evolves over time. For a fair comparison, we generate 10,000 random stations into all agents in MeMAS-E or MeMAS-C, and then derive the average information entropy of the agents' behavior in the given station. The results from 30 simulations are summarized in Fig. 13. The MeMAS-E always exhibits a larger diversity of vehicles' behavior than the MeMAS-C. Since the vehicle agents in MeMAS-E learn from both

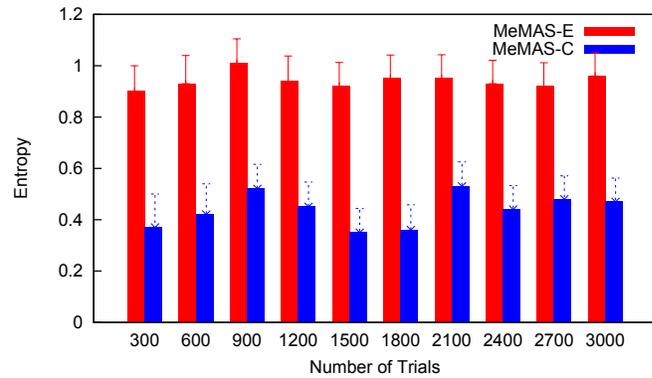


Fig. 13. MeMAS-E maintains a larger diversity of vehicles' behavior than the MeMAS-C in AMNT.

a similar type of vehicle and the elitist one, they would differ not only in the armor property, but also the emerged social behavior. In contrast, all agents in MeMAS-C learn from the elitist agents, thereby converging to similar behavior. This indicates that the different types of agents co-exist in MeMAS-E while the agents tend to evolve into a single form in MeMAS-C. As an indicator of effective co-evolution of memetic agents on solving the task, MeMAS should maintain a high diversity, which stimulates the emergence of human-like social behavior.

We take a further step to investigate the human-like behavior of the MeMASs by inviting 20 participants to observe the process. The vehicle team is trained for 1,000 trials and completes 10 tasks (including both failure and success cases) in AMNT. The observations include how the tanks move through dangerous areas filled with mines, how they avoid the collision with other vehicles, how they individually/collaboratively reach the target. We first ask the participants to rate on both the diversity and intelligence of the vehicles' actions, and then to rate the human-like performance of the MeMASs. We report the average scores (with the variance) of both MeMAS-E and MeMAS-C in Table II.

TABLE II
AVERAGE SCORES OF THE MEMASS' PERFORMANCE. 5 IS THE HIGHEST AND 1 IS THE LOWEST.

Criteria	Diversity	Intelligence	Human-like
MeMAS-E	3.80(0.76)	4.20(0.46)	3.90(0.39)
MeMAS-C	2.85(0.63)	3.30(0.51)	3.05(0.85)

The results show that MeMAS-E outperforms MeMAS-C on all evaluation criteria. It is a bit surprising that MeMAS-E exhibits more intelligence on solving the tasks. In MeMAS-C, *Veh1* performs some incompatible actions that are learnt from *Veh2* without being aware of the difference on their personal types. The subject study also confirms that the new learning mechanism combining the elitist and like-attracts-like principles improves the human-like behavior of MeMAS.

3) *Scalability Analysis*: We develop more complicated scenarios of the adapted MNT to test the scalability of the improved MeMAS. We add another type of tank that can be eliminated within some distance from mines and the distance

specification depends on the type of mines. The third type of tank agent is not clearly differentiated from the other two types defined previously in the domain. It can learn from both types of the other two types of tanks. The new selection strategy adapts the learning by adjusting the parameters (K_1 and K_2) online. This decides which type of tanks shall be learned by the third type of tanks. In Fig. 14, the other two types of tanks follow similar performance trend as shown in Fig. 12. As the third type of tanks learn from the most proper type of tanks, they achieve larger success rates than others in Fig. 14. In other words, learning from similar types often improves the tanks' success rates. The tanks with more general attributes may easily adapt the learning strategies thereby achieving better performance. As expected in Fig. 15, the new MeMAS framework still maintains a larger diversity of agent types, which endorses the good performance of the entire system.

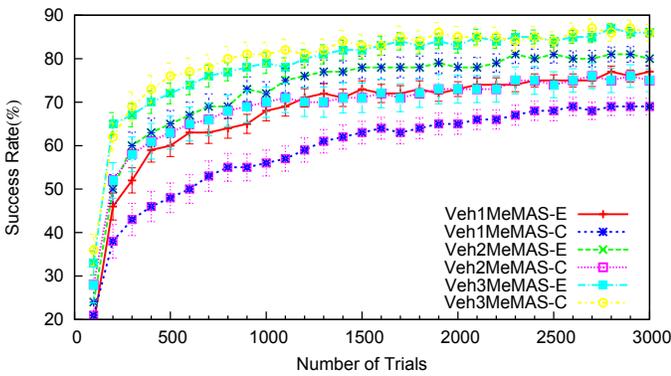


Fig. 14. Average success rates of various types of tank agents in both MeMAS-E and MeMAS-C.

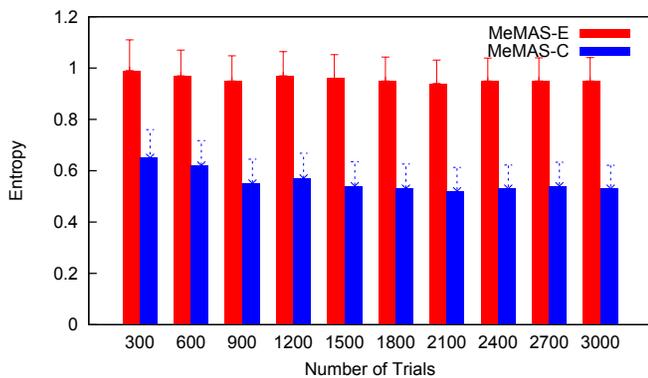


Fig. 15. MeMAS-E still achieves good diversity when more types of tank agents are involved in the tasks.

We can't arbitrarily increase the number of agent types since each type shall hold a meaningful property in the domain of study. Instead we increase the number of agents for each type to test the scalability of the improved MeMAS. We don't repeat the comparison of success rates and diversity since similar trends are observed in the study. We compare the run times for both types of MeMAS in Fig. 16. MeMAS with structured memes and new selections improves the efficiency of the original MeMAS. The improvement becomes more outstanding when the number of agents increases.

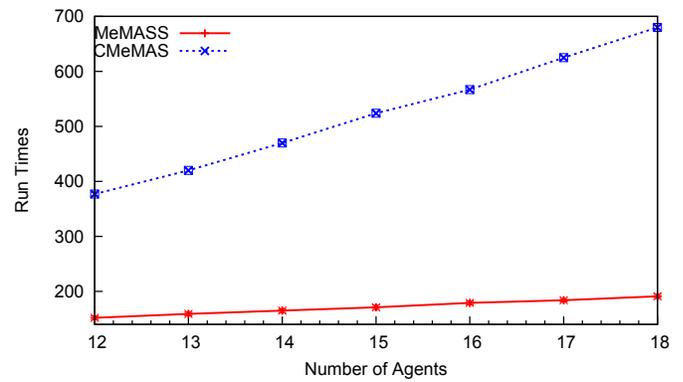


Fig. 16. Performance of the improved MeMAS in more complicated scenarios.

C. 3D Interactive Homeland Defense Game

To actively engage human-players in assessing the performance of the new principle via MeMAS-E, we designed a 3D interactive game based on the *Unity*¹ engine and integrated the MeMAS framework with the game engine. The game is an abstraction of the popular *Tower Defense Games*. A game screenshot is shown in Fig. 17.

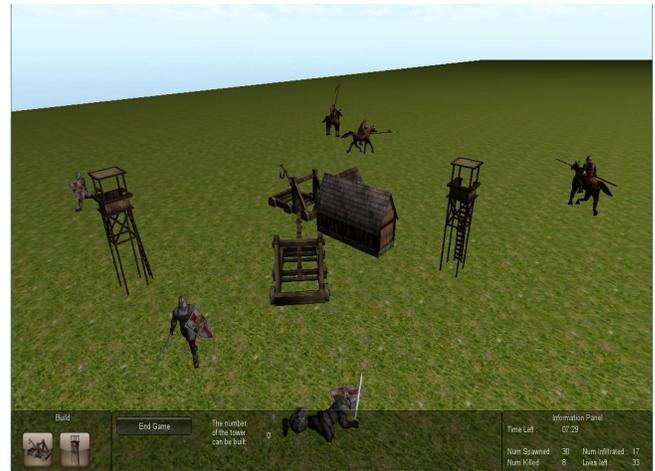


Fig. 17. Human-players intend to defend the house (homeland) in the middle of the scene. They are building the forts while combating the offenders.

In the homeland defense game, the task for a human-player is to prevent the house (homeland) from being destroyed by the offenders in a limited time. The defender (human-player) can construct two types of forts, namely an arrow tower and a stone tower, that shoot its enemies within a certain range. Meanwhile, the offenders have two types of arms, light cavalry and heavy infantry. Light cavalry has a high agility and a thin armor, while heavy infantry has a low agility and a thick armor. Hence the light cavalry can be eliminated immediately by an arrow and killed by a stone with a probability of 20%. In contrast, the heavy infantry can be eliminated by a stone and killed by an arrow with a probability of 1%. The house (homeland) is destroyed iff N offenders have entered it. The game is terminated when the house (homeland) is destroyed

¹<http://unity3d.com/>

(game over) or the house is safe in a limited time T minutes (mission success). For the following test, we set $N = 100$, $T = 5$, the size of map is $7 \equiv 7$.

We first train NPCs 300 trials in MeMAS-E and MeMAS-C respectively, and then enroll 20 persons (from novice to experienced game players) to play with the trained NPCs in the game. After that, the players were asked to rate the behavior of NPCs based on two questions: 1) How tricky and interesting are the routes selected by the NPCs? 2) how intelligent are the actions taken by the NPCs to avoid the forts? Finally, they also score the NPCs' human-like behavior. The average scores (with the variance) and the success rate of the NPCs when they compete with the human-players are summarized in Table III.

TABLE III
AVERAGE SCORES OF THE MEMASS' PERFORMANCE IN THE HOMELAND DEFENSE GAME. 5 IS THE HIGHEST AND 1 IS THE LOWEST.

Criteria	Interesting	Intelligence	Human-like	Success Rate
MeMAS-E	4.20(0.66)	3.90(0.79)	3.95(0.55)	65%
MeMAS-C	3.10(0.69)	3.05(0.75)	3.25(0.79)	40%

The results show that MeMAS-E performs better than MeMAS-C on both the human-like behavior of the NPCs and the success rate. Some interesting comments from the human-player are: 1) The behavior of MeMAS-E is diverse and unpredictable while the behavior of MeMAS-C is uniform and predictable. 2) In MeMAS-C, the light cavalry often takes some silly actions that imitate the heavy infantry by attempting to navigate through the dangerous areas filled with the arrow towers. The observations reveal some difference between the conventional elitism and the new learning mechanisms. In contrast to the elitism mechanism, the new learning mechanism gives more possible models for the agents to perform and learn. Hence the behavior of agents is more various and unpredictable. In a diverse environment, the agent should consider not only the performance but also the attributes for selecting the teacher agents. It illustrates that the like-attracts-like principle plays an important role in the development of human-like multiagent systems.

D. Discussion

We have shown the performance of the improved MeMAS in different scenarios of problem solving. It indicates the importance of using agent types to guide the learning for agents while the structured memes significantly improves the efficiency particularly in complicated domains. Since we currently consider cooperative agent systems where the agent types are known, we are able to adjust the parameter to adapt the selection (as shown in the analysis in Section V-B3). This would become difficult when the type information is unknown in competitive agents settings. Hence it would be very interesting to investigate the new selection strategy in the competitive agent systems.

VI. CONCLUSION

We introduce the memetic multiagent system that uses the TD-FALCON model for commanding observations and

actions in an uncertain setting. As a desirable property of multiagent systems, human-like behavior not only improves solutions to complex problems, but also allows the multiagent techniques to be seamlessly engaged into personal business. We proceed to improve the human-like social behavior of MeMAS. Particularly we focus on the improvement of the *meme internal evolution* and *meme external evolution* process.

We propose memetic agents with structured meme so that the agents can improve the meme search in *meme internal evolution*. The memetic agents adopt a hierarchical and adaptive classification method to maintain memes in a tree structure, which facilitates the decision making within a short time. Experimental results on MNT show that the memetic agents with structured memes improve the efficiency of MeMAS while keeping the high effectiveness on executing tasks.

We further present a learning mechanism to improve human-like social behavior of memetic agents. The new learning mechanism is a trade-off between the imitate-the-elite and like-attracts-like principles. Meanwhile, the influence of each principle is weighted in a dynamic way. Hence the new learning mechanism is self-adaptive in the changing environment. The performance comparison shows the emergence of human-like social behavior of memetic agents in the study.

Although research has been conducted on human-like behavior for a long period, few formal methods have been found on quantifying the human-like behavior of intelligent agents. Most of existing behavioral evaluation still relies on the subject study, which is also the line we follow in this paper. Quantitative formulations on human-like social behavior should be developed in immediate research. The behavior of MeMAS under the like-attracts-like based learning principles implies important factors, like diversity and intelligence of actions in the formulation. Our future work will include the investigation and examination for the formulation in various types of problem domains.

REFERENCES

- [1] Y. S. Ong and A. J. Keane, "Meta-lamarckian learning in memetic algorithm," *IEEE Transactions On Evolutionary Computation*, vol. 8, no. 2, pp. 99–110, 2004.
- [2] O. Kramer, "Iterated local search with powell's method: a memetic algorithm for continuous global optimization," *Memetic Computing*, vol. 2, no. 1, pp. 69 – 83, 2010.
- [3] K. Tang, Y. Mei, and X. Yao, "Memetic algorithm with extended neighborhood search for capacitated arc routing problems," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1151–1166, 2009.
- [4] L. Feng, Y. S. Ong, Q. H. Nguyen, and A. H. Tan, "Towards probabilistic memetic algorithm: An initial study on capacitated arc routing problem," *IEEE World Congress on Computational Intelligence, Congress on Evolutionary Computation 2010*, pp. 18–23, 2010.
- [5] A. B. Ullah, R. Sarker, D. Cornforth, and C. Lokan, "Ama: a new approach for solving constrained real-valued optimization problems," *Soft Computing*, vol. 13, no. 8, pp. 741–762, 2009.
- [6] L. Jiao, M. Gong, S. Wang, B. Hou, Z. Zheng, and Q. Wu, "Natural and remote sensing image segmentation using memetic computing," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 78 – 91, 2010.
- [7] H. Situngkir, "On selfish memes: culture as complex adaptive system," *Journal of Social Complexity*, vol. 2, no. 1, pp. 20–32, 2004.
- [8] F. Heylighen and K. Chielens, "Cultural evolution and memetics," *Encyclopedia of Complexity and System Science, B. Meyers, ed.: Springer*, 2008.
- [9] Q. H. Nguyen, Y. S. Ong, and M. H. Lim, "Non-genetic transmission of memes by diffusion," *Proceedings of the 10th annual conference on Genetic and evolutionary computation (GECCO '08)*, no. 8, pp. 1017–1024, 2008.

- [10] R. Meuth, M. H. Lim, Y. S. Ong, and D. Wunsch, "A proposition on memes and meta-memes in computing for higher-order learning," *Memetic Computing*, vol. 1, pp. 85–100, 2009.
- [11] G. Acampora, V. Loia, and M. Gaeta, "Exploring e-learning knowledge through ontological memetic agent," *Computational Intelligence Magazine*, vol. 5, no. 2, pp. 66 – 77, 2010.
- [12] X. S. Chen, Y. S. Ong, M. H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Transactions on Evolutionary Computation*, *In Press*, 2011.
- [13] Y.-S. Ong, M.-H. Lim, and X. Chen, "Memetic computation - past, present & future [research frontier]," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 24–31, 2010.
- [14] X. Chen, Y.-S. Ong, M.-H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *Trans. Evol. Comp.*, vol. 15, no. 5, pp. 591–607, Oct. 2011. [Online]. Available: <http://dx.doi.org/10.1109/TEVC.2011.2132725>
- [15] A. H. Tan, N. Lu, and D. Xiao, "Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback," *IEEE Transactions on Neural Networks*, vol. 9, no. 2, pp. 230–244, 2008.
- [16] A. Belatreche, *Biologically Inspired Neural Networks: Models, Learning, and Applications*. VDM Verlag, 2010.
- [17] T. B. Gutkin, G. C. Gridley, and J. M. Wendt, "The effect of initial attraction and attitude similarity-dissimilarity on interpersonal attraction," *Cornell Journal of Social Relations*, vol. 11, no. 2, pp. 153–160, 1976.
- [18] A. Sloman, "Architectural requirements for human-like agents both natural and artificial. (what sorts of machines can love?)," Birmingham Univ. (United Kingdom). School of Computer Science, Report BU-SCS-CSR–98-21, 1998.
- [19] L. Feng, Y.-S. Ong, A.-H. Tan, and X. Chen, "Towards human-like social multi-agents with memetic automaton," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2011, pp. 1092–1099.
- [20] 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 multitagent systems," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 1635–1642.
- [21] A. Lynch, "Thought contagion as abstract evolution," *Journal of Ideas*, vol. 2, pp. 3–10, 1991.
- [22] S. Blackmore, "The evolution of meme machines," *International Congress on Ontopsychology and Memetics Milan*, 2002.
- [23] R. Dawkins, "The selfish gene," *Oxford: Oxford University Press*, 1976.
- [24] 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. Springer, New York, NY, USA, 2004.
- [25] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.
- [26] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Addison-Wesley, 2005.
- [27] E. Oliveira and L. Nunes, "Learning by exchanging advice," *Studies in Fuzziness and Soft Computing*, pp. 279–313, 2005.