

# Artificial Intelligence Network Data Planning in Ant Colony Optimization

M.Sibisakkaravarthi<sup>1</sup> & P.Subramaniam<sup>2</sup>

<sup>1</sup>Research Scholar, Department of Computer Science,  
Muthayammal College of Arts & Science, India

<sup>2</sup>Hod & Associate Professor, Department of Computer Science,  
Muthayammal College of Arts & Science, India

**Abstract**— In this paper we propose particular notability of ants is their ability to create "ant streets". Long, bi-directional lanes of single file pathways in which they navigate landscapes in order to reach a destination in optimal time. These ever-changing networks are made possible by the use of pheromones which guide them using a shortest path mechanism. This technique allows an adaptive routing system which is updated should a more optimal path be found or an obstruction placed across an existing pathway. Computer scientists began researching the behaviour of ants in the early 1990's to discover new routing algorithms. The result of these studies is Ant Colony Optimisation (ACO) and in the case of well implemented ACO techniques, optimal performance is comparative to existing top-performing routing algorithms.

This article details how ACO can be used to dynamically route traffic efficiently. An efficient routing algorithm will minimise the number of nodes that a call will need to connect to in order to be completed thus; minimising network load and increasing reliability. An implementation of ANTNet based on Marco Dorigo & Thomas stützle has been designed and through this a number of visually aided test were produced to compare the genetic algorithm to a non-generic algorithm. The report will final conclude with a summary of how the algorithm perform and how it could be further optimised.

**Keywords**—Ant Colony Optimization, Artificial Neural Networks, Routing Algorithm, Network

## I. INTRODUCTION

The notion of complex collective behaviour emerging from the behaviour of many relatively simple units, and the interactions between them, is fundamental to the field of artificial life. The growing understanding of such systems offers the prospect of creating artificial systems which are controlled by such emergent collective behaviour; in particular, we believe that the exploitation of this concept might lead to completely new approaches for the management of distributed systems, such as load balancing in telecommunications networks. In such networks, Calls between two points are typically routed through a number of intermediate switching stations, or nodes; in a large network, there are many possible routes for each such call. It is thus possible to relieve actual or potential local congestion by routing calls via parts of the network which have spare capacity. Load balancing is essentially the construction of call –routing schemes which successfully distribute the changing load over the system and minimise lost calls. Of course it is possible to determine the shortest routes from every node to every other node of the network. In this way the average utilisation of nodes will be minimised, but this is not necessarily the ideal way to avoid node congestion, as this has to do with how the traffic on the network is distributed. Controlling distributed systems like these by means of a single central controller has several disadvantages.

The controller usually needs current knowledge about the entire system, necessitating communication links from every part of the system to the controller. These central control mechanisms scale badly, due to the rapid increase of processing and communication overheads with system size. Failure of the controller will often lead to failure of the complete system. There is the additional practical commercial requirement that centrally controlled systems may need to be owned by one single authority. Further, the nature of distributed systems like these is highly dynamic, complex and stochastic, and their behaviour can neither be predicted nor explained by reducing it to a single central controllable factor. A good decentralized control mechanism will not have the problems mentioned above. The field of artificial life has given us inspiration for such a mechanism that will be completely distributed, and highly adaptive to changes in the network and traffic patterns. This solution makes use of the distributed processing capability already inherently present in the network in the form of Introduction Load balancing in a network using Ant colony optimization technique 3 network nodes. The distributed nature of such an approach may make the system very robust against failures of individual control entities. Our approach is inspired by the work of biologists studying social insects, who have uncovered the mechanisms controlling the foraging behaviours of ants. The most important method is the

laying and sensing of trails of pheromones -specialised chemical substances which are laid in amounts determined by local circumstances, and which by their local concentration subsequently directly influence an ant's choice of route.

#### A. Artificial intelligence (AI)

Artificial intelligence (AI) is defined as intelligence exhibited by an artificial entity. Such a system is generally assumed to be a computer. Although AI has a strong science fiction connotation, it forms a vital branch of computer science, dealing with intelligent behaviour, learning and adaptation in machines. Research in AI is concerned with producing machines to automate tasks requiring intelligent behavior. Examples include control, planning and scheduling, the ability to answer diagnostic and consumer questions, handwriting, speech, and facial recognition. As such, it has become a scientific discipline, focused on providing solutions to real life problems. AI systems are now in routine use in economics, medicine, engineering and the military, as well as being built into many common home computer software applications, traditional strategy games like computer chess and other video games.

## 2. SYSTEM STUDY AND ANALYSIS

### A. EXISTING SYSTEM

Early work in this area focused to fuse information coming from both modalities has been applied to problems such as tracking of humans under surveillance and smart videoconferencing. Typically, the sensors are a video camera and an acoustic array (not necessarily collocated). The acoustic time-delay-of arrivals derived from the peaks of the generalized cross-correlation function, are used along with active contours to achieve robust speaker tracking with fast lock recovery. The tracking humans using audio-visual cues, based on foreground detection and image-differencing. The visual appearance models should be calculated online as opposed to using trained models for tracking. Although fixed image templates (e.g., frames) are very useful for face tracking, they are not effective for tracking neural data in outdoor environments. In vehicle-tracking problems, acoustics and video asynchronization causes biased localization estimates that can lead to filter divergence. This is because the bias in the fused cost function increases the video's susceptibility to drift in the background. Previous implementations make use of the basic bootstrap particle filter, whereas a more general approach involves the concept of importance sampling. None of the above works can handle the problems are continuously inferring, from audio and video data, the location and speaking status for several people in a realistic conversational setting. Many of these works use simple sensor configurations (e.g. one camera and a microphone pair). Among the existing techniques, probabilistic generative models based on approximate inference methods (both variation and sampling-based) appear to be the most promising, given their principled formulation and demonstrated performance.

### B. PROPOSED SYSTEM

The hybrid nodes that contain an acoustic array collocated with a camera was proposed for vehicle tracking problems. To intelligently fuse information coming from both modalities, novel strategies for detection and data association have to be developed to exploit the multimodal information. Moreover, the fused tracking system should be able to sequentially update the joint state vector that consists of multiple target motion parameters and relevant features (e.g., shape, color and so on), which is usually only partially observable by each modality. By using particle filters, whose proposal function uses audio cues, have better speaker tracking performance under visual occlusions.

To track neural data using acoustic and video measurements, we propose a particle filtering solution that can handle multiple sensor modalities. We use a fully joint tracker, which combines the video particle filter tracker and a modified implementation of the acoustic particle filter tracker at the state-space level. We emphasize that combining the output of two particle filters is different from formulating one fully joint filter or one interacting filter. The resulting filter has a lower Kullback-Leibler distance to the true target posterior than any output combination of the individual filters. Hence, by fusing the acoustic and video modalities.

## 3. LITERATURE REVIEW

Load balancing plays an essential role in providing quality of service (QoS) guarantees in cloud computing, and it has been generating substantial interest in the research community. There are a great deal of approaches that have coped with the load balancing problem in cloud computing. We discuss the previous related work of load balancing by dividing them into two classes according to the underlying algorithm. The first class consists of diverse conventional approaches without utilizing any kind of

swarm intelligence algorithms. Many load balancing approaches were proposed in recent years and each focused on different aspects of algorithms and policies, e.g., leveraging a central load balancing policy for virtual machines, the scheduling strategy on load balancing of virtual machine (VM) resources based on genetic algorithms, a mapping policy based on multi-resource load balancing for virtual machines, adaptive distributed algorithm for virtual machines, weighted least-connection strategy, and two-phase scheduling algorithms.

Additionally, several methods of load balancing were presented for different cloud applications, for example, a service-based model for large scale storage, data center management architecture, and a heterogeneous cloud. Although these contributions have made great progress in load balancing under cloud computing, it has a high degree of centralization and is not easy to extend. Furthermore, these presented approaches did not fully reflect the characteristics of resource nodes and are more suitable to the static situation of cloud computing.

The second class contains approaches use swarm intelligence algorithms, such as ant colony optimization artificial bee colony, and particle swarm optimization which is better for the dynamic situation of cloud computing. With self-organized behavior, these social insects can be imitated as such, or with necessary modifications, to solve analogous problems in cloud computing. In Nishant, K. *et al.* proposed an algorithm for load distribution of a workload with a modified approach of ACO from the perspective of cloud or grid network systems. In this approach, the ants only updated a single result set continuously in the process, rather than updating their own result set. In a load balancing mechanism was proposed based on ant colony and complex network theory in an open cloud computing federation. This is the first time that ACO and complex networks were introduced together into load balancing in cloud computing and obtained good performance. In Mishra, R. *et al.* gave a solution to load balancing in the cloud by ACO, to maximize or minimize different performance parameters, such as CPU load and memory capacity. However, few factors were considered as pheromones to find target nodes when using ACO in the above three approaches.

In Sesum-Cavic, V. *et al.* presented a novel approach for load balancing based on artificial bee colony. A generic architecture, named SILBA (self-initiative load balancing agents), was defined to support the exchange of different algorithms through plugging techniques. Six algorithms were applied in this architecture and the results demonstrated promising benefits in the Amazon EC2 cloud. Although SILBA is a good pattern, it did not take into account the lower demands for node servers in cloud computing environments and the dynamic user needs. Particle swarm optimization (PSO) was also adopted for load balancing in cloud computing, such as it proposed a new task scheduling model to avoid the serious load imbalance problem, with improvement of the standard PSO by introducing a simple mutation mechanism and a self-adapting inertia weight method. To solve the optimization problem of discrete space in cloud computing, Feng, X. *et al.* constructed an appropriate resource-task model based on a discrete particleswarm optimization algorithm.

The experiment results showed that the discussed PSO methods can enhance the utilization in load balancing of resources ,but they may take a large amount of time with a huge number of tasks. Other applications and research on load balancing with swarm intelligence algorithms can be found in. Since these algorithms were originally designed for distributed load balancing rather than cloud computing, much work needs to be done if we want to apply these algorithms into cloud computing. In this paper, we focus on how to utilize ACO to establish a model of dynamic load balancing for cloud computing, while fully considering the characteristics of cloud computing itself. Moreover, we emphasize on well-defined strategies of pheromone update, in order to not only avoid falling into a local optimum, but also improve the convergence speed and accuracy.

#### A. Form and Content Issues in Architectures

In computer science, a programming language corresponds to a virtual architecture. A specific program in that language describes a particular (virtual) machine, which then responds to various inputs in ways defined by the program. The architecture is thus what Newell calls the fixed structure of the information processor that is being analyzed, and the program specifies a variable structure within this architecture. We can regard the architecture as the *form* and the program as the *content*, which together fully instantiate a particular information processing machine. We can extend these intuitions to types of machines which are different from computers. For example, the connectionist architecture can be abstractly specified as the set  $\{\{N\}, \{nI\}, \{nO\}, \{z_i\}, \{w_{ij}\}\}$ , where  $\{N\}$  is a set of nodes,  $\{nI\}$  and  $\{nO\}$  are subsets of  $\{N\}$  called input and output nodes respectively,  $\{z_i\}$  are the functions computed by the nodes, and  $\{w_{ij}\}$  is the set of weights between nodes. A particular connectionist machine is then instantiated by the "program" that specifies values for all these variables.

We have discussed the prospects for separating intelligence (a knowledge state process) from other mental phenomena, and also the degree to which various theories of intelligence and cognition balance between fidelity to biology versus functionalism. We have discussed the sense in which alternatives such as logic, decision tree algorithms, and connectionism are all alternative languages in which to couch an information processing account of cognitive phenomena, and what it means to take a Knowledge Level stance towards cognitive phenomena. We have further discussed the distinction between form and content theories in AI. We are now ready to give an overview of the issues in cognitive architectures. We will assume that the reader is already familiar in some general way with the proposals that we are discussing. Our goal is to place these ideas in perspective.

### *B. Intelligence as Just Computation*

Until recently the dominant paradigm for thinking about information processing has been the Turing machine framework, or what has been called the discrete symbol system approach. Information processing theories are formulated as algorithms operating on data structures. In fact AI was launched as a field when Turing proposed in a famous paper that thinking was computation of this type (the term "artificial intelligence" itself was coined later). Natural questions in this framework would be whether the set of computations that underlie thinking is a subset of Turing-computable functions, and if so how the properties of the subset should be characterized.

Most of AI research consists of algorithms for specific problems that are associated with intelligence when humans perform them. Algorithms for diagnosis, design, planning, etc., are proposed, because these tasks are seen as important for an intelligent agent. But as a rule no effort is made to relate the algorithm for the specific task to a general architecture for intelligence. While such algorithms are useful as technologies and to make the point that several tasks that appear to require intelligence can be done by certain classes of machines, they do not give much insight into intelligence in general.

### *C. Is the Search View of Deliberation Too Narrow*

A criticism of this picture of deliberation as a search architecture is that it is based on a somewhat narrow view of the function of cognition. It is worth reviewing this argument briefly. Suppose a Martian watches a human in the act of multiplying numbers. The human, during this task, is executing some multiplication algorithm, i.e., appears to be a multiplication machine. The Martian might well return to his superiors and report that the human cognitive architecture is a multiplication machine. We, however, know that the multiplication architecture is a fleeting, evanescent virtual architecture that emerged as an interaction between the goal (multiplication) and the procedural knowledge of the human. With a different goal, the human might behave like a different machine. It would be awkward to imagine cognition to be a collection of different architectures for each such task; in fact, cognition is very plastic and is able to emulate various virtual machines as needed.

Is the problem space search engine that has been proposed for the deliberative architecture is also an evanescent machine? One argument against it is that it is not intended for a narrow goal like multiplication, but for all kinds of goals. Thus it is not fleeting, but always operational. Or is it? If the sole purpose of the cognitive architecture is goal achievement (or "problem solving"), then it is reasonable to assume that the architecture would be hard-wired for this purpose. What, however, if goal achievement is only one of the functions of the cognitive architecture, common though it might be? At least in humans, the same architecture is used to daydream, just take in the external world and enjoy it, and so on. The search behavior that we need for problem solving can come about simply by virtue of the knowledge that is made available to the agent's deliberation from long term memory. This knowledge is either a solution to the problem, or a set of alternatives to consider. The agent, faced with the goal and a set of alternatives, simply considers the alternatives in turn, and when additional subgoals are set, repeats the process of seeking more knowledge. In fact, this kind of search behavior happens not only with individuals, but with organizations. They too explore alternatives, but yet we don't see a need for a fixed search engine for explaining organizational behavior. Deliberation of course has to have the right sort of properties to be able to support search.

Certainly adequate working memory needs to be there, and probably there are other constraints on deliberation. However, the architecture for deliberation does not have to be exclusively a search architecture. Just like the multiplication machine was an emergent architecture when the agent was faced with that task, the search engine could be the corresponding emergent architecture for the agent faced with a goal and equipped with knowledge about what alternatives to consider. In fact, a number of other such emergent architectures built on top of the deliberative architecture have been studied earlier in our work on Generic Task architectures [1986]. These architectures were intended to capture the needs for specific classes of goals (such as classification). The above argument is not to deemphasize the importance of problem space search for goal achievement, but to resist the identification of the architecture of the conscious processor with one exclusively intended for search. The problem space architecture is still important as the virtual architecture for goal-achieving, since it is a common, though not the only, function of cognition. Of course, that cognition goes beyond just goal achievement is a statement about human cognition. This is to take a biological rather than a functional standard for the adequacy of an architectural proposal. If we take a functional approach and

seek to specify an architecture for a function called intelligence which itself is defined in terms of goal achievement, then a deliberative search architecture working with a long term memory of knowledge certainly has many attractive properties for this function, as we have discussed.

#### *D. Alternative Proposals*

The various proposals differ along a number of dimensions: what kinds of tasks the architecture performs, degree of parallelism, whether it is an information processing architecture at all, and, when it is taken to be an information processing architecture, whether it is a symbolic one or some other type.

With respect to the kind of tasks the architecture performs, we mentioned Newell's view that it is just a recognition architecture. Any smartness it possesses is a result of good abstractions and good indexing, but architecturally, there is nothing particularly complicated. In fact, the good abstractions and indexing themselves were the result of the discoveries of deliberation during problem state search. The real solution to the problem of memory, for Newell, is to get chunking done right: the proper level of abstraction, labeling and indexing is all done at the time of chunking. In contrast to the recognition view are proposals that see relatively complex problem solving activities going on in subdeliberative cognition. Cognition in this picture is a communicating collection of modular agents, each of whom is simple, but capable of some degree of problem solving. For example, they can use the means-ends heuristic (the goal-subgoaling feature of deliberation in the Soar architecture).

Deliberation has a serial character to it. Almost all proposals for the subdeliberative architecture, however, use parallelism in one way or another. Parallelism can bring a number of advantages. For problems involving similar kinds of information processing over somewhat distributed data (like perception), parallelism can speed up processing. Ultimately, however, additional problem solving in deliberation may be required for some tasks.

## 4. CONCLUSION

In this paper we conclude that if the machine could successfully pretend to be human to a knowledgeable observer then you certainly should consider it intelligent. AI systems are now in routine use in various field such as economics, medicine, engineering and the military, as well as being built into many common home computer software applications, traditional strategy games etc. AI is an exciting and rewarding discipline. AI is branch of computer science that is concerned with the automation of intelligent behavior. The revised definition of AI is - AI is the study of mechanisms underlying intelligent behavior through the construction and evaluation of artifacts that attempt to enact those mechanisms. So it is concluded that it work as an artificial human brain which have an unbelievable artificial thinking power.

## *References*

- [1] Artificial Intelligence and cognitive science © 2006, Nils J. Nilsson Stanford AI Lab. Off-line Handwriting Recognition Using Artificial Neural Networks © 2000, Andrew T. Wilson University of Minnesota, Morris <http://wilsonat@mrs.umn.edu>
- [2] Neural Networks to produce an Adaptive quality gratitude System © 2002, Alexander J. Faaborg Cornell University, Ithaca NY
- [3] Hand-Printed Character Recognizer using Neural Network © 2000, Shahzad Malik
- [4] Neural Networks and Fuzzy Logic. © 1995, Rao, V., Rao, H. MIS Press, New York
- [5] Neural Networks © 2003, StatSoft Inc. <http://www.statsoft.com/textbook/stneunet.html>.
- [6] M.Balaji, E.Aarthi, K.Kalpana, B.Nivetha, D.Suganya "Adaptable and Reliable Industrial Security System using PIC Controller" 2017/5, Journal International Journal for Innovative Research in Science & Technology, Volume 3, Issue 12, Page 56-60.
- [7] M.Ravi A.M.Nirmala, P.Subramaniam, A.Anusha Priya, Enriched Performance on Wireless Sensor Network using Fuzzy based Clustering Technique, 2013, International Journal of Advanced Studies in Computer Science and Engineering, Volume 2, Issue 3, Pages 11-17.