A Roadmap of Nature-Inspired Systems Research and Development

Enda RIDGE [a], Edward CURRY [b]
[a] *The Department of Computer Science, The University of York, YO10 5DD, England. Tel: +44 1904 434731. Fax: +44 1904 432767. Email: ERidge@cs.york.ac.uk*
[b] *The Department of Information Technology, National University of Ireland, Galway. Email: EdCurry@acm.org*

Abstract. Nature-inspired algorithms such as genetic algorithms, particle swarm optimisation and ant colony algorithms have successfully solved computer science problems of search and optimisation. The initial implementations of these techniques focused on static problems solved on single machines. These have been extended by adding parallelisation capabilities in the vein of distributed computing with a centralised master/slave approach. However, the natural systems on which nature-inspired algorithms are based possess many additional characteristics that are of potential benefit within computing environments. In this paper, we discuss the benefits of nature-inspired techniques within modern and emerging computing environments. Software entities within these environments execute and interact in a fashion that is parallel, asynchronous, and decentralised. Given that the natural environment is in itself parallel, asynchronous and decentralised, nature-inspired techniques are an excellent fit for computing environments that exhibit these characteristics. Future research challenges for nature-inspired techniques within emerging computing environments are also discussed.

Keywords. nature-inspired systems, parallel, asynchronous, decentralised

1. Introduction

Nature-inspired techniques are problem solving approaches that share the characteristic of being loosely based on a natural metaphor. Such metaphors include evolution's search through the vast space of potential organisms and the coordinated movement of flocks of birds through a three dimensional space. Further examples of natural metaphors and their corresponding nature-inspired techniques are listed in Table 1.

Nature-inspired algorithms such as genetic algorithms [7], particle swarm optimisation [11] and ant colony algorithms [3] have achieved some remarkable successes. Indeed, they are the solution technique of choice for some problems. With a few exceptions, nature-inspired techniques have primarily been applied in the domain of heuristics. That is, these techniques have been used to create approximate algorithms for hard search and optimisation problems. The solutions they offer are beginning to draw attention from technology investors [4], resulting in the establishment of several companies that develop and deploy these technologies. Bonabeau and Meyer [4] cite several large multinational companies such as Unilever, McGraw-Hill, Capital One, Hewlett-Packard, and France Télécom that have benefited from nature-inspired approaches. Some smaller companies are also beginning to consult on swarm technologies[1].

Applications have, for the most part, been to static problems solved on single machines. Any attempts at parallelisation of these heuristics have been in the vein of traditional distributed computing with a centralised master/slave approach. This is, of course, a sensible approach when engineering performance improvements for the original domains tackled.

However, the natural systems on which these successful heuristics are based possess many other interesting properties.

- They typically contain massive numbers of relatively simple participants.
- They are completely decentralised.
- They operate in parallel and asynchronously.
- They use relatively simple signals for communication.
- Their desired functionality emerges from the interactions of their participants. This is achieved despite (and because of) their simple participants that have no global information.

These characteristics make natural systems robust to loss of members, inherently parallel in 'execution', and evidently adaptable to a highly dynamic problem domain—their natural environment.

Depending on circumstances, we can consider these as properties that are *desirable* for a computer system or as properties that are *enforced* on the systems we develop. As an example of the former, simple communication and emergent functionality reduce communication costs and eliminate centralised bottlenecks. As an example of the latter, modern computing environments such as the Internet, Grids and Enterprise Computing contain massive numbers of participants interacting and executing in a fashion that is (1) parallel, (2) asynchronous and (3) decentralised[2].

Each of these three characteristics poses a challenge for the development of software entities that reside in the environment. These include scalability, robustness, adaptability, manageability, redundancy, cooperation, and coordination. Given that the natural environment is in itself parallel, asynchronous and decentralised, nature-inspired techniques are an excellent fit for computing environments that exhibit these characteristics. It is vital that disciplined scientific and engineering investigations are undertaken to successfully transfer these algorithms, techniques and infrastructures into emerging computing environments.

Section 2 reviews some examples of research that has applied nature-inspired techniques to computing environments that are parallel, asynchronous and/or decentralised. Section 3 details our vision of the most important challenges and open questions for this field. Section 4 summarises our conclusions.

---

[1] AntOptima (http://www.antoptima.com/) and EuroBios (http://www.eurobios.com/)

[2] One should note that *distribution* is often assumed within these environments. However, the environment can be parallel, asynchronous, and decentralised without being distributed.

2. Nature-Inspired Systems for Parallel, Asynchronous and Decentralised Environments (NISPADE)

Since nature-inspired systems are the foundation of NISPADE research, we begin with an overview of the most popular traditional systems and highlight their current limitations.

2.1. Traditional Nature-Inspired Systems

One of the most well-established nature-inspired techniques is the Genetic Algorithm (GA) [7]. This approach is based on nature's Darwinian evolution. Potential solutions to a search or optimisation problem are encoded as strings, reminiscent of nature's DNA. These solutions are evaluated with a fitness function that determines how good a solution they represent. The fittest solutions are selected as parents to breed the next generation of solutions. Other genetic operators such as crossover and mutation can be performed on the encoding strings. These add a randomising component to the search process. After many such generations, the evolved solutions improve in fitness. Traditionally, GA implementations use a centralised comparison of all population members and a synchronised fitness evaluation phase. Parallel extensions of this have used a master/slave approach.

Nature-inspired systems based on social insect colonies have grown in popularity in recent years. We can see two major behaviours that have been adapted into successful algorithms. The first of these is *marker-based stigmergy*. This occurs when social insects such as ants and termites place artificial markers in the environment to help coordinate their actions. In ant foraging, one of the most popular sources of inspiration for ant algorithms, the ants use chemical markers called *pheromones* to guide one another along routes to food. This indirect communication via the environment allows relatively simple ants with limited local information to find optimal routes across relatively large distances. This foraging behaviour has been adapted to find paths through graphs, for example [6]. A second major behaviour is called *sematectonic stigmergy*. This involves insects adjusting their behaviours according to specific environment states. The most successful algorithms to adapt sematectonic stigmergy have been based on ant colony brood sorting. In the natural environment, some ant species move and sort their broods and cemeteries based on environment states such as the size of larvae adjacent to one another and the sizes of clusters of larvae they observe. When combined with probabilistic decision processes and thresholds for activation of behaviours, colonies of insects perform remarkable sorting without any global knowledge of the order they are creating. Again, ant colony algorithms typically use globally calculated pheromone updates and synchronised pheromone trail laying.

Of course, one of the most awe-inspiring natural intelligences is the human mind itself. There are many theories of how minds work and perhaps one of the better known is Minsky's 'Society of Mind' [13]. Simply stated, this theory argues that minds can be composed of a 'society' of many fundamental competencies. When a mind is stimulated, these competencies compete with one another to be activated as the mind's response. The appeal of such a theory is that the overwhelming complexity of minds can be 'reduced' to the competition and interaction of relatively simple competencies (albeit on a massively parallel scale).

There are, of course, many other natural metaphors that have been adapted into artificial nature-inspired systems. Some, such as Particle Swarm Optimisation [11], are based on further examples of natural swarms such as flocks of birds. Others, such as Simulated Annealing [12], are based on physical systems.

As discussed, the current implementations of these techniques have focussed on centralised, synchronous approaches. The next logical step is to investigate the other characteristics from the natural environment—parallelism, asynchronoicity and decentralisation.

3. NISPADE Challenges Ahead

We identify the following engineering and scientific challenges for the NISPADE field.

3.1. Engineering

A key step for mainstream acceptance of nature-inspired techniques is a maturing of their software engineering practices. As the development of nature-inspired techniques matures it is important to capture successful design and engineering practices to manage their complexity.

3.1.1. Design and Programming Abstractions

The investigation of suitable design and programming abstractions can help to manage the complexity of parallel, asynchronous, and decentralised nature-inspired techniques. Techniques such as software architectures, component frameworks, programming abstractions and design patterns simplify the development of nature-inspired systems. They provide the programmer with the mental infrastructure needed to manage the complexity large-scale parallel, asynchronous and decentralised computing environments. Appropriate design abstractions arm developers with the necessary mental building blocks to embrace the concepts of parallel, asynchronous, and decentralised systems in a straightforward manner. A number of challenging design considerations merit investigation. These include the choice of where to place the burden of complexity and computation; should environments be richer to reduce the load on agents [15], [17], or should the agent contain the complexity?

3.1.2. Tool and Infrastructure Support

The development of tool support and infrastructure for nature inspired systems is an important step. Development tools such as IDE extensions and debugging tools are important to improve the productivity of developers. These tools will encourage the acceptance of nature-inspired techniques.

High-quality support infrastructure such as middleware, software architectures, and software platforms provide common plumbing for systems, allowing developers to concentrate on application logic. Support infrastructure can perform a number of tasks such as distribution, consistency, persistence and mobility. Brueckner, for example, presents an infrastructure to support ant-based pheromone activities [5]. Furthermore, the integration of nature-inspired system with legacy systems already operating within the environment will require the development of integration services and brokers.

3.1.3. Standards

Another important step is the standardisation of relevant protocols and ontologies to facilitate interaction between NISPADE systems and other legacy systems currently operating within these environments. In order for any standardization effort to succeed, there is a need for consensus within the community. Efforts such as FIPA[3] illustrate the benefit of developing standards in an open collaborative environment. When developing such standards it is important to reach a balanced a mixed between industrial and academic interests. Community developed standards benefit from risk reduction afforded by an internationally coordinated and managed effort, reducing risk for individual research efforts by creating useable standards that are implemented and supported by others.

3.1.4. Self-organisation

The challenge of how to engineer self-organising behaviour is still an open issue. We are fortunate to have the benefit of biological models of self-organisation and successful nature-inspired heuristics based on these models. However, their transfer to a parallel, asynchronous and decentralised environment is a new endeavour and we cannot assume that this can be achieved without some modifications and further insights.

3.1.5. Agent Redundancy

The use of large numbers of participants intuitively enhances the robustness of a nature-inspired system to loss of members. However, much experimentation is needed to measurably demonstrate that this is the case. Furthermore, the impact of large numbers on system coordination is not known. Typical ant algorithms, for example, use a number of ants in the order of tens rather than the tens of thousands seen in natural colonies. Is there a cut-off point between sufficient numbers for robustness and sustainable system performance?

3.1.6. Messaging costs

Even with experiments that do address some of the aforementioned issues, it is frequently the case that research ignores the engineering reality of considerations such as messaging costs. While simulations demonstrate useful results, we must not lose sight of the NISPADE field's aim—developing software for a parallel, asynchronous and decentralised environment. At the very least, simulated results that fail to incorporate considerations such as messaging should acknowledge this omission. Ideally, realistic estimates of messaging costs should be included in simulations or results should be tested in realistic environments. Some nature-inspired systems rely on large numbers of simple communications. We believe that balancing the complexity and frequency of communication is one of the most important considerations.

3.2. Science

As with any scientific or engineering discipline, reproducibility and the strength of conclusions are of prime importance. Ultimately, progress of any value will depend on the use of rigorous experiment designs, clear reporting and well-defined hypotheses sup-

[3] http://www.fipa.org/

ported by appropriate statistical tools. Unfortunately, such rigour is currently the exception rather than the rule. While performance analysis engineers and heuristics designers have been aware of these issues for quite some time [10] [8,9], the nature-inspired systems community has only recently acknowledged them [1] [2]. These issues are also extremely relevant to NISPADE practitioners.

3.2.1. Benchmarks

A representative class of benchmark problem instances is vital to consolidate research and to facilitate the comparison of various research approaches. What are appropriate benchmarks for the performance of NISPADE systems? How to we experimentally control important factors in dynamic environments? It would greatly help progress in the field if authors made available their problem instances and problem generators.

3.2.2. Parameter Tuning

In common with their nature-inspired heuristic heritage, NISPADE systems often use a large number of tuning parameters. Unfortunately, these parameters are usually tuned in an ad-hoc way with authors reporting that after some experimentation, particular parameter values were found to be satisfactory. This poses several problems.

Firstly, we have no idea of the human and machine resources used to perform this tuning. Such resources are clearly an important factor for the practicality of the proposed method when competing with simpler alternatives. Such a simpler alternative could have been run many times when ad-hoc parameter tuning of the NIPSADE system was taking place.

Secondly, by not performing tuning in a clearly defined and methodical way, we may be misrepresenting the potential of the proposed NISPADE system. Even small changes in tuning parameters can result in dramatic improvements in performance. There are well-established experiment designs and statistical tools for performing such tuning [1,16,14].

Finally, in the absence of a clearly defined and reported tuning procedure, any conclusions about the proposed system *must* be limited to the particular parameter values reported. Clearly, addressing the parameter tuning problem with scientific rigour will increase the impact and usefulness of NISPADE research.

3.2.3. Design of Experiments

The field of Design of Experiments (DOE) [14] provides established experiment designs and statistical analysis tools that allow us to efficiently gather data and draw conclusions with mathematical preciseness. DOE helps us answer questions such as: which factors to vary and what levels to vary them at; how many replicates of an experimental treatment to run; how confident can we be in our conclusions; is one system better than another or is the difference just due to chance? We see an excellent opportunity to improve the experimental rigour in both the design and analysis of NISPADE experiments. We would hope to see greater attention paid to the existing DOE literature and software.

4. Concluding Remarks

Nature-inspired heuristics have already achieved noteworthy successes. The genetic algorithm has demonstrated the usefulness of a population-based search with randomisation from artificial implementations of cross-over and mutation. Ant Colony Optimisation has profited from the self-reinforcing trail-laying behaviour of real ant colonies. Clustering algorithms have been developed from the brood sorting behaviours of real ants. Particle Swarm algorithms have adapted flocking and shoaling behaviour to search through problem spaces.

This article has highlighted the common characteristics shared between emerging computing environments and natural environments. We believe that further characteristics of natural environments are worth investigating, either because these characteristics are desirable or because similar characteristics are forced on us by new computing environments.

In certain circumstances, parallel execution and asynchronous communication can improve performance. In other circumstances, these characteristics are a requirement of the operating environment. As computing systems increase in size, the feasibility of a centralised design approach diminishes. Decentralisation is a necessity to overcome bottlenecks in communication and obstacles to scalability. It is possible that existing nature-inspired systems may benefit from parallel implementations. Furthermore, their self-organising characteristics may help address the problem of coordinating decentralised execution.

We have highlighted our view of the most important engineering and scientific challenges that lie ahead. We look forward to continued advances in this new and challenging field of NISPADE research.

References

[1] Workshop on empirical methods for the analysis of algorithms. In *International Conference on Parallel Problem Solving From Nature*, Reykjavik, Iceland.

[2] T. Bartz-Beielstein and M. Preuß. Experimental research in evolutionary computation. Tutorial at the genetic and evolutionary computation conference, June 2005.

[3] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence*. Santa Fe Institute Studies in the Sciences of Complexity. Oxford University Press, 1999.

[4] Eric Bonabeau and Christopher Meyer. Swarm intelligence: A whole new way to think about business. Technical report, Harvard Business School Publishing Corporation, May 2001.

[5] Sven Brueckner. *Return from the Ant: Synthetic Ecosystems for Manufacturing Control*. Phd, Humboldt University.

[6] Marco Dorigo and Alberto Colorni. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B*, 26(1):1–13, 1996.

[7] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press, Cambridge, Massachusetts, USA, 1995.

[8] John N. Hooker. Needed: An Empirical Science of Algorithms. *Operations Research*, 42(2):201–212, 1994.

[9] John N. Hooker. Testing heuristics: We have it all wrong. *Journal of Heuristics*, 1:33–42, 1996.

[10] Raj Jain. *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation and modeling*. John Wiley and Sons Inc., 1991.

[11] J. Kennedy, R. C. Eberhart, and Y. Shi. *Swarm Intelligence*. Morgan Kaufmann, 1 edition, 2001.

[12] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

[13] Marvin Lee Minsky. *The Society of Mind*. Simon and Schuster, New York, USA, 1 edition, 1985.

[14] Douglas C. Montgomery. *Design and Analysis of Experiments*. John Wiley and Sons Inc, 6 edition, 2005.

[15] James J. Odell, H. Van Dyke Parunak, Mitch Fleischer, and Sven A. Brueckner. Modeling Agents and Their Environment: the Physical Environment. *Journal of Object Technology*, 2(2), 2003.

[16] Enda Ridge and Daniel Kudenko. Sequential Experiment Designs for Screening and Tuning Parameters of Stochastic Heuristics. In Luıs Paquete, Marco Chiarandini, and Dario Basso, editors, *Workshop on Empirical Methods for the Analysis of Algorithms at the Ninth International Conference on Parallel Problem Solving from Nature (PPSN)*, pages 27–34. 2006.

[17] Danny Weyns, H. Van Dyke Parunak, Fabien Michel, Tom Holvoet, and Jacques Ferber. Environments for Multiagent Systems: State-of-the-art and Research Challenges. In *Post-proceedings of the First International Workshop on Environments for Multiagent Systems*, Lecture Notes in Computer Science. Springer-Verlag, 2005.

Behaviour	**Nature-inspired technique**
Path formation by ants	Network routing and discrete combinatorial optimisation.
Nest sorting by ants	Data clustering
Task allocation by several social insect species	Task assignment in computer and human domains
Schooling and flocking in fish and birds	Particle swarm optimisation.
Darwinian selection of successful organisms	evolutionary computation for search and optimisation

Table 1. Example natural metaphors and their corresponding nature-inspired techniques

Biographies

Enda Ridge

Enda Ridge is currently a researcher at the University of York, England. His main research interests are algorithms for search and optimisation, particularly heuristic (approximate) algorithms, and Design of Experiments techniques for evaluating search and optimisation algorithms.

Edward Curry

Edward Curry served as a research fellow in the Information Technology Department at the National University of Ireland, Galway. His main research interests include self-* and nature-inspired middleware. Curry has a PhD in computer science from the National University of Ireland, Galway. He is a member of the IEEE and the ACM.