Asian Journal of Information Technology 11 (6): 300-311, 2012
ISSN: 1682-3915

Knowledge Distribution in Multi Agent Systems

[1]William J. Teahan, [2]Nidal A. Al-Dmour and [1]Peter G. Tuff
[1]School of Informatics, University of Wales, Bangor, United Kingdom
[2]Department of Computer Engineering, Mutah University, Karak, Jordan

Abstract: Researchers explore the problem of how to distribute knowledge within a multi-agent system and discuss how question and answering, knowledge sharing and evolution of knowledge are key parts to the solution of this problem. Researchers also review genetic programming search algorithms and re-consider them from a knowledge sharing and knowledge evolution perspective. The hope is that this will highlight situations where there might exist possible ways for improving the search algorithms.

Key words: Knowledge, distribution, Multi Agent system, genetic programming, evolution, Jordan

INTRODUCTION

Thought and kowledge: What is thought? Baum has examined in depth this question in his recent book of the same title (Baum, 2004). To Baum, thought can be thought of as a program made up of components that are semantically meaningful modules. Researcher also argues that the complexity of the mind is the outcome of evolution, adaptation and learning and that underlying mind is a complex but compact program that corresponds to the underlying structure of the world.

Historically, one of the primary focus areas of Artificial Intelligence (A.I.) has been on building intelligent systems i.e., systems that think. One outcome of this process of building intelligent systems is that we might develop a better understanding of how we think that is how a mere handful of stuff can perceive, understand, predict and manipulate a world far larger and more complicated than itself (Russell and Norvig, 2003). Russell and Norvig characterize definitions of A.I. into four categories: systems that think like humans (what Russell and Norvig call the cognitive modelling approach), systems that act like humans (the Turing test approach), systems that think rationally (the laws of thought approach) and systems that act rationally (the rational agent approach). In their standard text book on A.I., they adopt the latter concept of rational agents as being central to their approach. The emphasis is on developing agent systems that can reasonably be called intelligent.

However, there is much confusion over what people mean by an agent. From the A.I. perspective, a key idea is that an agent is embodied (i.e., situated) in an environment. For example, a game-based agent is situated in a virtual game environment whereas robotic agents are situated in a real (or possibly simulated) environment. Russell and Norvig list four key attributes in this sense: utonomy (acting on one's own behalf without intervention); social-ability (able to communicate in some manner with other agents); reactivity (reacting to stimuli) and proactivity (being proactive). The weak definition of agents includes these four attributes but adds temporal continuity and goal-orientedness as two further key attributes. The strong definition of agents includes all six previous attributes, plus five more: mobility, benevolence (i.e., not being destructive), rationality, adaptivity and collaborative ability.

Researchers add yet more attributes in the strongest notion of an agent: knowledgeability, intelligence, self-awareness, consciousness and thoughtfulness (i.e., an agent that thinks as we do) (Teahan, 2003). We maintain that for an agent to think, it must first have knowledge of the environment it finds itself in as well as knowledge of how to act within it to maintain its competitive edge (in terms of fitness to survive compared to other agents). An agent must also be intelligent i.e., be able to understand the meaning of its knowledge, be able to make further inferences to add to its knowledge and to act in an intelligent manner in order to react to whatever is happening in its environment or whatever is likely to happen (again in order to maintain or improve its fitness). Self-awareness, consciousness and thoughtfulness correspond to the human traits researchers are all familiar with but there is a lack of real understanding of how they happen or of how researchers might go about developing artificial systems that have these properties.

The definitions of these properties are quite arbitrary and are the subject of much debate and research in

Corresponding Author: William J. Teahan, School of Informatics, University of Wales, Bangor, United Kingdom

philosophy, psychology, biology, neurology and cognitive science. Interestingly enough, self-awareness may not be such a difficult property for an agent to achieve depending on how you define it. For example, one test for self-awareness devised by Gordon Gallup in the 1960s (Harrub and Thompson, 2004) simply involves performing an experiment with an animal inside a room with a mirror. Once the animal becomes familiar with its environment, the animal is then anaesthetized and a mark is placed on its forehead which it can only see in a mirror. The animal is then placed back into the room and is deemed to be self-aware if it notices the mark on its forehead. Animals that have been deemed to be self-aware using this definition are orangutans and chimpanzees. An interesting possibility exists for performing a similar experiment with a computer-based agent: place the agent in a virtual environment that has a virtual mirror and program it to reproduce the same behaviour. The question is does this really mean that the computer-based agent is self-aware?

Part of the purpose of this study is to document preliminary explorations into how to build knowledgeable agents. The belief is that of the strongest agent requirements we defined above, knowledgeability is achievable with current computer systems and technologies. However, the other traits are more problematic. Developing true intelligence, for example, even in the restricted setting of a Turing test (Turing, 1950), presents many difficulties despite many sites on the Web claiming otherwise. We will leave that exercise for another time once we have tackled the problem of knowledgeability (as researchers believe that knowledgeability holds a key to their solution).

The study reviews the knowledgeable agents framework we have developed for representing knowledge. In this study researchers explore the problem of how to distribute knowledge within a multi-agent system and discuss how question and answering, knowledge sharing and evolution of knowledge are key parts to the solution of this problem. In this study we review knowledge-based approaches to genetic programming and re-consider them from a knowledge sharing and knowledge evolution perspective. Researchers also propose a novel variation to the standard genetic programming algorithm based on an approach that shares and evolves knowledge amongst family units.

Knowledgeable agents: The question what is knowledge? is an interesting one. Although, researchers all are very familiar with the concept, it is very difficult to define precisely. Knowledge engineering is an important area of research in artificial intelligence for example but the literature often avoids defining what knowledge is and often assumes or adopts a traditional approach based on some form of inferencing and logic.

Knowledge can be defined as all information needed by a human being or machine, to complete a task considered as being complex (Ferber, 1999). That is knowledge concerning a specific topic is organised and retained by a human or machine that can be utilised when a task needs to be completed. According to Ferber (1999) such knowledge can be divided into two categories: knowing something and knowing how to do something. Knowing something concerns the knowledge and understanding of objects and phenomena encountered while the knowing how to do something relates to the analysis of the relationship between different phenomena. Knowledge relating to how to do something allows a human or machine to select an appropriate action given the current state of the world and to anticipate the effect of the action on the state of the world.

The approach (Clifton and Teahan, 2005; Teahan, 2003) employs a similar task-based definition of knowledge when classifying agents as knowledgeable. In the definition, knowledge must be associated with some agent; knowledge cannot exist on its own. If we define information as being data (numbers or text) that is potentially useful in answering a question, then in the definition, an agent has knowledge if it knows how to use that information to help answer a question in a certain context.

Researchers see questioning and answering as being key to the process of obtaining knowledge and of being knowledgeable. If we consider selecting an appropriate action as the question then the agent utilises its knowledge to provide an answer in the form of an appropriate action. Therefore, we can describe knowledge as information that enables the agent to select an action that is appropriate for the current state of the world and the adopted goal of the agent.

An agent is equipped with knowledge of an environment and knowledge of how its actions affect the environment. This inherent knowledge is recognised as the agent's knowledge base. Arguably, the agent's knowledge is not limited to simply determining what action is appropriate. The agent's knowledge also incorporates information gathered by perceiving its environment and interactions with other agents.

Researchers describe a framework for designing and implementing knowledgeable agents and the knowledge grid (Cannataro and Talia, 2003) (Teahan, 2003) defines the knoeledge grid as the computational grid organized into three layers: a data grid at the bottom, an information grid in the middle and the knowledge grid on top) based on the concept of Knowing-Aboutness. The framework comprises three types of knowledge relations: Knows, KnowsAbout and KnowledgeableAbout. These are used to define what an agent knows what it knows about and whether an agent has been judged to be knowledgeable by other agents. Knows is used to describe what answers an agent knows to a question in a particular context.

KnowsAbout is used to describe the topics and contexts that the agent knows about (where knowing about a topic implies that you know something about the topic but it does not imply that you know everything about the topic). Knowledgeable is used to describe whether an agent has been deemed by an external testing agent to be knowledgeable about a particular topic and context.

The key idea behind these concepts is as follows. Researchers as humans have an intuitive understanding of what it means for another human to be knowledgeable we can judge from their answers to the questions and from the own knowledge whether that person seems to be knowledgeable or not about a certain topic. By having humans be the judges, this allows us to sidestep the deep philosophical issues of what knowledge is or what it means to be knowledgeable in a manner similar to the way Turing (1950) sidestepped the issue of how to define intelligence when he devised the Turing test B in essence he said that we know it when we observe it, so why do we need to define it?

The knowledge grid architecture is based on using knowledgeable agents as a middle layer between the user and the information resources. The users do not interface directly with the information resources. Instead, they must go through a knowledgeable agent who effectively acts as a knowledge broker in determining which of the information resources are likely to contain an answer to the user's questions. Notice that knowledgeable agents may need to go though other knowledgeable agents in the hunt to find the most relevant answer to the user's questions.

Knowledgeable agents are knowledgeable about a topic or topics. Knowledgeable agents are similar to expert systems in that both employ knowledge bases and are capable of using such domain knowledge to solve problems and answer questions.

The main difference between a system employing knowledgeable agents and an expert system is that an expert system is inherently disembodied (Wooldridge, 2002; Maher and Gu, 2002). That is an expert system does not interact with the environment, unlike an agent that may have sensors enabling it to perceive and gather information in an autonomous fashion; information, for expert systems is supplied by the user. We have implemented a Question and Answering system called QITEKAT (Question Inferencing Tools Employing Knowledgeable Agent Technologies) based on this architecture (Clifton and Teahan, 2005).

The system is very competitive compared to other systems based on annual evaluations in Question Answering track at the Text Retrieval Conference (TREC). The QITEKAT Question-Answering system knowledge base is generated by transforming corpora into a series of question and answers relations or sentences. The knowledge base is subsequently utilised by the agent to answer questions asked by the user.

In addition, individual agents can communicate with other knowledgeable agents. Therefore, a question asked of an agent can be sent to another agent with an alternative knowledge base, allowing a more knowledgeable agent to answer the question. If an answer is found then the agents that have participated in the communication acquire this new knowledge thus, expanding their own knowledge base. Consequently, a user only has to ask a single knowledgeable agent to query the whole accumulated of knowledge of every agent in the QITEKAT system.

The CYCic Friends Network (Mayfield *et al*., 1995) uses an alternative approach based on traditional logic and rules of inference to build knowledgeable agents. Communication between agents is usually performed using an Agent Communication Language (ACL) such as KQML (Knowledge Query and Manipulation Language). KQML defines a common format for messages being passed between agents. In the CYCic Friends Network, KQML is used to allow three agents, representing different domain expertise, to communicate with each other.

Therefore, when a question is asked, the system can reason and combine the knowledge of the three agents to provide an answer that each of the single agents by itself could not provide themselves.

In this, researchers look at some broader issues related to the knowledgeable agent theme specifically, the problem of knowledge evolution within a multi-agent system (where the agents are both human and computer-based).

The knowledge web, meta-questioning and knowledge evolution: One of the growing problems today in research is the increasing specialisation of knowledge. Hillis (2004) states:

> There is a growing mountain of research. But there is increased evidence that we are being bogged down today as specialization extends. The investigator is staggered by the findings and conclusions of thousands of other workers conclusions that he cannot find time to grasp, much less to remember

One result of this specialisation of knowledge is the repeated use of the same terms but with different meanings in various contexts (for example, the use of the term agent is especially problematic). This often leads to increased confusion, requiring a substantial amount of effort to avoid or to explain around. Time must be spent describing the background for scientific papers and presentations and discussions often involve a combined effort between speakers to fill in each other's lack of knowledge or misunderstanding.

Hillis proposed the notion of a knowledge web to overcome this confusion in terminology. In his definition, a knowledge web is a database containing all the world's knowledge.

It is organized (in his words) according to concepts and ways of understanding them and contains specific knowledge about how the concepts relate who believes them and why and what they are useful for. He chose the term knowledge web to distinguish it from the World Wide Web.

Researchers describe the application of agent-oriented systems to Teaching, Learning and Research (T and R) (Teahan, 2005). The approach requires the establishment of an environment called the knowledge web that facilitates T and R. We see the knowledge web as a mixed initiative (Hearst, 1999) network of collaborative agents who provide mediated access to a federation of knowledge bases.

The network consists of many tutors, students and we, each actively assisting each other to enhance the quality of the knowledge web. Importantly, the key agents in this web are human-based rather than computer-based that is we do not seek to replace human-based tutors with computer-based ones (and in any case, under the definition, it is the humans judging what knowledge is). Importantly, each agent in the knowledge web acts on each other's behalf to enhance the knowledgeability of the web.

Researchers believe that an important part of the process of decision making and of becoming and being knowledgeable is knowing how to ask the right questions. This is an important factor if we are to achieve the goal of a knowledge web comprising knowledgeable agents.

This can be partially achieved using a meta-questioning process (Teahan, 2005) that consists of consciously asking structured questions about the questions and answers under consideration. The purpose of the meta-questioning process is to provide structure to the knowledge elicitation process for the agent, to make conscious note of the inconsistencies that they have with their own knowledge and to seek the clarification needed to fill in any gaps in understanding.

Referring to biological systems humans for an interesting analogy, psychological studies have shown that men, women and children ask questions differently. Women, unlike men, tend to answer with a question or ask more additional questions (Baird, 1976). A child will also continually ask further questions such as Why? There seems an inhibition among many adults and older children to continue this meta-questioning process, perhaps being afraid to lose face by displaying a lack of knowledge of what the speaker is talking about.

Some interesting questions are what is research? and how can we go about doing it? The process of re-search can be presented as a meta-questioning process: an initial posing of questions and answers about the research topic followed by a continued process of searching for further questions and answers. Meta-questions can be asked about the fundamental questions being posed by the researchers that are at the core of the research and these meta-questions can form the structured basis for driving the research forward i.e., by a conscious questioning of the fundamental questions and answers that are either implicitly or explicitly stated when the research question is posed.

For example, a pertinent question to ask for research into agents is the following: what is an agent? repeated re-examination of this question, taking in new developments in the field and re-appraising old approaches and definitions can lead to new insights that can drive the research forward. A meta-question to this question might be: why are we asking the question in the first place? One response might be as follows: we need to keep asking the question what is an agent? because there is much confusion over what people mean by an agent as they define it in different ways depending on the context. We need to make clear which definition people are using not just for research purposes but to help increase clarity and therefore, improve overall knowledge and understanding.

The field of A.I. provides a rich set of questions that can be examined at various levels and depths. For example, the fundamental question can machines think? is a good candidate for examination by meta-questioning. If we examine each word in the question, this leads to further questions. For example: can machines think? asks whether it will ever be possible for machines think; can machines think? emphasizes that the real question may be what are machines? and can machines think? questions what we mean by thinking. This last question leads to a much more fundamental question at the heart of this line of thinking What is thought? Let us examine this question using the meta-questioning process. One possible line of reasoning is shown (this represents a line of thought by the primary researcher of this study):

A short line of reasoning using meta-questioning:

- What is thought?
- Is it meta-questioning?
- But what is meta-questioning?
- What is questioning for that matter?
- And what is a question?

Let us examine these questions in more detail (this is itself a form of meta-questioning) since, the questions themselves have been kept short and need further elaboration. The first question is the one under examination which started the thought process. The second one proposes the concept of meta-questioning as a possible model for what thought is. Questions 3 and 4 delve deeper by examining the meaning of the underlying concepts.

Note that the answers to the questions have been couched as further questions. Also motivation seems to play an important role in this meta-questioning process. One of the motives of the researchers line of reasoning is to put forward meta-questioning as an alternative way or model of how an agent might go about reasoning and decision making. This motivation is what drives the thought process in this example. Ahn and Picard (2005) and other researchers have noted that for humans and other animals, motivation is essential to their learning and decision-making.

Further clarification is needed (and so the meta-questioning process continues). What is a model? For example, there are many models of thought some fit well while others do not. That is the model may only fit part of the process and other models may be required to fit other aspects of it.

For example, Gardenfors in his interesting book Conceptual Spaces: the geometry of thought (Gardenfors, 2000) attempts to unify the opposing camps of the classical symbolic approach to A.I. (which states that high level knowledge can be represented by symbols and logic) and the connectionist or sub-symbolic approach (which states that knowledge can be represented as a pattern of weights of neuron connections). He proposes a middle conceptual spaces layer between the symbolic and sub-symbolic layers (he prefers to use the term perspective rather than a layer or model). Concepts are represented as geometrical regions where axes or dimensions are properties of the concept. For example, colour spaces in humans appear to have three dimensions hue, brightness and chromaticity and taste appears to be generated from four distinct types of receptors: saline, sour, sweet and bitter.

Lee Carlson in his July 10 review on Amazon.com notes one shortcoming of Gardenfors book: it does not however, give any advice on how to implement its ideas into a real thinking machine. One of the motivations behind this study is to explore ideas that can be used as a starting point for work that may (or may not!) eventually lead to a thinking machine.

Note this study has provided a deliberately broad discussion (partially as an example of the complicated nature of thought itself) that reflects the current thinking concerning thought and knowledge and whose purpose has been to suggest question and answering as being fundamental to both. To some extent, the ideas we have discussed have involved a repeated recycling from different perspectives of two of the theme questions in this study: what is thought? and what is knowledge? In the recycling of these ideas however, although the same questions are being re-examined and re-searched, there is a process of knowledge evolution that of the ideas evolving to better suit some fitness criteria driven by motivation.

Researchers see knowledge evolution as being an important aspect of thought. This leads on to the two other main themes which are discussed in this study evolution and search (since, the process of evolution can be thought of as a process of search). The process of knowledge evolution is none more aptly demonstrated by the research of Horn (1999). Researcher has proposed a visual language as a form of visual thinking. Part of his research has involved the production of seven posters that summarize the Turing debate in A.I. to demonstrate his visual language and visual thinkin (Fig. 1). The posters clearly show the evolution of knowledge through a series of questions and meta-questions.

Knowledge sharing: Let us now consider how agents can interact and exchange knowledge (researchers call this knowledge sharing). This study will also present results for two applications searching in peer-to-peer networks and traffic simulation both of which provide strong evidence that knowledge sharing can help improve search effectiveness.

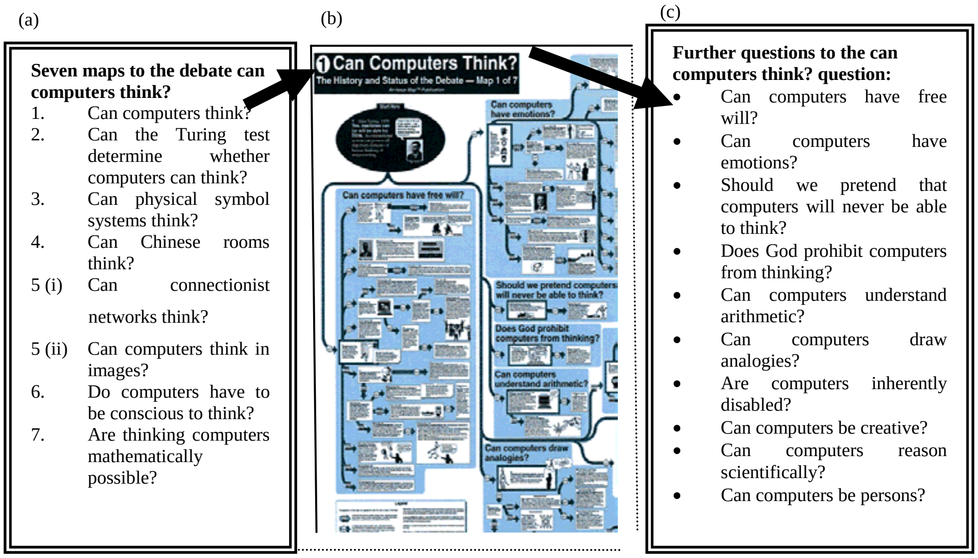

Fig. 1: a) The seven maps of important debates in Artificial Intelligence; b) Poster for the Can Computer Think? debate and c) Further questions which appear in b

Knowledgeable agents are equipped with background knowledge such knowledge is expanded as the agent perceives and makes inferences about the state of its environment. Knowledge attainment should not be seen as solely a consequence of the agent's interaction with the environment; it also gains knowledge through interaction between other agents and through sharing the knowledge.

Knowledge sharing between agents can occur in many ways. Researchers see agent communication as a separate issue. Communication involves message passing whereas researchers are more concerned here with the exchange and assimilation of knowledge, i.e. what is being communicated between the agents is key. If it concerns knowledge (as we have defined it above, i.e. how an agent can process information to perform a task or answer a question) then, knowledge is being shared rather than just information. Knowledge sharing amongst agents can be thought of as the means by which agents create a form of swarm intelligence. Knowledge sharing allows the individual agents to co-operate and exchange knowledge from their knowledge bases. The ability to share knowledge between agents is beneficial as it allows agents to gather knowledge about areas of the environment they have not explored. As the effectiveness of an agent in terms of selecting the most appropriate actions is determined by the amount and quality of knowledge it has the sharing of knowledge promotes a more effective agent. Consequently, agents that collaborate and share knowledge should discover solutions to problems more effectively than a single sophisticated agent working in isolation.

One well-known method of knowledge sharing (although not usually labelled as such) is word of mouth. The word of mouth method of communication between agents replicates the word of mouth communication between humans. Human communication is divided into two categories including mass and personal communication (Ohtani and Minami, 1998). Mass media is an example of mass communication. For example, newspapers and television provide information for many people in a one-to-many style. Conversely, personal communication is bi-directional in a one-to-one style. Therefore, information exchanged in one-to-one style and peer-to-peer fashion is recognised as word of mouth communication.

For example, word of mouth communication is employed by the WOM agent system (Ohtani and Minami, 1998). Agents are used by a search engine to retrieve relevant web documents. Each agent maintains a database which stores a portion of available web documents. When the user submits a query, a search request is sent to an available agent. The agent searches its database for relevant web documents if no relevant web documents are located then the search request is passed to other agents in a word of mouth fashion. This process continues until an agent does locate relevant web documents. Upon locating relevant web documents, the information is returned via the route the search query was sent. Each agent along the route copies and stores the reply data. Thus, popular web documents are propagated amongst agents. As the system propagates popular web documents between agents, maintenance is reduced as unpopular web documents are not distributed and are ignored. Furthermore, network load and traffic is reduced as the most accessible agent in terms of the user's network environment and location is selected to perform the initial search (Ohtani and Minami, 1998).

An API for simulating knowledge sharing amongst agents: An agent-based API has been designed to provide the functionality to simulate knowledge sharing amongst agents (Tuff, 2005). The sharing of knowledge can be performed in a word of mouth fashion whereby agents that meet as they traverse the network, exchange knowledge. Similarly, the API incorporates the blackboard knowledge distribution method. Knowledge can be distributed via agent accessible repositories or blackboards. Blackboards are a well-known architectural concept for sharing knowledge which was first coined by researchers in the field of A.I. (Ferber, 1999). The blackboard knowledge sharing method is an example of an indirect form of knowledge distribution.

The blackboard method does not permit direct knowledge distribution amongst agents; alternatively, agents distribute knowledge via repositories or blackboards situated at nodes in the network. Any agent traversing a node exchanges knowledge with the blackboard. In addition, the API also allows a simulation to be performed without any knowledge being distributed amongst agents. Consequently, the agents move around the network in a random fashion, until they reach their designated destination node.

The API has been applied to the problem of traffic simulation. The goal of an agent in this case is to reach its designated destination node. Upon reaching the destination node, the agent generates statistical information concerning its journey. The statistical information includes the amount of nodes traversed, distance travelled and journey time. This information is subsequently used to calculate the average nodes traversed, distance travelled and journey time for each agent. In addition, the percentage of agents successfully reaching the designated destination node can also be calculated. This information is used to evaluate and compare the effectiveness of the word of mouth and blackboard knowledge distribution methods, along with simulations devoid for the purpose of knowledge distribution.

Various traffic simulations were performed using the API on randomly generated networks and a manually generated network (corresponding to the England and Wales motorway structure). The path of each agent represents a simulated car journey; the driver has no knowledge of the road network prior to starting the journey and gathers knowledge by observing the network and through exchanges with other agents or blackboards. Figure 2 shows one sample network the network has been randomly populated with a number of agents (the small circles) with different destination nodes. The word of mouth configuration is shown on the left with the blackboard configuration on the right. The larger circles represent node objects and the lines between the nodes are links in the network. As the simulation progresses, the agents move from their starting node, along the network via nodes and links, until they reach their designated destination node.

Space precludes a full description of the results but a typical set for a randomly generated network of 100 nodes and 100 agents is shown in Table 1. The Table 1 shows the percentage of the assigned agents that find their designated destination node. In addition, the tables show the average amount of nodes traversed, distance covered and journey time of the agents as they travel from their starting node to their destination node.

The statistics show that the blackboard knowledge distribution method is the most successful in terms of agents reaching their destination node with 94% of agents reaching their destination. In addition, the blackboard simulation's agents on average traverse fewer nodes, cover less distance and experience a shorter journey time than the other methods. These results were repeated in all the other simulations. Whilst the blackboard method has been identified as the superior knowledge sharing method, both the word of mouth and blackboard share similarities in performance. Both methods are reliant on agents to transport knowledge to different areas of a given network. For example, blackboards remain static and the agents store and retrieve knowledge from the blackboards as they travel through the node. Likewise when employing the word of mouth method, agents exchange knowledge with other agents as they meet on their journey around the network. In addition, the agents perceive and gather knowledge regarding the topology and traffic levels of the network. Therefore, the

Table 1: Average agent statistics for a randomly generated traffic simulation network of 100 nodes and 100 agents

Parameters	No knowledge sharing	Word of mouth	Blackboard
Percentage of destinations found	21.00	89.00	94.00
Ave. No. of nodes traversed	23.19	18.35	15.94
Distance covered	3799.57	2918.92	2392.64
Time steps	4542.33	3507.65	2848.94

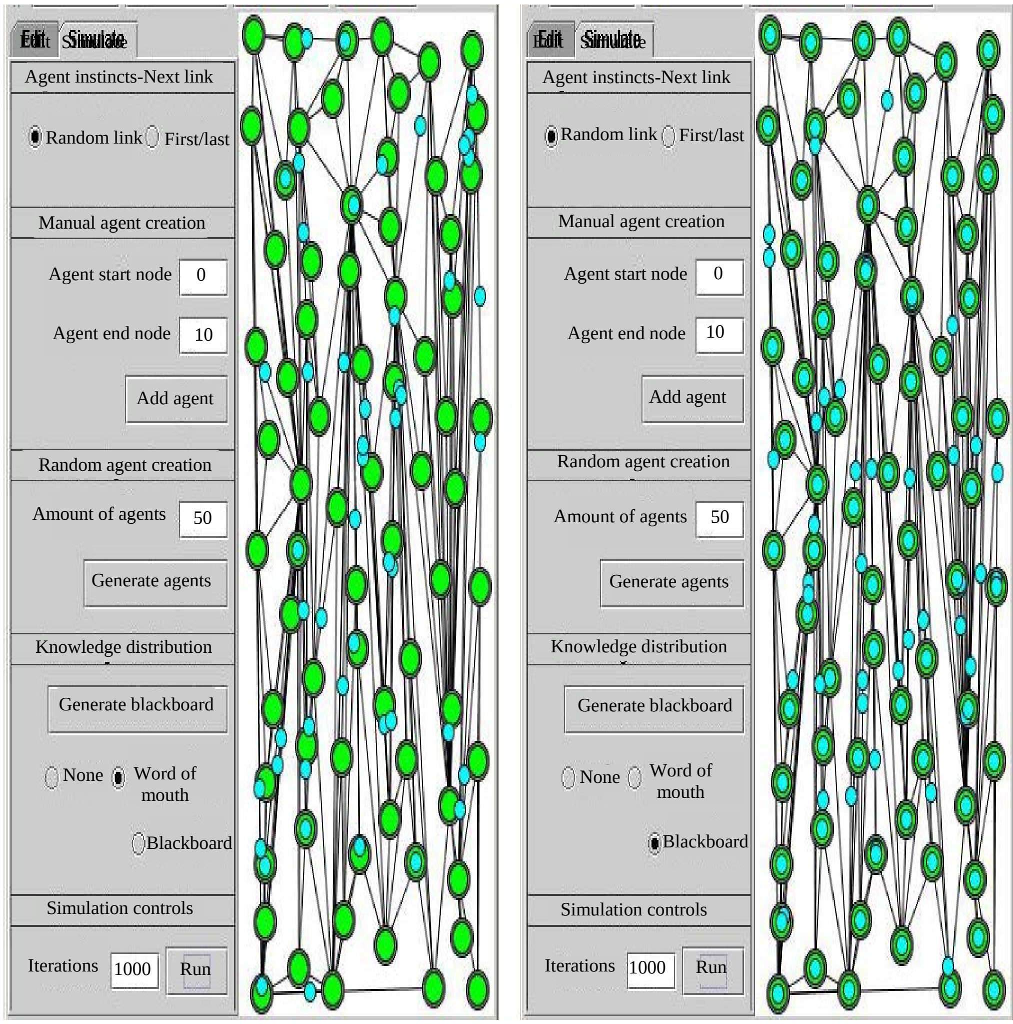

Fig. 2: Screenshots of Word-of-mouth and blackboard simulation using knowledge sharing API

greater the number of agents employed by a simulation with the size of the network remaining the same, the greater the volume of knowledge gathered and distributed amongst agents. As the number of agents added to a network is increased, the percentage of agents locating their destination nodes also increases.

Family, culture, evolution and search: Let us consider the role of knowledge to help improve search in the sense that the agents are able to find a more accurate solution or are able to adapt more quickly requiring less generations to evolve if evolutionary algorithms are being used. The motivation for this is the strong evidence detailed above that knowledge sharing can help improve search. The discussion in this section will focus specifically on genetic programming. We suggest a novel variation of the traditional genetic programming algorithm (as described by Koza (1994)) that incorporates knowledge sharing as outlined in the previous study (Fig. 3). Note that genetic evolution combined with knowledge sharing can be thought of as employing a kind of meta-questioning process if: firstly, the knowledge being shared is considered to be questions and answers and secondly as suggested below, the shared knowledge is being evolved at the same time as the individuals in the population are being evolved (In a sense, the meta-question being asked each generation is what are the best questions to ask? based on how well the answers fit what is found out during the evolutionary process).

First however, researhers will provide a brief review of some work directly related to that described in this section. Koza *et al*. (1999) has looked at the problem of incorporating domain knowledge into genetic programming. They state that the vast majority of

```
Procedure GPKS_Evolution (G, N, M, F, pC1, pC2, pX, pM)
/*  G = maximum number of generations to be run; N = size
    of population of computer programs; M = number of
    families;
    F = fitness evaluation function for the programs executed
    using the shared familial knowledge; pC1, pC2, pX, and pM
    are probabilities for cloning-with-knowledge, cloning-
    without-knowledge, crossover and mutation respectively.
    */
Assert pC1 + pC2 + pX + pM = 1; M ≤ N
Generate initial population of size N with M blackboards by
combining randomly selected functions and terminals. Set
initial knowledge in the M blackboards to blank.
For each generation
    Execute each program in the population and calculate its
    fitness using function F. Update blackboards with
    knowledge gained during execution.
    Set best-so-far = individual with best fitness.
    If termination criteria satisfied (e.g. generation ≥ G)
    Then Return best-so-far.
    Repeat
        Select genetic operator (cloning-with-knowledge,
        cloning-without-knowledge, crossover or mutation)
        based on probabilities pC1, pC2, pX and pM.
        Switch (operator)
            Cloning-with-knowledge: Select one program
            from current population, and copy it to new
            population. Copy its family's blackboard
            (unless it already exists).
            Cloning-without-knowledge: Select one
            program from current population, and copy it
            to new population and create a blank
            blackboard for it.
            Crossover: Select a pair of families, randomly
            choose one individual from each, and then
            perform crossover of the pair at a random point
            to create two new offspring for the new
            population. Combine the parents' blackboards
            to create the blackboard for the new offspring.
            Mutation: Select one program from current
            population, randomly change either a function
            with another function, or a terminal with
            another terminal, and add the mutant and its
            blackboard into new population.
          End Switch
    Until size of new population = N.
    Replace current population with new population.
End For
```

```
Procedure GPKS_Evolution (G, N, M, F, pC1, pC2, pX, pM)
/*  G = maximum number of generations to be run; N = size of
    population of computer programs; M = number of families;
    F = fitness evaluation function for the programs executed
    using the shared familial knowledge; pC1, pC2, pX, and pM are
    probabilities for cloning-with-knowledge, cloning-without-
    knowledge, crossover and mutation respectively. */
Assert pC1 + pC2 + pX + pM = 1; M ≤ N.
Generate initial population of size N with M blackboards by
combining randomly selected functions and terminals. Set initial
knowledge in the M blackboards to blank.
For each generation
    Execute each program in the population and calculate its
    fitness using function F. Update blackboards with
    knowledge gained during execution.
    Set best-so-far = individual with best fitness.
    If termination criteria satisfied (e.g. generation ≥ G) Then
    Return best-so-far.
    Repeat
        Select genetic operator (cloning-with-knowledge,
        cloning-without-knowledge, crossover or mutation)
        based on probabilities pC1, pC2, pX and pM.
        Switch (operator)
            Cloning-with-knowledge: Select one program
            from current population, and copy it to new
            population. Copy its family's blackboard (unless
            it already exists).
            Cloning-without-knowledge: Select one
            program from current population, and copy it to
            new population and create a blank blackboard for
            it.
            Crossover: Select a pair of families, randomly
            choose one individual from each, and then
            perform crossover of the pair at a random point to
            create two new offspring for the new population.
            Combine the parents' blackboards to
            create the blackboard for the new offspring.
            Mutation: Select one program from current
            population, randomly change either a function
            with another function, or a terminal with another
            terminal, and add the mutant and its blackboard
            into new population.
          End Switch
    Until size of new population = N.
    Replace current population with new population.
End For
```

Fig. 3: Pseudo-code for Koza's traditional Genetic Programming algorithm (on the left) and for modified algorithm with knowledge sharing (on the right)

contemporary researchers in artificial intelligence believe that a system for automatically creating computer programs must employ an explicit knowledge base i.e., study of a knowledge representation, knowledge acquisition, the codification of the knowledge into a knowledge base in a computer and the manipulation of the knowledge using formal logic inference methods. Koza *et al*. (1999) continue:

> Conspicuously, genetic programming does not rely on an explicit knowledge base to achieve the goal of automatically creating computer programs. While there are numerous optional ways to incorporate domain knowledge into a run of genetic programming, genetic programming does not require (or usually use) an explicit knowledge base to guide its search

Koza (1992) himself proposed introducing mechanisms for the manipulation of memory (or repositories) so that agents could be created capable of building and using stored representations. The most basic form of memory consisted of a fixed number of individually named storage locations and read and write functions were added to the programming language that was being evolved. Teller (1994) extended this by allowing read and write to a linear indexed array. In Teller's applications, agents had the task of pushing boxes to a wall; memory was required because agents could not sense whether the box was at a wall. Brave (1996) points out, however that Teller's agents did not employ a mental model or internal representation of the world. Andre (1994) extended Teller's approach by having the agents use memory to evolve mental models isomophic to the world; note that the agent's behaviour did not evolve but their mental model of the world stored in their memory did. Brave (1996) went further and had the agents create plans and dynamically evolve the mental model structure rather than have it fixed in advance as was the case for Andre. Spector and Luke (1996a, b) have devised a novel extension to genetic programming with results that indicate a knowledge-based approach to genetic programming has promise. Interestingly, their approach bears a strong similarity to the blackboard approach described above in that a central respository is being used to transfer the knowledge amongst the community of agents. Spector and Luke's study show how the performance of a genetic programming system can be improved through the addition of mechanisms for non-genetic transmission of information between individuals (they call this culture). Whereas Teller andre and Brave's research had previously shown how genetic programming systems were enhanced using memory, Spector and Luke applied a simple modification of the memory mechanism to allow for communication between individuals within and across generations. They allowed all individuals to share the same memory; this memory was initialized only at the start of a genetic programming run. A program was able to pass information to itself, to its contemporaries, to its offspring and to unrelated members of future generations. They showed that culture (implemented in this way using the shared indexed memory mechanism) can substantially reduce the computational effort required to solve various search problems. Spector and Luke also cite other approaches that have experimented with cultural elements within evolutionary systems but they distinguish their work through the straightforward shared use of Teller's memory mechanism.

Researchers will now describe a modification of Koza's traditional Genetic Programming algorithm that takes into account these ideas and combines them with the own ideas concerning knowledge sharing and knowledge evolution described above. The approach is to use blackboards for memory to share knowledge of how to answer questions using the knowledgeable agent framework described however, the knowledge being shared is not across the whole population but only amongst family members. Also, instead of just one shared memory (or blackboard), there are several, one for each family and the purpose of the genetic process is to determine the fitness of the family's knowledge as well as the fitness of the individual's program.

A key factor governing search effectiveness in GP is the fitness of each agent to survive in order to produce offspring. However, just as important as the fitness of the parent is the fitness of the offspring. The recombination of the parents' genes during reproduction may make the offspring less or more fit and this is what ultimately determines viability. Consider an alternative mimetic approach which factors into account the role of nurture as well as nature where parents play a further role to ensure the survival of their offspring. Assume that the best parents are able to ensure their offspring are the fittest they can be to ensure long-term survival. One way parents can do this is by transferring knowledge to their offspring. Parents who do not share their knowledge in an effective way will produce less viable offspring in the long term. Pseudo-code for the modified algorithm that takes into accounts these ideas is shown in Fig. 3. This is shown alongside Koza's original algorithm (based on the description in (Negnevitsky, 2002). In the approach, searching is governed by the fitness of the family and not just the individual. This involves the transfer of knowledge from the parents to their offspring (and so this is where knowledge sharing plays an important role as the parents who share/transfer their knowledge more effectively than other parents will ultimately produce fitter offspring).

The evolved programs themselves must make use in some way of the knowledge available to the parents. For this method to distinguish itself from standard GP then, this knowledge must be in a form that could not be incorporated directly into the evolved programs. That is the knowledge is external to the agent. For example: it only exists in the environment or the environment is dynamic (changing too rapidly for the process of evolution) or the knowledge is too complicated to evolve or it is knowledge that has been accumulated over a number of generations or it is distributed amongst two or more agents.

Assume each blackboard (i.e., the shared memory) is available only amongst the same family unit i.e., the original individuals and their direct descendants. In the first generation, blackboards are blank. Blackboards provide a means by which each individual can write knowledge they have gained during the execution of their evolved programs. This knowledge then becomes available to all future generations.

Like Koza's algorithm, there are the two standard operators crossover and mutation but for cloning, two choices are possible cloning with and without knowledge. The latter operator is the only operator where propagation of knowledge in the blackboards does not occur.

The fitness evaluation function should be a measure of the fitness of the family's knowledge as well as the individual. One way this can be done is by having the individual program's behavior determined in some way by the use of the stored family knowledge. This alleviates the problem of destructive individuals mentioned by Spector and Luke (1996b). Since, the primary purpose of the individual is to ensure the survival of the family unit then, there is no advantage to destroying useful information so that other individuals cannot take advantage of it (since, only family members have access to their shared knowledge).

Knowledge fusion also becomes an interesting issue. This is required when conflicting knowledge exists in both parents' blackboards (this is when the parents have different answers to the same question for example in searching for P2P networks, the families may have learnt to find a resource using different paths). One solution is to record knowledge effectiveness alongside the knowledge itself (i.e., the blackboard stores the answer to the meta-question how good is this knowledge? Then, knowledge can be fused by choosing the most effective knowledge from either the maternal or paternal blackboard. Other possibilities exist of course. For example, one could adopt a mother is always right policy instead where knowledge from the paternal side is ignored unless the knowledge is new.

The parameter M is the number of blackboards used by the algorithm. Different values of M determine the overall behavior of the system. If $M = 0$, for example then the behavior corresponds to Koza's traditional algorithm shown on the left of Fig. 3. If $M = 1$ and $p_{C2} = 0$ then this is equivalent to Spector and Luke (1996b)'s single repository cultural approach. If $M = 2$ and $p_{C2} = 0$, the extra blackboard is essentially redundant across generations since, it is almost certain at least one pair from both families will be chosen to mate so, both families will end up with the same knowledge. However, the blackboards do provide the means for two sets of knowledge to develop independently within the same generation and if knowledge effectiveness is recorded then, this can be exploited by the next generation.

M could be allowed to increase (up to N) or M could be fixed then some families will lose out on the chance to propagate their knowledge if $p_{C2}>0$. In a sense, p_{C2} determines the rate of knowledge rebirth in this manner as it ensures new search paths will be pursued.

Higher values of M are needed if independently evolved familial knowledge is to occur. If $M = N$ then, this means that each individual in the population will have their own blackboard and exchange of knowledge will only occur across generations when an individual is selected to mate. Note that in this case, the probability p_{C2} will determine how much of the population will fail to have their knowledge propagate into the next generation.

CONCLUSION

Researchers have proposed a novel variation of Koza's algorithm for evolving computer programs. The ideas behind the algorithm are based on experimental results in two search applications file searching in peer-to-peer networks and traffic simulation where it was found that improved search performance can be achieved through knowledge sharing between agents. Researchers have also proposed a novel way of structuring the knowledge this is based on a question and answering framework researchers have devised for knowledgeable agents and on the use of meta-questioning (actively questioning the effectiveness of both the questions and answers). However, researchers have few experimental results to support the ideas as yet and the current state of the work raises more questions than answers.

REFERENCES

Ahn, H. and R.W. Picard, 2005. Affective-cognitive learning and decision making: A motivational reward framework for affective agents. Proceedings of the 1st International Conference on Affective Computing and Intelligent Interaction, October 22-24, 2005, Beijing, China.

Andre, D., 1994. Evolution of mapmaking: Learning, planning, and memory using genetic programming. Proceedings of the 1st IEEE Conference on Evolutionary Computation, June 27-29, 1994, Orlando, FL, USA.

Baird, J.E., 1976. Sex differences in group communication: A review of relevant research. Q. J. Speech, 62: 179-192.

Baum, E.B., 2004. What Is Thought?. MIT Press, Cambridge, MA.

Brave, S., 1996. The evolution of memory and mental models using genetic programming. Proceedings of the 1st Annual Conference on Genetic Programming, July 28-31, 1996, MIT Press, Cambridge MA., pp: 261-266.

Cannataro, M. and D. Talia, 2003. The knowledge grid. Commun. ACM., 46: 89-93.

Clifton, T. and W.J. Teahan, 2005. Knowing-aboutness: Question-answering using a logic-based framework. Proceedings of the 27th European Conference on Information Retrieval Research, April 2-5, 2005, Rome, pp: 230-244.

Ferber, J., 1999. Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. 1st Edn., Addison-Wesley Longman Inc., New York.

Gardenfors, P., 2000. Conceptual Spaces: The Geometry of Thought. The MIT Press, Cambridge Massachusetts.

Harrub, B. and B. Thompson, 2004. What about animal consciousness? Apologetics Press, Reason and Revelation, http://64.246.50.221/articles/505.

Hearst, M.A., 1999. Trends and controversies: Mixed-initiative interaction. IEEE Intel. Syst., 14: 14-23.

Hillis, D., 2004. The unreasonable man. Aristotle: The Knowledge Web, http://www.unreasonableman.net/2004/05/aristotle_the_k.html.

Horn, R.E., 1999. Visual Language: Global Communication for the 21st Century. 1st Edn., MacroVU Press, Bainbridge Island, ISBN-13: 978-1892637093, Pages: 270.

Koza, J.R., 1992. Genetic Programming: On the Programming of Computers by Means of Nature Selection. MIT Press, Cambridge, MA., ISBN: 0-262-11170-5.

Koza, J.R., 1994. Genetic Programming II: Automatic Discovery of Reusable Programs. MIT Press, Cambridge, ISBN-13: 9-780-262-11189-8.

Koza, J.R., F.H. Bennett III, D. Andre and M.A. Keane, 1999. Seven differences between genetic programming and other approaches to machine learning and artificial intelligence. http://www.genetic-programming.com/sevendiffs.html.

Maher, M.L. and N. Gu, 2002. Designing virtual architecture: From place to user centred design. Int. J. Design Comput. Vol., 4.

Mayfield, J., T. Finin, R. Narayanaswamy and C. Shah, 1995. The CYCic friends network: Getting CYC agents to reason together. Proceedings of the Workshop on Intelligent Information Agents, December 1-2, 1995, Baltimore, Maryland.

Negnevitsky, M., 2002. Artificial Intelligence: A Guide to Intelligent Systems. 2nd Edn. Addison-Wesley, New York.

Ohtani, T. and T. Minami, 1998. The word-of-mouth agent system for finding useful Web documents. Proceedings of the Asia Pacific Web Conference, September 27-30, 1998, Beijing.

Russell, S.J. and P. Norvig, 2003. Artificial Intelli-gence: A Modern Approach. 2nd Edn. Prentice-Hall, USA.

Spector, L. and S. Luke, 1996b. Cultural transmission of information in genetic programming. Proceedings of the 1st Annual Conference on Genetic Programming, July 28-31, 1996, The MIT Press, Cambridge, MA., pp: 209-214.

Spector, L. and S. Luke, 1996a. Culture enhances the evolvability of cognition. Proceedings of the Cognitive Science and Society Meeting, July 12-16, 1996, San Diego, USA., pp: 10-17.

Teahan, W.J., 2005. Agent-oriented systems for teaching, learning and research. Technical Report AIIA: 05.4, University of Wales, Bangor.

Teahan, W.J., 2003. Knowing about knowledge: Towards a framework for knowledgeable agents and knowledge grids. Technical Report AIIA: 03.2, University of Wales, Bangor.

Teller, A., 1994. The Evolution of Mental Models. In: Advances in Genetic Programming, Kinnear, K.E.Jr. (Ed.). MIT Press, Cambridge, MA., pp: 199-219.

Tuff, P.G., 2005. An agent-based framework for knowledge distribution. Masters Thesis, University of Wales, Bangor, UK.

Turing, A.M., 1950. Computing machinery and intelligence. Mind, 59: 433-460.

Wooldridge, M., 2002. An Introduction to MultiAgent Systems. John Wiley and Sons, Chichester, England.