system verification. Furthermore, learning and adaptation with the goal of continual improvements in performance will mean that emergent behavior patterns simply cannot be fully predicted through the use of traditional system development methods.

The Formal Approaches to Swarm Technology (FAST) project aims at devising a formal approach to the development and verification of complex swarm-based systems, using ANTS as a baseline for comparing approaches.

An effective formal method for use in the ANTS mission must be able to predict the emergent behavior of 1000 agents operating as a swarm, as well as the behavior of the individual agents. Crucial to the success of the mission will be autonomic properties and the ability to modify operations autonomously to reflect the changing nature of the mission. For this, the formal specification will need to be able to track the goals of the mission as they change and to modify the model of the universe as new data comes in. The formal specification will also need to allow for specification of the decision-making process to aid in the decision of which instruments will be needed, at what location, with what goals, etc.

The project is currently working on integrating existing formal techniques and on building tools to support the integrated method.

Links:
ANTS website: http://ants.gsfc.nasa.gov
NASA Software Engineering Laboratory: http://sel.gsfc.nasa.gov
UU Autonomic Systems: http://www.infj.ulst.ac.uk/~autonomic

Please contact:
Dr. Mike Hinchey, Director, NASA Software Engineering Laboratory, USA
Tel: +1 301 286 9057
E-mail: Michael.G.Hinchey@nasa.gov

Collective Intelligence and Evolution

by Akira Namatame

The mission of collective evolution is to harness the systems of selfish agents to secure a sustainable relationship, so that desirable properties can emerge as 'collective intelligence'.

Why do colonies of ants work collectively, and how do they do it so effectively? One key to answering this question is to look at interactions among ants. For the last decade, attempts have been made to develop some general understanding, which has produced the theory of collective systems, that is, systems consisting of a large collection of agents. It is common to refer to the desirable emergent properties of collective systems as 'collective intelligence'. Interactions are able to produce collective intelligence at the macroscopic level that is simply not present when the components are considered individually.

The concept of collective intelligence observed in social insects can be extended to humans. In his book, The Wisdom Of Crowds, Surowiecki explores a simple idea that has profound implications: a large collection of people are smarter than an elite few at solving problems, fostering innovation, coming to wise decisions, and predicting the future. His counterintuitive notion, rather than crowd psychology as traditionally understood, provides us with new insights for understanding how our social and economic activities should be organized.

On the other hand, the fact that selfish behaviour may not achieve full efficiency is also well known in the literature. It is important to investigate the loss of collective welfare due to selfish and uncoordinated behavior. Recent research efforts have focused on quantifying this loss for specific environments, and the resulting degree of efficiency loss is known as 'the price of anarchy'. Investigations into the price anarchy have provided some measures for designing collective systems with robustness against selfish behaviour. Collective systems are based on an analogous assumption that individuals are selfish optimizers, and we need methodologies so that the selfish behaviour of individuals need not degrade the system performance. Of particular interest is the issue of how social interactions should be restructured so that agents are free to choose their own actions, while avoiding outcomes that none would choose.

Darwinian dynamics based on mutation and selection form the core of models for evolution in nature. Evolution through natural selection is often understood to imply improvement and progress. If multiple populations of species are adapting each other, the result is a co-evolutionary process. However, the problem to contend with in Darwinian co-evolution is the possibility of an escalating arms race with no end. Competing species may continually adapt to each other in more and more specialized ways, never stabilizing at a desirable outcome.

The Rock-Scissors-Paper (RSP) game is a typical form of representing the triangular relationship. This simple game has been used to explain the importance of biodiversity. We generalize a basic rock-scissors-paper relationship to a non-zero-sum game with the payoff matrix shown in Table 1. In this triangular situation, diversity resulting from proper dispersal by achieving Nash equilibrium is not efficient, and the agents may benefit from achieving a better relationship.

In particular, we have examined the system of interactive evolving agents in the context of repeated RSP games, by con-

Opponent Choice / Own choice	S_1 (0) (Rock)	S_2 (1) (Scissors)	S_3 (2) (Paper)
S_1 (0) (Rock)	1 / 1	0 / λ	λ / 0
S_2 (1) (Scissors)	λ / 0	1 / 1	0 / λ
S_3 (2) (Paper)	0 / λ	λ / 0	1 / 1

Table 1: The generalized rock-scissors-paper game ($\lambda \leq 2$).

sidering a population of agents located on a lattice network of 20x20. They repeatedly play the generalized RSP game with their nearest eight neighbours based on the coupling rules, which are updated by the crossover operator. 400 different rules, one for each agent, are aggregated at the beginning into a few rules with many commonalities. The game between two agents with the learned coupling rule becomes a kind of stochastic process. The transitions of the outcome are represented as the phase diagram in Figure 1, and they converge into the limit cycle, visiting the Pareto-optimal outcomes: (0,1) (1,2) (2,0) (1,0) (2,1) (0,2). Therefore each agent learns to behave as follows: win three times and then lose three times. In this way, the agents succeed in collectively evolving a robust learning procedure that leads to near-optimal behaviour based on the principle of give and take.

The framework of collective evolution is distinguished from co-evolution in three aspects. First, there is the coupling rule: a deterministic process that links past outcomes with future behaviour. The second aspect, which is distinguished from individual learning, is that agents may wish to optimize the outcome of the joint actions. The third aspect is to describe how a coupling rule should be improved, using the criterion of performance to evaluate the rule.

In biology, the gene is the unit of selection. However, the collective evolutionary process is expected to compel agents towards ever more refined adaptation, resulting in sophisticated behavioural rules. Cultural interpretations of collective evolution assume that successful behavioural rules are spread by imitation or

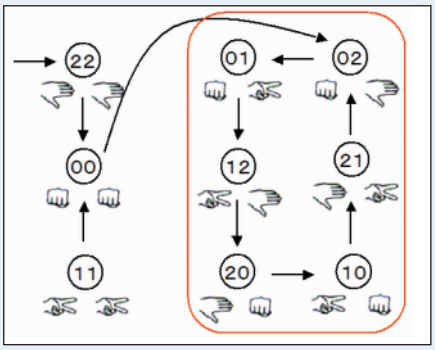

Figure 1: The state diagram of the strategy choices between two agents.

learning by the agents. This approach to collective evolution is very much at the forefront of the design of desired collectives in terms of efficiency, equity, and sustainability. Further work will need to examine how collective evolution across the complex socio-economical networks leads to emergent effects at higher levels.

Please contact:
Akira Namatame,
National Defense Academy, Japan
Tel: +81 468 3810
E-mail: nama@nda.ac.jp
http://www.nda.ac.jp/~nama

Evolving Game-Playing Strategies with Genetic Programming

by Moshe Sipper

We have recently used genetic programming, wherein computer programs evolve by artificial selection, to obtain human-competitive game-playing strategies for three games: chess, backgammon, and Robocode.

The idea of applying the biological principle of natural evolution to artificial systems, introduced over four decades ago, has seen impressive growth in the past few years. Evolutionary algorithms have been successfully applied to numerous problems from different domains, including optimization, automatic programming, circuit design, machine learning, economics, immune systems, ecology, and population genetics, to mention but a few. Our group focuses on the evolutionary methodology known as genetic programming.

In genetic programming we evolve a population of computer programs, whose basic building blocks are designed for the problem at hand. For example, when we evolved backgammon-playing programs the list of elements from which programs could be constructed included:

- Player-Exposed(n): Test whether the player has exactly one checker at board location n
- Player-Blocked(n): Test whether the player has two or more checkers at location n
- Enemy-Exposed(n): Test whether the enemy has exactly one checker at board location n
- Sub(F, F) : Subtract two real numbers.

The full list (for backgammon) comprises over 20 such programmatic elements.

The main mechanism behind genetic programming is that of a generic evolutionary algorithm (which, in turn, is inspired by nature), namely, the repeated cycling through four operations applied to the entire population: evaluate, select, crossover, and mutate. Start with an initial population of randomly generated programs composed of the program elements for the problem at hand (eg, backgammon); this random population is known as generation zero. Each individual is then evaluated in the domain envi-