Coevolutionary Computation: An Introduction

Larry Bull

Intelligent Computer Systems Centre
Faculty of Computer Studies & Mathematics
University of the West of England
Bristol BS16 1QY, U.K.
larry@ics.uwe.ac.uk

We might instead consider what happens when we make up [an intelligent] machine in a comparatively unsystematic way from some kind of standard components ... from a rather large number N *of similar units.*

Further research into intelligence of machinery will probably be very greatly concerned with 'searches' ... [t]here is the genetical or evolutionary search by which a combination of genes is looked for, the criterion being survival value.

Alan Turing, *Intelligent Machinery*, 1948.

1 Introduction

Agent technology is synonymous with advanced computer software and a growing body of work exists on the use of adaptive techniques within the agent paradigm (e.g. [Kozierok & Maes 1993]). The distributed and/or multi-faceted nature of complex problem domains, such as network routing or process control, has lead to the extension of agent frameworks to multi-agent systems, both within traditional (Distributed) Artificial Intelligence (e.g.[O'Hare & Jennings 1996]) and Machine Learning (e.g. [Weiss 1997]). Here each node/aspect of a problem is under the control of an agent, with many such agents interacting to generate a global solution. The subject of this book is the use of Evolutionary Computation in such systems, termed *Coevolutionary Computation*.

Evolutionary Computing techniques are search algorithms based on the mechanisms of natural selection and genetics. That is, they apply Darwin's principle of the survival of the fittest [Darwin 1859] among computational structures with the stochastic processes of gene mutation, recombination, etc. Central to all evolutionary computing techniques is the idea of searching a problem space by evolving an initially random population of solutions such that better - "fitter" - solutions are generated over time; the population of candidate solutions is seen to *adapt* to the problem space.

These techniques have been applied to a wide variety of domains such as optimization, design, classification, control, economics, biological modelling, and many others. A full review of evolutionary computation is beyond the scope of this article, but recent introductions can be found in [Koza 1992], [Fogel 1995], [Mitchell 1996] and [Baeck 1996].

New issues arise when evolutionary computation is applied to the multi-agent paradigm. In these systems evolutionary algorithms must adapt to dynamic problem spaces, where changes are caused by the interactions of the agents in the environment of the global system [Kauffman & Johnsen 1991]; agents evolve in environments which are themselves made up of evolving agents. The contributions in this book are concerned with the use and extension of evolutionary computing techniques in multi-agent systems.

2 Multi-Agent Systems

The concept of a "computational agent" has become increasingly important in computer science, representing a new level of abstraction with which to design software [e.g. Shoham 1993]. Whilst an exact definition of what is meant by the term has remained a contentious issue, a number of general characterstics have been determined (based on [Wooldridge & Jennings 1995a]):

- agents are said to be *autonomous*, able to operate without the intervention of any other entity and hence have control over their own actions.

- agents are *reactive* to their environment through their perception of its state.

- agents are *pro-active*, able to exhibit goal-directed behaviour as well as behaving in response to their environment.

With the emergence of Distributed Artificial Intelligence [e.g. Bond & Gasser 1988] agents have also become associated with systems containing many such computational entities:

- agents are *social*, able to interact with other agents (possibly humans) via communication frameworks.

Such definitions usually assume a high-level representation for the agent, capable of symbolic reasoning for example. However, within the context of coevolutionary computation, with its background in function optimization, the entities/agents can be much more low-level; Darwinian multi-agent systems range from multi-population optimization approaches [e.g. Husbands & Mill 1991] to multiple physical robot systems [e.g. Floreano & Nolfi]. Nwana [1996] notes that more recently a strand of agent technology has emerged which studies a much broader range of agent types, the emphasis being simply to achieve a given task. He categorises current research efforts as: *collaborative agents, interface agents, mobile agents, information/Internet agents, reactive agents, hybrid agents,* and *smart agents*. Coevolutionary computation represents a technique by which to generate agents in many of these categories, e.g. information/Internet filtering [Moukas 1996].

Distributed artificial intelligence/multi-agent systems are typically applied in two ways. In the first, the problem domain is itself distributed, e.g. telecommunications routing, and as such the multi-agent paradigm is a natural "fit" since each aspect of the system can be attributed to an agent. In the second, complex tasks are divided into multi-aspect problems to allow for the construction of a solution through the combination of a number of simpler (in respect to the global task), interacting agents, e.g. plant control. As will be seen, examples of both approaches using coevolutionary computation have been presented, e.g. [Sipper 1994] and [Parmee 1996] respectively.

Multi-agent systems have a number of characteristics beyond those (listed above) for agency in general, including:

- the type of agent-agent interactions (cooperative, competitive, or some combination)
- the types of agents involved (ranging from all identical to all different)
- the number of agents involved
- the frequency, duration, importance, and variability of agent-agent interactions
- the sophistication of the agents involved (ranging from reasoning to purely reactive)

The effects of each of these characteristics on the resulting system is the subject of much on-going research. It will be seen that coevolutionary computing systems can also be described in these terms.

The most relevant area of multi-agent systems to coevolutionary computation involves the inclusion of Machine Learning techniques [e.g. Weiss & Sen 1996]. Traditional agent systems (multi and single) are pre-programmed, carrying out their task(s) in the same way until their termination. In systems using machine learning techniques the aim is for the agents to be able to adapt themselves to perform their task(s) increasingly well; agents can improve their behaviour and/or cope with changes in their environment automatically. This is seen as particularly poignant in multi-agent systems since they are typically complex, and so it is often extremely difficult to determine appropriate behaviour for each agent a priori. As noted above, coevolutionary computation is an adaptive approach to multi-agent systems using biologically-inspired mechanisms, as opposed to the cognition-inspired mechanisms of machine learning and artificial intelligence. It will be seen that this difference in focus results in new perspectives on multi-agent systems. The reader is referred to [Weiss 1996] for an overview of the issues arising from the use of machine learning techniques in multi-agent systems.

A full review of the rapidly developing field of software agents is beyond the scope of this article, but recent introductions include [Singh 1994], [Wooldridge & Jennings 1995b], and [Bradshaw 1997].

Therefore it can be seen that coevolutionary computation represents a different, and possibly complementary [e.g. Matos et al. 1998], approach to the use of the multi-agent paradigm.

3 Darwinian Multi-Agent Systems

The use of evolutionary computing techniques in systems containing many interacting agents/entities goes back to the earliest days of experiments in machine intelligence. For example, Barricelli [e.g. 1957] used an abstract ecological model to examine the evolution of complexes of cooperative entities, based in the idea of "symbiogenesis" - the evolution of complexity by the bringing together of previously autonomous entities. Conrad [e.g. 1970] developed a series of increasingly complicated ecosystems to examine evolution and population dynamics. This use of evolutionary computation to study artificial ecologies continues today, see for example [Hogweg 1989], [Ray 1991], [Yaeger 1993], etc. In these models the type of interactions between the evolving agents often emerges over time; whether relationships are cooperative or competitive is not predetermined.

The examples of coevolutionary computing considered in this book consist of systems in which agents' roles are predetermined as being either competitive or cooperative or a mixture of the two, i.e. agents are assigned particular tasks within the global system. A summary of the preceding work is now presented.

3.1 Competition

The evolution of competitive multi-agent systems can be seen as analogous to the evolution of parasites and their hosts, or viruses and immune systems. Indeed both phenomena have been examined using coevolutionary computation, see for example [Kaneko & Ikegami 1992] and [Forrest et al. 1993] respectively. In this class of models the quality of an agent's solution at any given time is *relative* to the solutions of the agents with which it interacts.

One of the most well-known models of competitive coevolution is Axlerod's *Iterated Prisoner's Dilemma* model (IPD). The IPD is a simple two-player game in which the agents evolve via a Genetic Algorithm (GA) [Holland 1975] from within a single population and play against a number of other agents for an evaluation. Hence the agents are searching for effective game strategies. The evolution of a multi-agent system in which the agents inter-breed in this way are termed *homogeneous* systems. In initial work [1984] Axlerod showed that agents could be evolved which beat handcrafted strategies in the game. Subsequent work [1987] allowed each agent to play against all others within the population, showing that strategies of similar complexity to those previously seen would emerge from this coevolutionary version of the system. This can be seen as the evolutionary computing equivalent of Samuel's [1959] pioneering work on the role of competitive machine learning. Since then a large number of investigators have examined the use of coevolutionary computation in the IPD, e.g. see [Fogel 1993] and [Lindgren & Nordahl 1995] for overviews.

After Axlerod, Reynolds [1994] evolved Genetic Programming (GP) [Koza 1992] agents for the game of "tag", where each agent was a simulated three-wheeled robot. Reynolds evaluated each agent with six others from within the population. At the end of the simulations coevolved players were played against the (known) optimal player for the game and it was shown that near-optimal performance could be reached. Angeline and Pollack [1993] showed it was possible to evolve competent GP players for the game of Tic Tac Toe without evaluating each agent against all others in the population. They presented a single-round (binary) tournament scheme for use in such systems, showing that equally good players could be found to those evolved under the more computationally expensive scheme of Axlerod.

Miller [1989] introduced a version of Axlerod's model in which two explicit populations existed, where individuals in one population did not breed with those in the other - a *heterogeneous* coevolutionary system. He used Axlerod's "all vs. all" strategy for evaluations and examined the effects of noise in the game. Sims [1994] presented a single evaluation scheme, using the best individual from the other population, in his work on using a GA to evolve the neural controllers of two 3D virtual creatures that competed in a physically simulated world. The creatures had to evolve both morphology and behaviour to capture an object initially placed between them. Miller and Cliff [1994] presented a similar model to Sim's to evolve unpredictable behaviour in a simulated pursuit-evasion game between two neural network controlled robots. They showed that increasingly complex strategies would emerge; predators from later generations could easily catch prey from earlier generations (see also [Floreano & Nolfi 1997]). Rosin [1997] has applied this general form of coevolutionary system to process control for a well-known bioreactor model and Porto [1998] has used Evolutionary Programming (EP) [Fogel et al. 1966] to evolve the strategies of simulated military vehicles on opposing sides in a well-known tactics model.

The work of Hillis [1991] showed that the performance of evolutionary algorithms can be improved in complex problem domains by casting them as competitive coevolutionary computing systems. That is, using a GA he showed that the general principle of coevolutionary "arms races" could force the agents involved to search their problem spaces more effectively; solutions can only propagate if they are consistently good. Hillis used the well-known Batcher sorting network task to show how, by coevolving a population of test cases with the population of required sorters, solutions could be found which did as well as the best of those designed by hand. Here the test cases' fitness was the inverse of that obtained by the sorter with which they were evaluated, with each sorter tried against a sub-set of the test case population (see [Shapiro 1998] for a recent review).

3.2 Cooperation

Nature is full of examples of cooperation between entities: within species, as in the workings of social insect colonies; and between species, as in the cleaning mutualism seen between the Pederson shrimp and the fish of the Bahamas. Similarly, both homogeneous and heterogeneous cooperative coevolutionary systems have been presented.

Colins and Jefferson [1991] used a GA to evolve neural network ants which lived in colonies of twenty individuals. Each ant in a colony was identical (cloned) and had to learn to cooperate with the others to produce effective foraging behaviour (see also [Koza 1991] for a GP version). Reynolds [1993] used GP in the same way to evolve simulated mobile agents capable of rudimentary group motion. Mitchell et al. [1993] used a GA to examine the computational characteristics of cellular automata, in which the state table for the cells were evolved. This has been extended [e.g. Mitchell et al. 1994] to examine its use in the evolution of cellular automata classifiers. More recently, Bull and Holland [1997] have presented a form of evolutionary algorithm explicitly for these kinds of multi-agent systems, loosely based on the genetic constitution of natural colony members. They show that a limited amount of random genetic diversity between the initially cloned agents can improve performance over the use of strictly identical agents due to a process analogous to random learning.

Sipper [e.g. 1994] has presented an alternative strategy to the evolution of such systems, using Mitchell et al.'s [1994] model. Here each member of the population creates the multi-agent system for one evaluation and hence this can be seen as a cooperative version of Axlerod's model described above. White et al. [1998] have recently applied the same approach to a version of the ant optimization algorithm [e.g. Dorigo et al. 1996] for path finding. Bull and Fogarty [1994a, 1995] also presented a cooperative version of Axlerod's homogeneous system in the evolution of multiple rule-based agents. Here the number of required agents for an evaluation are taken from the single population (without replacement). They evolved cooperative agents for the control of the wheels/legs of a simulated robot, with the coevolving system performing better than the equivalent traditional approach.

Husbands and Mill [1991] were the first to use a heterogeneous cooperative coevolutionary approach for complex problems. They used a GA to optimize the schedule of a manufacturing plant. Populations of processing plans for each sub-component for a given product were coevolved with a population of arbitrators to resolve machine usage conflicts; in effect the arbitrators do the scheduling. Individual plans were ordered according to rank according to their *local* fitness - that of their tooling costs, etc. - and then global system solutions were constructed by partnering individuals of equivalent rank for each sub-component. Individuals also received the *global* fitness before selection. Efficient schedules were shown to be produced on a task previously assumed too difficult to tackle with such techniques. Bull and Fogarty [1993,1994b, 1995] used a similar approach to evolve rule-based agents for a number of simulated robot tasks, where each agent was responsible for one particular aspect of the system, e.g. sensor noise filtering [1994b]. The rule-bases also contained a simple agent-agent communication framework; message passing protocols were occasionally seen to emerge (after [MacLennan 1991][Werner & Dyer 1991]). Paredis [1995] presented a two population approach to the automatic determination of an appropriate variable ordering on a GA string. Here one population represented the possible ordering of the variables and the second their actual values. It was shown that interdependent variables would group together such that they were less likely to be disrupted by recombination. Parmee [1996] used a similar two-population system for an engineering design task, showing improvements over the traditional single GA approach. Significantly, each population used a different representation and type of search operator. Ahluwalia and Bull [e.g. 1997] have presented a coevolutionary approach to the development of hierarchical programs in GP. They showed that by evolving functions in separate populations from the main program, improved performance could be obtained over Koza's original function mechanism [Koza 1992, p534]. They also showed that it is possible to dynamically vary the number of function populations (agent types) during a simulation, based on a given function's recent usage/worth [1998]; the number of coevolving populations emerges to suit the problem.

This survey has given a summary of a number of key works in the development and use of coevolutionary computation. On the whole, to avoid repetition, the work of the contributors to this book has not been covered except where the work varies from that which they present here. The next section of this article briefly reviews the contributions to follow.

4 Coevolutionary Computation: An Overview

This book, in keeping with the themes emerging from the field, is divided into two main sections: competition and cooperation. Within each section the first few contributions describe new techniques which have been presented for the use of evolutionary computation in multi-agent systems. The contributions then move into examples of the application of coevolutionary computation, both to well-known tasks from (Distributed) Artificial Intelligence and to real-world applications.

4.1 Competition

Paredis - Coevolution, Memory and Balance. Paredis has used Hillis' [1991] approach extensively. Here he describes a modification to avoid the solution agents becoming too specialised to the current test case agents, an effect shown in a number of competitive models, e.g. [Miller & Cliff 1994]. Further, he shows that by balancing the *rate* of evolution in the differing populations improved results can be obtained.

Rosin and Belew - On Competitive Coevolution. The way in which partners are chosen from the other agent populations for an evaluation is an extremely significant aspect of coevolutionary computation. Here a number of strategies are introduced and compared, along with a number of methods for allocating fitness to individuals.

Juille & Pollack - Dynamics of Coevolutionary Learning. Juille and Pollack describe the use of a GP approach to solve the well-known classification problem of intertwined spirals. They show that the use of a coevolutionary approach performs better than a traditional approach. They also describe how the coevolutionary system, by constantly forcing the solution agents into different regions of the search space, encourages the formation of modular building blocks in the program trees; the role of crossover is found to be significant.

Haynes and Sen - The Evolution of Multi-Agent Coordination Strategies. This contribution describes the use of GP on the well-known "Serengeti" predator-prey model. In this task a team of predators must catch a prey; the model contains both agent cooperation and competition. They describe the successful evolution of predator strategies using a fixed strategy prey, and that the resulting strategy performs better than the handcrafted ones previously presented. They then describe the evolution of teams with the prey also evolving.

Luke and Spector - Evolving to Cooperate to Compete. This work, also using GP and the Serengeti model, presents a number of strategies for breeding teams and a number of mechanisms to improve agent coordination. They then report the application of their results to the successful evolution of a simulated football team for the 1997 RoboCup. Their system beat two handcrafted systems before being knocked out.

Smith - Classifier Systems for Combat. Classifier systems [Holland et al. 1986] are rule-based blackboard architectures which use reinforcement learning to assign utility to individual rules and evolutionary computing to provide rule discovery. Their use in multi-agent systems is now growing (see [Bull 1998] for an overview). In this contribution the use of classifier systems in the testing of possible fighter aircraft designs is described, both against fixed and evolving opponents. It is shown that often complicated manoeuvres known to pilots emerge.

Tesfatsion - Evolutionary Labour Markets with Adaptive Search and Behaviour. After [Arthur 1990] a number of investigators are examining the use of coevolutionary computation in economic systems, see for example [Chattoe 1994][Palmer et al. 1994]. Indeed, one of the motivations for a lot of the early IPD work [e.g. Miller 1989] was to study economic markets as coevolutionary systems. This contribution describes the use of partner choice and refusal in versions of a game representing a small labour market. It is shown that the type of underlying market has a significant impact on the types of contract networks which emerge.

4.2 Cooperation

Bennet III - Emergence of a Multi-Agent Architecture and New Tactics for the Ant Colony Food Foraging Problem Using Genetic Programming. This work presents a GP-based approach to the evolution of complex multi-agent systems. Significantly, the approach involves the use of a (heterogeneous) multi-agent architecture to control each agent *within* a (homogeneous) multi-agent system. The technique produces two new solutions to the foraging task.

Iba - Evolutionary Learning of Communicating Agents. The use of agent-agent communication, both in terms of message passing during the agents' actions and the sharing of knowledge obtained by agents, is a fundamental aspect of the multi-agent paradigm. This contribution describes the use of agent-agent communication primitives in GP to improve the performance of a team of agents (see also [Juille & Pollack 1997] for a continuous communication neural network mechanism). Iba also examines a speciation process between the different agent populations for the task, finding that speciation between a diverse mix of agents is most beneficial (see also [Bull & Fogarty 1996a, 1996b]); the periodic passing of information between populations is shown to improve performance.

Bull - Genetic Algorithms in Multi-Agent Environments. Just as Rosin and Belew's contribution shows how partnering strategies are significant in competitive domains, this contribution compares a number of strategies presented in the literature for cooperative domains. It is shown that, depending on the type of GA used and how fitness is assigned the performance of a given strategy can vary greatly. The performance of the standard genetic operators - mutation and recombination - are also examined, with recombination found to be less useful in most closely coupled systems.

Seredynski - Coevolutionary Game-Theoretic Multi-Agent Systems. In some multi-agent systems where a cooperative solution is required the individual agents have the ability to degrade the quality of that solution by their actions. Seredynski presents and examines a number of fitness sharing schemes for use in such systems.

Unemi - Evolving Cooperative Robot Teams. The use of evolutionary computing techniques to develop robot controllers is one of the main areas of application in the field [e.g. Floreano & Nolfi]. However, problems can arise due to discrepancies between the simulation and the real robot. This contribution uses a similar approach to Sipper's [1994] to tackle this problem in a multi-agent robot domain. Here, each member of the population is placed in a robot. As the robots execute their task they record their performance. Periodically, the robots ask their immediate neighbours for their fitnesses and then (conditionally) breed with them to produce their own successor on the robot. A number of mate selection strategies are examined in a collective cleaning task.

Carse, Munro and Fogarty - Evolving Temporal Fuzzy Rule-Bases for Distributed Routing Control in Telecommunications Networks. This contribution describes the use of coevolutionary computation to develop fuzzy logic agents for routing. Here a cloning strategy is used to generate an agent for each node of a simulated telecommunications network. It is shown that the approach can give improved performance over well-known handcrafted algorithms.

McIihagga, Husbands and Ives - A Comparison of Optimization Techniques for Integrated Manufacturing and Planning. This contribution provides an up-to-date description of the seminal work presented in [Husbands & Mill 1991]. Since then a number of changes have been made to the original method, including the use of a distributed population model. The latest version of the coevolutionary approach is favourably compared with a number of others for the scheduling task.

Summary

Just over ten years after Axlerod first described the coevolution of agents for a simple game, the use of evolutionary computation in multi-agent/multi-population models is rapidly increasing. This article has given a general introduction to, and an historical overview of, coevolutionary computation. The rest of the book collects together work from a number of individuals who have contributed to the development of such systems to date; the book represents a metaphorical "line in the sand".

At this point a number of challenging issues for future research are emerging, including, but not limited to, a need for:

- more formal models and understanding of coevolutionary algorithms
- improvement in the performance of the search process in such systems, possibly via the use of self-adapting operator parameters (see [Kaneko & Ikegami 1992])
- improved agent-agent communication mechanisms
- the inclusion of other "system-level" processes to exploit the population-based characteristics of evolutionary search techniques, e.g. gene sharing [e.g. Meuleau 1991]
- the application of such systems to a better understanding of analogous natural multi-agent phenomena such as symbiosis, multicellularity, and social insects.
- methods for parallel implementations of coevolutionary computation
- an understanding of the effects of including mixed representations and search strategies, particularly during their application to real-world problems [e.g. Parmee 1996]

As these and other emerging topics are investigated, along with an increased understanding of those areas already identified, the use of coevolutionary computation will also increase; the potential of Darwinian multi-agent systems is now being realised by the wider agent community.

References

Ahluwalia M, Bull L & Fogarty T C (1997), "Coevolving Functions in Genetic Programming: A Comparison in ADF Selection Strategies", J R Koza, K Deb, M Dorigo, D B Fogel, M Garzon, H Iba, R Riolo (eds.) *Proceedings of the Second Annual Conference on Genetic Programming*, Morgan Kaufmann, pp3-8.

Ahluwalia M & Bull L (1998), "Coevolving Functions in Genetic Programming: Dynamic ADF Creation using GLiB", in V W Porto, N Saravanan, D Waagen & A E Eiben (eds.) *Proceedings of the 7th International Conference on Evolutionary Programming*, Springer Verlag, pp809-818.

Angeline P & Pollack J (1993), "Competitive Environments Evolve Better Solutions for Complex Tasks", in S Forrest (ed.) *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann, pp264-270.

Arthur W B (1990), "A Learning Algorithm that Replicates Human Learning", Santa Fe Institute Technical Report 90-026.

Axlerod R (1984)(ed.), *The Evolution of Cooperation*, Basic Books.

Axlerod R (1987), "Evolution of Strategies in the Iterated Prisoner's Dilemma", in L Davis (ed.) *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann, pp32-41.

Baeck T (1996)(ed.), *Evolutionary Algorithms in Theory and Practice*, Oxford University Press.

Barricelli N A (1957), "Symbiogenetic Evolution Processes Realised by Artificial Methods", *Methodos* 9: 35-36.

Bond A H & Gasser L (1988)(eds.), *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann.

Bradshaw J M (1997)(ed.), *Software Agents*, MIT Press.

Bull L (1998), "On ZCS in Multi-Agent Environments", in A E Eiben, T Baeck, I Rechenberg & H-P Schwefel (eds.) *Parallel Problem Solving from Nature - PPSN V*, Springer Verlag, pp471-480.

Bull L & Fogarty T C (1993), "Coevolving Communicating Classifier Systems for Tracking", in R F Albrecht, C R Reeves & N C Steele (eds.) *Artificial Neural Networks and Genetic Algorithms*, Springer Verlag, pp522-527.

Bull L & Fogarty T C (1994a), "Evolving Cooperative Communicating Classifier Systems" in A V Sebald & L J Fogel (eds.) *Proceedings of the Third Annual Conference on Evolutionary Programming*, World Scientific, pp308-315.

Bull L & Fogarty T C (1994b), "Parallel Evolution of Communicating Classifier Systems", in *Proceedings of the 1994 IEEE Conference on Evolutionary Computin*g, IEEE, pp680-685.

Bull L & Fogarty T C (1996a), "Evolutionary Computing in Cooperative Multi-Agent Systems", in S Sen (ed.) *Proceedings of the 1996 AAAI Symposia on Adaptation, Coevolution and Learning in Multi-Agent Systems*, AAAI, pp22-27.

Bull L & Fogarty T C (1996b), "Evolutionary Computing in Multi-Agent Environments: Speciation and Symbiogenesis", in H-M Voigt, W Ebeling, I Rechenberg & H-P Schwefel (eds.) *Parallel Problem Solving from Nature - PPSN IV*, Springer Verlag, pp12-21.

Bull L, Fogarty T C & Snaith M (1995), "Evolution in Multi-Agent Systems: Evolving Communicating Classifier Systems for Gait in a Quadrupedal Robot", in L J Eshelman (ed.) *Proceedings of the Sixth International Conference on Genetic Algorithms*, Morgan-Kaufmann, pp382-388.

Bull L & Holland O (1997), "Evolutionary Computing in Multi-Agent Environments: Eusociality", in J R Koza, K Deb, M Dorigo, D B Fogel, M Garzon, H Iba, R Riolo (eds.) *Proceedings of the Second Annual Conference on Genetic Programming*, Morgan Kaufmann, pp347-352.

Chattoe E (1994), "The use of Evolutionary Algorithms in Economics: Metaphors or Models of Social Interaction ?", in E Hillebrand & J Stender (eds.) *Multi-Agent Simulation and Artificial Life*, IOS, pp48-83

Collins & Jefferson (1991), "AntFarm: Towards Simulated Evolution", in C G Langton, C Taylor, J D Farmer & S Rasmussen (eds.) *Artificial Life II*, Addison-Wesley, pp579-602.

Conrad M & Pattee H H (1970), "Evolution Experiments with an Artificial Ecosystem", *Journal of Theoretical Biology* 28:393-409.

Darwin C R (1859)(ed.), *The Origin of Species*, John Murray.

Dorigo M, Maniezzo V & Colorni A (1996), "The Ant System: Optimization by a Colony of Cooperating Agents", *IEEE Transactions on Systems, Man, and Cybernetics* 26 (1): 29-41.

Floreano D & Nolfi S (1997), "Adaptive Behaviour in Competing Coevolving Species", in P Husbands & I Harvey (eds.) *Proceedings of the 4th European Conference on Artificial LIfe*, MIT Press, pp378-387.

Fogel D B (1993), "Evolving Behaviours in the Iterated Prisoner's Dilemma", *Evolutionary Computation* 1(1):77-97.

Fogel D B (1995)(ed.) *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE.

Fogel L J, Owens A J & Walsh M J (1966)(eds.) *Artificial Intelligence Through Simulated Evolution*, Wiley.

Forrest S, Javornik B, Smith R & Perelson A (1993), "Using Genetic Algorithms to Explore Pattern Recognition in the Immune System", *Evolutionary Computation* 1(1):191-211.

Hogweg P (1987), "Mirror Beyond Mirror: Puddles of Life", in C G Langton (ed.) *Artificial Life*, Addison-Wesley, pp297-316.

Holland J H (1975)(ed.) *Adaptation in Natural and Artificial Systems*, University of Michigan Press.

Holland J H, Holyoak K J, Nisbett R E & Thagard P (1986)(eds.), *Induction: Processes of Inference, Learning and Discovery*, MIT Press.

Husbands P & Mill F (1991), "Simulated Coevolution as the Mechanism for Emergent Planning and Scheduling", in R K Belew & L B Booker (eds.) *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, pp264-270.

Kaneko K & Ikegami T (1992), "Homeochaos: dynamics stability of a symbiotic network with population dynamics and evolving mutation rates", *Physica D* 56:406-429.

Kauffman S A & Johnsen S (1991), "Coevolution to the Edge of Chaos: Coupled Fitness Landscapes, Poised States and Coevolutionary Avalanches", in C G Langton, C Taylor, J D Farmer & S Rasmussen (eds.) *Artificial Life II*, Addison-Wesley, pp325-370.

Koza J R (1992)(ed.), *Genetic Programming*, MIT Press.

Kozierok R & Maes P (1993), "A Learning Interface Agent for Scheduling Meetings", in *Proceedings of ACM SIGCHI International Workshop on Intelligent User Interfaces*, ACM Press, pp81-88.

Lindgren K & Nordahl M G (1995), "Cooperation and Community Structure in Artificial Ecosystems", *Artificial Life* 1(1):15-37.

MacLennan B (1991), "Synthetic Ethology: An Approach to the Study of Communication" in C G Langton, C Taylor, J D Farmer & S Rasmussen (eds.) *Artificial Life II*, Addison-Wesley, pp631-658.

Matos N, Sierra C & Jennings N r (1998), "Determining Successful Negotiation Strategies: an evolutionary approach", in *Proceedings of the Third International Conference on Multi-Agent Systems*, AAAI, pp182-189.

Meuleau N (1992), "Simulating Coevolution with Mimetism" in F J Varela & P Bourgine (eds.) *Proceedings of the 1st European Conference on Artificial LIfe*, MIT Press, pp179-184.

Miller J H (1989), "The Coevolution of Automata in the Repeated Prisnoer's Dilemma Game", Santa Fe Institute Technical Report 89-003.

Miller G F & Cliff D (1994), "Protean Behaviour in Dynamic Games: Arguments for the Coevolution of Pursuit-Evasion Tactics", in D Cliff, P Husbands, J-A Meyer & S W Wilson (eds.) *From Animals to Animats 3*, MIT Press, pp411-420.

Mitchell M, Hraber P, & Crutchfield J (1993), "Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations", *Complex Systems* 7:89-130..

Mitchell M, Crutchfield J & Hraber P (1994), "Evolving Cellular Automata to Perform Computations: Mechanisms and Impediments", *Physica D* 75: 361-391.

Mitchell M (1996)(ed.), *An Introduction to Genetic Algorithms*, MIT Press.

Moukas A (1996), "Amalthea: Information Discovery and Filtering using a Multi-Agent Evolving Ecosystem", in *Proceedings of the First International Conference and Exhibition on the Practical Applications of Intelligent Agents and Multi-Agent Technology*, The Practical Application Company Ltd, pp421-436.

Nwana H (1996), "Software Agents: an overview", *Knowledge Engineering Review* 11(3):205-244.

O'Hare G & Jennings N (1996)(eds.), *Foundations of Distributed Artificial Intelligence*, Wiley-Interscience.

Palmer R, Arthur W B, Holland J H, LeBaron B & Tayler P (1994), "Artificial Economic Life: A Simple Model of a Stockmarket", *Physica D* 75:264-274.

Paredis J (1995), "The Symbiotic Evolution of Solutions and their Representations", in L J Eshelman (ed.) *Proceedings of the Sixth International Conference on Genetic Algorithms*, Morgan Kaufmann, pp359-365.

Parmee I (1996), "The Development of a Dual-Agent Strategy for Efficient Search Across Whole System Engineering Design Hierarchies", in H-M Voigt, W Ebeling, I Rechenberg & H-P Schwefel (eds.) *Parallel Problem Solving from Nature - PPSN IV*, Springer Verlag, pp523-532.

Porto V W (1998), "Evolving Integrated Low-Level Behaviours into Intelligently Interactive Simulated Forces", in V W Porto, N Saravanan, D Waagen & A E Eiben (eds.) *Proceedings of the 7th International Conference on Evolutionary Programming*, Springer Verlag, pp777-788.

Ray T S (1989), "An Approach to the Synthesis of Life", in C G Langton, C Taylor, J D Farmer & S Rasmussen (eds.) *Artificial Life II*, Addison-Wesley, pp371-408.

Reynolds C W (1993), "An Evolved, Vision-based Behavioural Model of Coordinated Group Motion", in J-A Meyer, H L Roitblat & S W Wilson (eds.) *From Animals to Animats 2*, MIT Press, pp384-392.

Reynolds C W (1995), "Competition, Coevolution and the Game of Tag", in R Brooks & P Maes (eds.) *Artificial Life IV*, MIT Press, pp59-69.

Rosin C D (1997), "Coevolutionary Search Among Adversaries", PhD Thesis, University of California San Diego.

Samuel A (1959), "Some Studies in Machine Learning Using the Game of Checkers", *IBM Journal of Research and Development* 3:210-229.

Shapiro J L (1998), "Does Data-Model Coevolution Improve Generalization Performance of Evolving Learners ?", in A E Eiben, T Baeck, I Rechenberg & H-P Schwefel (eds.) *Parallel Problem Solving from Nature - PPSN V*, Springer Verlag, pp540-549.

Shoham Y (1993), "Agent-Orientated Programming", *Artificial Intelligence* 60(1):51-92.

Sims K (1994), "Evolving 3D Morphology and Behaviour by Competition", in R Brooks & P Maes (eds.) *Artificial Life IV*, MIT Press, pp28-39.

Singh M P (1994)(ed.) *Multiagent Systems: A Theoretical Framework for Intentions, Know How, and Communication*, Springer Verlag.

Sipper M (1994), "Non-uniform Cellular Automata: Evolution in Rule Space and Formation of Complex Structures", in R Brooks & P Maes (eds.) *Artificial Life IV*, MIT Press, pp394-399.

Turing A M (1948), "Intelligent Machinery", from D C Ince (1992)(ed.) *Collected Works of A M Turing*, North Holland, pp107-128.

Weiss G (1996), "Adaptation and Learning in Multi-Agent Systems: Some Remarks and a Bibliography", in G Weiss & S Sen (eds.) *Adaptation and Learning in Multi-Agent Systems*, Springer Verlag, pp1-21.

Weiss G (1997)(ed.), *Distributed Artificial Intelligence Meets Machine Learning*, Springer Verlag.

Weiss G & Sen S (1996)(eds.) *Adaptation and Learning in Multi-Agent Systems*, Springer Verlag.

Werner G M & Dyer M G (1991), "Evolution of Communication in Artificial Organisms", in C G Langton, C Taylor, J D Farmer & S Rasmussen (eds.) *Artificial Life II*, Addison-Wesley, pp659-688.

White T, Pagurek B & Oppacher F (1998), "ASGA: Improving the Ant System by Integration with Genetic Algorithms" in J R Koza, W Banzhaf, K Chellapilla, K Deb, D B Fogel, M Garzon, D E Goldberg, H Iba & R L Riolo (eds.) *Proceedings of the Third Annual Genetic Programming Conference*, Morgan Kaufmann, pp610-616.

Wooldridge M & Jennings N R (1995a), "Intelligent Agents: Theory and Practice", *The Knowledge Engineering Review* 10(2):115-152.

Wooldridge M & Jennings N R (1995b)(eds.) *Agent Technology*, Springer Verlag.

Yaeger L (1993), "Computational Genetics, Physiology, Metabolism, Neural Systems, Learning, and Behavour or Polyworld: Life in a New Context", in C Langton (ed.) *Artificial Life III*, Addison-Wesley, pp263-298.