Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

SEN

Software Engineering

Software ENgineering

Co-evolving automata negotiate with a variety of opponents

D.D.B. van Bragt, J.A. La Poutré

REPORT SEN-R0218 NOVEMBER 30, 2002

CWI is the National Research Institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organization for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-369X

Co-evolving Automata Negotiate with a Variety of Opponents

D.D.B. van Bragt J.A. La Poutré[†]
David.van.Bragt@cwi.nl Han.La.Poutre@cwi.nl
[†] *Also with the School of Technology Management, Eindhoven University of Technology*
De Lismortel 2, 5600 MB Eindhoven, The Netherlands

CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT

Real-life negotiations typically involve multiple parties with different preferences for the different issues and bargaining strategies which change over time. Such a dynamic environment (with imperfect information) is addressed in this paper with a multi-population evolutionary algorithm (EA). Each population represents an evolving collection of bargaining strategies in our setup.

The bargaining strategies are represented by a special kind of finite automata, which require only two transitions per state. We show that such automata (with a limited complexity) are a suitable choice in a computational setting. We furthermore describe an EA which generates highly-efficient bargaining automata in the course of time.

A series of computational experiments shows that co-evolving automata are able to discriminate successfully between different opponents, although they receive no explicit information about the identity or preferences of their opponents. These results are important for the further development of evolving automata for real-life (agent system) applications.

2000 Mathematics Subject Classification: 68T05, 68T20, 91A05, 91A10, 91A18, 91A20, 91A22, 91A26
1998 ACM Computing Classification System: G.1.6, I.2.6, I.2.8
Keywords and Phrases: co-evolution, finite automata, automated negotiations, evolutionary algorithms, evolutionary intelligent agents
Note: Work carried out under theme SEN4 "Evolutionary Systems and Applied Algorithmics". An earlier version of this paper has been published in the proceedings of the 11^{th} Belgian-Dutch Conference on Machine Learning (BENELEARN'2001) (Antwerp, Belgium, December 21, 2001, pp. 77-84) and the proceedings of the 2002 IEEE Congress on Evolutionary Computation (CEC'2002) (Honolulu, HI, USA, May 12-17, 2002, Vol. 2, pp. 1426-1431). This research has been performed within the framework of the project "Autonomous Systems of Trade Agents in E-Commerce", which is funded by the Telematics Institute in the Netherlands.

1. Introduction

The rapid growth of a global electronic market place, together with the establishment of standard negotiation protocols, currently leads to the development of multi-agent architectures in which artificial agents can negotiate on behalf of their users [7, 14]. Such bargaining agents should be able to deal successfully with a variety of opponents (with different tactics and different preferences). Furthermore, these agents should be able to *adapt* their strategies over time to deal with changing circumstances.

Such a dynamic environment (with imperfect information) is addressed in this paper with a multi-population evolutionary algorithm (EA). Each population represents an evolving collection of bargaining strategies. This setup can be considered as an abstract model of multiple adaptive agents who are updating their strategies over time. Our research is thus at the intersection of the fields of multi-agent systems, learning by co-evolution, and evolutionary computing.

We represent the bargaining strategies by a special kind of finite automata, which require only two transitions per state. Such automata (with a limited complexity) are a suitable representation in a

computational setting. Computational experiments indeed show that our modelling choice for the transition function makes it possible for an EA to rapidly find highly-fit solutions. In particular, we show in this paper that our approach (automata with a limited complexity which are optimized by an EA) yields very efficient bargaining strategies for complex bargaining problems involving multiple issues and multiple opponents (with different preferences). Results obtained with the multi-population EA are in fact close to optimal (from a utilitarian point of view), even though the automata receive no explicit information about the identity or preferences of their opponents.

The adaptation of finite automata by means of evolutionary computing has received considerable attention in the past in the field of evolutionary programming (EP) [9, 10, 11]. Recently, several researchers outside the EP community have also started to use evolving automata to solve various tasks related to strategic decision-making and prediction. A well-known example is Miller's work [15] on the iterated prisoner's dilemma (IPD). Miller encoded the game-playing automata as binary strings and adapted them using a genetic algorithm (GA). In 1997, Ashlock [1] introduced a new type of genetic programming (GP), which he calls "GP-automata". This technique is a combination of finite automata with the tree representation of programs which is commonly used in GP. More precisely, a "GP-automaton" is a modified finite automaton with a parse tree associated with each state. These automata were used in [1, 2] to evolve strategies for a very simple bargaining problem ("dividing the dollar"). The suitability of the GP-automata approach for more difficult bargaining problems has not been established yet.

Carmel and Markovitch [5, 6] have proposed a model-based approach for learning effective strategies in multi-agent systems. They restrict the agents' strategies to deterministic finite automata and show that a best response strategy for a given opponent (with a known strategy) can be derived efficiently. They furthermore present an unsupervised learning algorithm that infers a model of the opponent's automaton from its input/output behavior. These techniques were applied successfully to the IPD game. Although the framework of Carmen and Markovitch appears to be promising in case of fixed opponents, the case of non-stationary opponents, which is studied in this paper, is not covered yet by their methods. A second drawback of their approach is that explicit information about the identity of the opponents should be available in a setting with multiple opponents (to facilitate the development of separate finite automaton models for the different opponents). Information about the identity (or preferences) of the opponents is not used (or needed) in our evolutionary model: the only feedback used in our model is the average score obtained by the automata against the different opponents.

Our application domain is rather complex compared to the simple games considered in previous works (e.g., the IPD [15, 22, 5, 6] or dividing the dollar [1, 2]). In this paper, we focus on so-called *multi-issue* negotiations. In multi-issue negotiations not just one issue (like the price of a product) is important, but other aspects are also taken into account (for instance accessories, quality, delivery time, etc.). A key advantage of these multi-issue negotiations is that often mutually beneficial outcomes can be obtained if both parties concede on the proper issues. The complexity of the bargaining problem increases rapidly, however, if the number of issues becomes large. This explains the need for advanced search techniques (like EAs) to generate successful bargaining strategies.

We show in this paper how successful strategies (represented as finite automata) can be generated for this class of complex bargaining problems. Computational experiments show that the co-evolving automata are able to discriminate successfully between different opponents, although they receive no explicit information about the identity or preferences of their opponents. Moreover, performance of the evolving automata remains high in case the bargaining problem is (i) stochastic (e.g., because of stochastic breakdown of the bargaining process or because of noisy communication between the negotiating parties), (ii) asymmetric (e.g., because the players' preferences for the valuable items are not the same), or (iii) time-dependent (e.g., because the players' preferences are changing over time). Our evolutionary approach also works well in case of highly complex multi-issue/multi-opponent bargaining problems. For example, we generated very efficient automata for a problem with 20 co-evolving populations, where each automaton negotiates with automata from 10 different populations over 5 issues. The evolutionary techniques developed in this paper are thus important for the further

development of evolving automata for real-life (agent system) applications.

The remainder of this paper is organized as follows. Section 2 contains a description of the multi-issue bargaining problem that we consider in this paper. Section 3 discusses the generic structure of the bargaining automata. The genetic representation of these automata is presented in Section 4. Section 5 gives an outline of the evolutionary system. The EA, which evolves the automata, is also described in this section. Section 6 gives an overview of the computational experiments. These experiments are presented and discussed in Sections 7 through 10. Section 11 concludes.

2. The multi-issue bargaining problem

Our bargaining game is essentially a discrete-choice, multi-round, and multi-issue version of Nash's demand game [17, 23, 4]. In Nash's game, the two players simultaneously announce a demand. Their demands x_1 and x_2 are either compatible or incompatible. They are compatible if the pair (x_1, x_2) lies in the set X of feasible payoff pairs. Otherwise they are incompatible. In case of compatible demands, both players receive what they demand. If the demands are incompatible, both players receive their disagreement payoffs. This game has an infinite number of Nash equilibria, which creates a major equilibrium selection problem. Only one equilibrium is stable, however, when the players are not quite sure what the feasible set X is. Nash proved (see [17]) that, when the uncertainty about the location of X is sufficiently small, this stable equilibrium corresponds to the regular Nash bargaining solution (see also the discussion in [23, 4]).

In our version of Nash's demand game, the two players are negotiating over several discrete issues (i.e., issues that cannot be divided). The players need to specify in their bids whether they claim a certain issue or not. If both players claim the same issue, the demands are considered to be incompatible and no deal is made. We consider a multi-round game, so the players have the opportunity to submit more than one bid. To perform well in this situation, a player should react on the opponent's bids and play differently against opponents with different preferences. In particular, it is important that a player concedes on a number of issues which are of major importance for the opponent, otherwise many demands will be incompatible.

We also consider two extensions of this game which are important in practical situations. In the first extension, the game does not always last a fixed number of rounds. Instead, the game can terminate prematurely with a certain probability after each round. Such premature termination of the bargaining process may occur in reality when players get dissatisfied as negotiations take too long. In the second extension, not all moves made by the opponent are actually received by a player. This models noisy communication between the two players or (equivalently) players who sometimes ignore the last move made by the opponent and remain in the same state. These two aspects (a stochastic risk of breakdown and noisy communication) further increase the difficulty of the bargaining problem.

More formally, we consider a simultaneous-move game which starts at time $t = 0$ and is terminated with a certain probability p after each round. The game also stops if the deadline is reached (at $t = n$). We set n equal to 10 in this paper. Both players submit demands in each round. When the bargaining process stops, the last demands made by the players are evaluated. If these demands are compatible (as specified below), both players receive what they demand. Both players receive nothing, however, if the demands are incompatible. The probability that a player receives the last bid of the opponent is denoted as p_{com}. By default, we set p_{com} equal to unity in this paper (unless stated otherwise).

We now specify when two demands are compatible. We denote a multi-issue demand as a vector $\vec{x}$. The i-th component of this vector, denoted as x_i, is equal to unity if the player demands issue i, otherwise x_i is equal to zero. The index i ranges from 0 to $m - 1$, where m is the total number of issues. Two demands $\vec{x}^A$ and $\vec{x}^B$ are now compatible if $x_i^A + x_i^B \leq 1$ for *all* $i = 0, ..., m - 1$. Stated otherwise, two demands are compatible if no issue is claimed by both players. Note that the situation can occur where an issue is not claimed by one of the two players. In this case, we assume that both players do not receive this issue. An alternative procedure would be to assign unclaimed issues to one of the two players (e.g., by assigning each issue which "remains on the table" at random to one of the players). Additional computational experiments show that the evolutionary techniques developed in

this paper also work when such a procedure is used.

The payoff received by a player is evaluated using an additive utility function. The utility function $u(\vec{x})$ can then be written as $u(\vec{x}) = \sum_{i=0}^{m-1} w_i x_i$, where $\vec{w}$ is an m-dimensional vector of "weights" for all issues. The weight vector is normalized (i.e., $\sum_{i=0}^{m-1} w_i = 1$).

3. Bargaining automata

We represent the bargaining strategies by a special kind of finite automata. Our automata require only two transitions per state. Their complexity is thus limited, which leads to a good performance in a computational setting.

Our automata are adaptations of Moore automata [12]. A Moore automaton can be represented by a four-tuple (S, s_{init}, h, f) where $S = \{s_0, s_1, ..., s_{q-1}\}$ is a finite set of q states; $s_{init} \in S$ is the automaton's initial state; $h(s) : S \rightarrow A$ is an output function, associating an action $h(s)$ from a set of actions A with every state $s \in S$; and $f(s, a)$ is a transition function identifying the state to which the automaton shifts after receiving action $a \in A$ as an input in state $s \in S$. Note that the set of possible outputs associated with a state (i.e., the set A) is simply equal to the set of all possible (multi-dimensional) demands in the negotiation, i.e., $\{0, 1\}^m$.

We now consider the problem of constructing an efficient transition function for the bargaining automata. In principle, arbitrarily complex mappings from actions to states are allowed. It is for instance possible to construct automata with q different transitions per state. Such automata are very complex, however, and are therefore not useful in a computational setting. We therefore propose a more efficient transition model, which requires automata with only two different transitions per state. In our transition model, we first associate an m-dimensional "transition threshold" $\vec{\tau_{tr}}$ with each state. Transition 1 is then selected in this model if $x_i \geq \tau_{tr,i}$ for all $i = 0, ..., m-1$. Otherwise, transition 0 is made. This procedure partitions the set A in only two disjunct sets.

For comparison, we also considered a more complex transition model. The $\vec{\tau_{tr}}$-vector is again attached to each state. The proper transition is selected as follows in this model. First, define an m-dimensional vector $\vec{c}$. The i-th component of this vector, c_i, is set equal to unity if $x_i \geq \tau_{tr,i}$, otherwise its value is set equal to zero. Transition i, with i equal to $\sum_{i=0}^{m-1} 2^{c_i}$, is then selected by the automaton. For complex bargaining problems with multiple issues (e.g., for $m = 5$) the performance of the computationally simpler transition model (with only two transitions per state) is significantly better. This indicates that our modelling choice for the transition function increases the efficiency in EAs for obtaining highly-fit automata.

4. Genetic representation

A bargaining strategy, consisting of a finite automaton, is represented as an ordered sequence of genes in our evolutionary model. All genes together constitute the chromosome. The typical structure of such a chromosome is shown in Fig. 1. Figure 1a shows the global structure of a chromosome which encodes a finite automaton with q internal states. States are indexed by the numbers $0, ..., q-1$. From left to right, the chromosome consists of several packets of genes containing (i) a pointer to the initial state s_{init} and (ii) the description of all internal states $s_0, ..., s_{q-1}$. By default, we use automata with 16 states in the experiments reported in this paper (i.e., $q = 16$).

The structure of a single state is shown in detail in Fig. 1b. One state contains the consecutive description of the (m-dimensional) demand $\vec{x}$, the components of the transition threshold $\vec{\tau_{tr}}$, and the components of the transition function $\vec{f}$ (being two states in our case). Figure 1c shows the genetic code of a simple two-state automaton (which is able to negotiate over 2 issues). Figure 1d is a graphical description of this automaton.

The initial state of this automaton is state s_0 (since $s_{init} = 0$). At $t = 0$, the automaton thus concedes issue 0 and claims issue 1 ($x_0 = 0$ and $x_1 = 1$ in state s_0). Now suppose that the opponent of this automaton claims both issues at $t = 0$. The demands are then incompatible (since both players claim issue 1). Assume that the game continues at $t = 1$. The automaton then shifts to state s_1

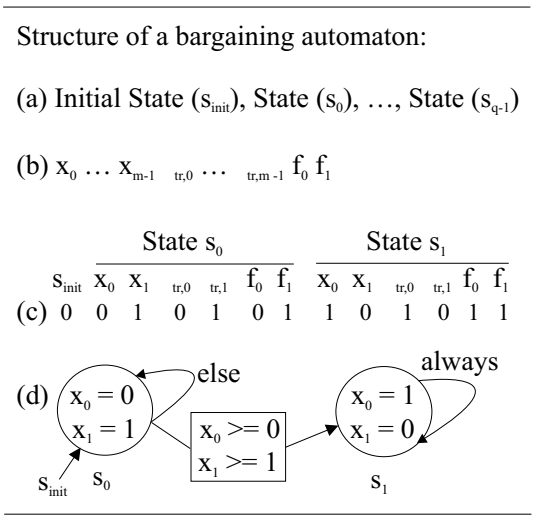

Figure 1: Genetic representation of an automaton. (a) The global structure of an automaton with q states. The initial state s_{init} refers to one of the q machine states $s_0 - s_{q-1}$. (b) Structure of a single internal state. (c) Genetic code of an example automaton. This automaton has two internal states and is capable of negotiating over two issues. (d) Structure of the automaton.

because the opponent's demand at $t = 0$ is not smaller than the transition thresholds specified in state s_0 (the opponent demands $x_0 = 1 > 0 = \tau_{tr,0}$ and $x_1 = 1 = \tau_{tr,1}$). The automaton therefore selects transition $f_1 = 1$. Because $f_0 = f_1 = 1$ in state s_1, the automaton will remain in this state during the remainder of the game.

5. The evolutionary system

5.1 Overview

We continue with a description of the evolutionary system. Our evolutionary system is essentially a multi-population EA. It is important to note in this respect that we do not allow the migration of automata from one population to another population. The EA, which evolves the automata, is described in detail in Section 5.2.

In Section 7, we consider the situation in which automata in 12 separate populations negotiate with each other over 4 issues. The interactions between the automata in the different populations are depicted schematically in Fig. 2. Each population of evolving automata is represented by a distinct node in this figure. The edges between the different nodes indicate which automata negotiate with each other. The topology of the network in Fig. 2 is such that automata in populations 1 through 6 bargain with automata in populations 7 through 12 (and vice versa). The weight vector $\vec{w}$ is also indicated (within the circle) for each population.

The bargaining network shown in Fig. 2 has a large degree of symmetry. We investigate in additional experiments (see Section 8) whether our approach also works in case of more asymmetric bargaining problems. The problem studied in Section 8 is depicted in Fig. 3. Note that the weights for the valuable items are not the same in this case (0.4 and 0.6 instead of 0.5 for both the valuable items). Furthermore, the weights for populations 2 and 8, populations 3 and 9, and populations 6 and 12 are different in this case (compare this with the symmetric situation shown in Fig. 2).

In Section 9, we consider situations where the players' preferences (i.e., weight vectors) are changing over time. We consider two different scenarios in these experiments. In the first scenario, the weight vectors of a limited number of populations change abruptly at certain points in time (i.e., the evolutionary system is perturbed by sudden, but local shocks). In the second scenario, the weight

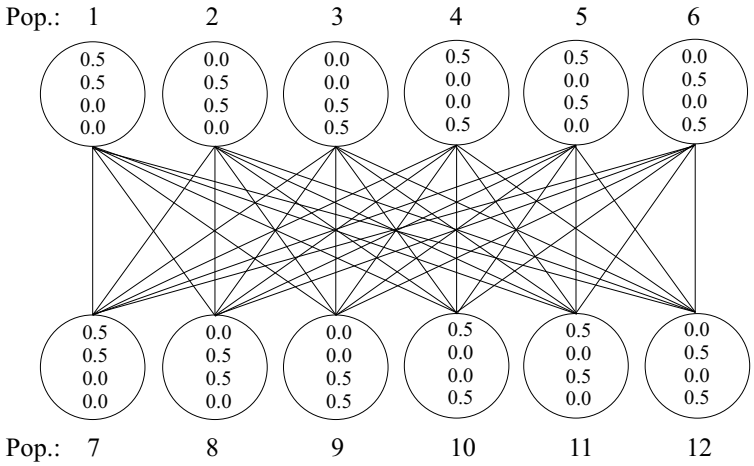

Figure 2: Topology of the bargaining network that we study in Section 7. Automata in 12 populations negotiate over 4 issues in this case. Each node in this network (indicated by a circle) represents a population of evolving automata. An edge between two nodes indicates that automata in the two connected populations can bargain with each other.

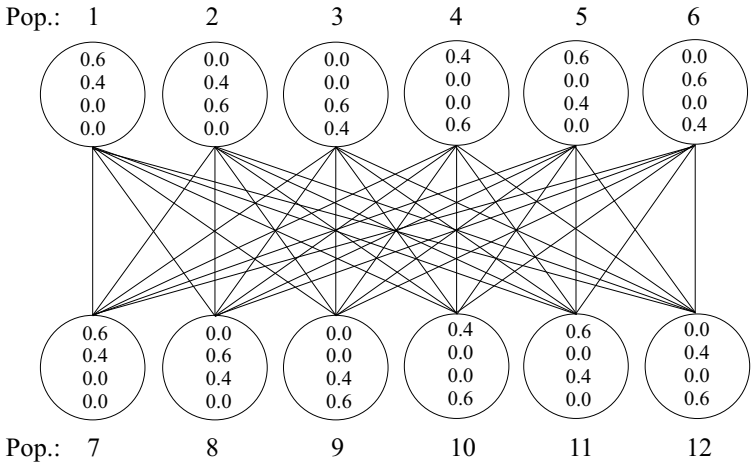

Figure 3: Topology of the asymmetric bargaining network that we study in Section 8.

vectors of all populations are gradually changing over time (i.e., the changes are more gradually, but affect the entire co-evolving system).

The first ("sudden shock") scenario is implemented as follows in the computational experiments. The first 200 generations, the co-evolving populations have the issue weights as shown in Fig. 2. At generation 200, we then permanently interchange the weight vectors of populations 1 and 2 (i.e., population 1's weight vector becomes $(0, 0.5, 0.5, 0)^T$ and population 2's weight vector becomes $(0.5, 0.5, 0, 0)^T$). The weight vectors of populations 10 and 11 are then interchanged at generation 300, of populations 5 and 6 at generation 400, of populations 8 and 12 at generation 500, of populations 7 and 9 at generation 600, and, finally, of populations 3 and 4 at generation 700. After generation 700, the weight vectors remain the same.

The second ("gradual transition") scenario also starts with the bargaining problem shown in Fig. 2. After generation 500, we then gradually adjust the issue weights in such a way that after 600 generations the populations have the issue weights as shown in Fig. 3. We thus make a slow transition from the symmetric bargaining problem shown in Fig. 2 to the asymmetric problem shown in Fig. 3. After generation 600, the weight vectors remain the same (i.e., as in Fig. 3).

The scalability of our approach is investigated in Section 10. In that section, we consider a highly complex problem involving automata in 20 populations which negotiate with 10 different opponents over 5 issues. This bargaining problem is depicted graphically in Fig. 4.

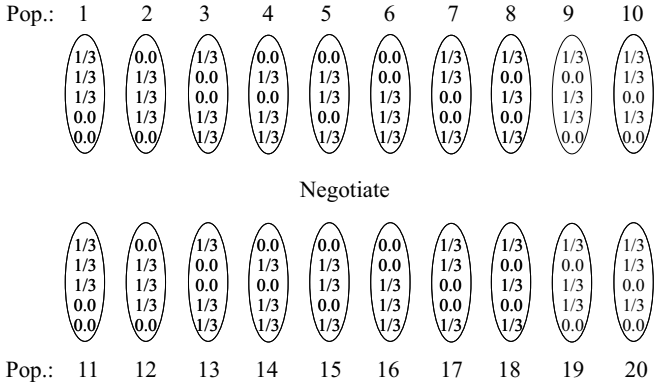

Figure 4: Topology of the bargaining network that we study in Section 10. Automata in 20 populations negotiate over 5 issues in this case. Automata in populations 1 through 10 negotiate with automata in populations 11 through 20 (and vice versa). The weight vector $\vec{w}$ is again indicated for each population. (Connections between the different populations are omitted for clarity.)

5.2 The evolutionary algorithm

The EA starts with a randomly initialized population of μ automata. The fitness of these automata is determined as follows. First, we select 5 opponent strategies (at random, without replacement) from each of the connected populations. The fitness of the automaton is then equal to the mean payoff obtained across all games (i.e., $6 * 5 = 30$ games in Sections 7–9 and $10 * 5 = 50$ games in Section 10). Recall that the payoff obtained by an automaton is evaluated using an additive utility function (see Section 2).

Subsequently, λ "offspring" automata are produced by the EA. A pair of offspring is created by (i) selecting two automata in the current "parental" population (at random, with replacement) and (ii) applying the appropriate genetic operators (i.e., mutation and crossover) to create two offspring automata from the parents. We use two-point crossover with a crossover probability of 0.6. The mutation probability is set equal to 0.01 (per bit). We follow the common practice to use a Gray-code interpretation of the bitstring segments. Additional experiments show that our settings for the mutation and crossover probabilities are chosen properly.

This process is repeated until λ offspring have been created. The fitness of the new offspring is then determined by negotiation with a pool of opponents drawn from the connected parental populations (as we described above). The μ fittest automata from the parental and offspring populations are then selected as the new parents. Finally, the fitness of each new parent is re-evaluated in a competition with a group of opponents from the new parental populations. This completes one iteration (or "generation") of the EA.

Our selection scheme is formally denoted as $(\mu + \lambda)$ selection [3]. We also investigated the performance of various alternative selection methods (e.g., fitness proportional and tournament selection).

Results obtained with $(\mu+\lambda)$ selection are superior to results obtained with these alternative selection schemes. By default, we set $\mu = \lambda = 25$ in the computational experiments. A further increase of μ or λ has no significant effect, so our settings appear to be a reasonable choice for the problem at hand.

We conclude this section with some comments on the complexity of the problem at hand. If we consider a bargaining problem with 4 issues, and if we assume that each game-playing automaton has 16 internal states, each strategy is encoded by a string of 260 bits (see Section 4). The total number of different strategies is thus equal to $2^{260} \approx 1.9 * 10^{78}$. Note, however, that the total number of unique strategies is much smaller, since some of the automata are isomorphic. For example, there are 16! ways to relabel the internal states of each of these machines. Furthermore, an automaton cannot always access all states (starting from the initial state). Obviously, the structure of the inaccessible states does not affect the functioning of an automaton. Hence, although the binary encoding of the inaccessible states can differ, the resulting machines will behave exactly the same. Despite these considerations, the total number of unique strategies is still huge.[1]

The above analysis considers the number of different strategies that can be generated using our finite automaton encoding. It is also useful to consider the number of different allocations that can be reached in a co-evolving system of automata. An allocation is a distribution of the different issues over the various players (for each encounter between two competing players, see the appendix for a precise definition). As an example, we determined the total number of allocations for a bargaining problem involving 12 players and 4 issues in the appendix. This analysis shows that the total number of *optimal* allocations (which is of the order of 10^9) is much smaller than the total number of allocations (which is of the order of 10^{39}). It therefore appears to be infeasible to generate optimal allocations by brute-force random search.

We also investigated if (near) optimal allocations can be generated by using a simple local search strategy. For this purpose, we repeated some experiments with a random-mutation hill climber [16, Ch. 4.2] instead of the $(25 + 25)$ EA which is used by default.[2,3] Poor results were obtained in this case. Our evolutionary approach thus appears to be a proper choice for the bargaining problem at hand: the search space is huge, the probability of finding an optimal solution by random search is very small, and the application of a parallel search algorithm appears to be necessary to obtain good results.

6. Overview of the computational experiments

The remainder of this paper, which contains the computational results, is organized as follows. Section 7 contains results for the bargaining problem shown in Fig. 2 (where automata in 12 populations negotiate over 4 issues). The asymmetric bargaining problem shown in Fig. 3 is studied in Section 8. Section 9 then considers situations where the players' preferences are changing over time. Finally, Section 10 contains results for the complex bargaining problem shown in Fig. 4 (where automata in 20 populations negotiate over 5 issues). We refer to Section 5.1 for a detailed overview of the different bargaining problems considered in this paper.

7. Automata in 12 populations negotiate over 4 issues

We investigate the performance of the co-evolving automata from three different perspectives in this section. In Section 7.1, we consider the performance of the entire system of co-evolving populations. We next investigate the performance of the individual populations in Section 7.2. Finally, we examine the structure of the evolved automata in more detail in Section 7.3.

[1] Miller gives an estimate of 10^{21} for 16-state automata with a length of 148 bits [15, p. 94]. Our automata are more complex, so the total number of unique strategies will be even larger.

[2] We implemented a random-mutation hill climber by setting $\mu = \lambda = 1$ in the $(\mu+\lambda)$ selection scheme. We performed experiments in which the mutation probability (per bit) was constant (similar to a zero-temperature Metropolis method). In addition, we experimented with an exponentially decaying mutation probability (similar to a simulated annealing approach). Both approaches yield poor results.

[3] These alternative experiments were performed for the bargaining problem shown in Fig. 2.

7.1 Performance of the entire system of co-evolving populations

Figure 5 shows the performance of the co-evolving automata for the bargaining problem shown in Fig. 2. We consider the performance of the whole system of 12 co-evolving populations in Fig. 5. We

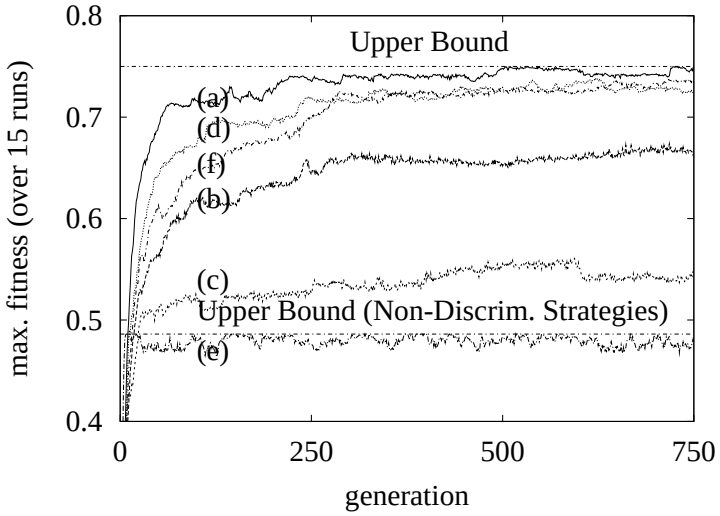

Figure 5: Maximum fitness level reached by the system of co-evolving automata for different bargaining models (across 15 EA-runs). Results are shown for (a) $p = 0$, (b) $p = 0.1$, (c) $p = 0.5$, (d) $p = 0.5$ (after 5 rounds), (e) $p = 1$, and (f) $p_{com} = 0.5$ (and $p = 0$).

define the performance of the whole system as the mean fitness across all strategies in the evolutionary model. Recall that each population consists of 25 strategies (see Section 5.2), so fitness is averaged across $12 * 25 = 300$ strategies in total. We repeated the experiments 15 times (with different random seeds). The maximum fitness level reached across these runs is reported in Fig. 5.

The performance of the co-evolving automata is compared in Fig. 5 with the theoretical upper bounds for discriminating and non-discriminating players. Non-discriminating players submit the same demand to *all* opponents. This situation also occurs in our bargaining model when the game only lasts a single round (or, equivalently, when $p = 1$). In this case, there are basically three options for a player: claim both issues with a non-zero weight,[4] or claim only one of the issues. These 3 alternatives for each of the 12 players yield a total of 3^{12} ($\approx 5.3 * 10^5$) possible bids. Evaluating all possibilities, we find that the maximum fitness level that can be reached in this case is ≈ 0.486.[5]

When $p < 1$, the negotiations can last for more than one round in our bargaining model. This allows players to adjust their bid, depending on the opponent they are facing. The maximum fitness level that can be reached in this case is equal to 0.75. This upper bound is reached when each player receives those issues to which he attaches more weight than his opponent. If both players have an equal weight for the same issue (arbitrarily) one of the two should receive it. This (utilitarian) bargaining solution maximizes the sum of utilities obtained by the two parties.

Figure 5 shows that the evolving automata reach the theoretical upper bound for discriminating players after ≈ 500 generations when $p = 0$ (see curve a). In the long run, the evolving automata are thus able to discriminate perfectly between all (six) opponents. This is a positive result, since

[4]Note that only two out of four issues have a non-zero weight for each player, see Fig. 2.

[5]When we also allow players to claim no issue at all, the maximum fitness level that can be reached is slightly higher (0.5). These states are not evolutionarily stable, however, because some "altruistic" players receive nothing at the end of the bargaining game in this case (they concede everything to their opponent). For these players, there is no incentive to keep conceding everything to their opponent, since they receive the same (namely, nothing) in case of a disagreement. Less altruistic strategies can thus easily invade such a player's strategy pool.

the automata receive no explicit information about the identity or preferences of their opponents. In fact, the only feedback used by the EA is the *average* score obtained by the automata against the different opponents. Furthermore, the automata do not play against all opponents in the connected populations (see Section 5.2). The EA thus only has an estimate of the performance of an automaton (i.e., the fitness determination is "noisy").

Figure 5 also shows that the performance of the evolving automata remains rather high when the probability of breakdown is small (e.g. 10% per round, see curve b). When this breakdown probability becomes very large, however, most negotiations are terminated quickly. For example, if $p = 0.5$ (see curve c in Fig. 5), 75% of all games terminates after one or two rounds. This diminishes the opportunity for the automata to discriminate successfully between the different opponents, and, as a result, the performance level drops. The performance of the bargaining automata improves significantly if a risk of breakdown only exists after a few negotiation rounds have been played. For instance, if we set p equal to 0.5 after five negotiation rounds (see curve d), a high fitness level is again reached by the automata.

Curve f in Fig. 5 shows results obtained for p_{com} =0.5. In this case, the automata receive the last bid of their opponent with a probability of only 50%. When an automaton does not receive a new bid, no transition will be made to a new state. This means that two game-playing automata cannot easily synchronize their bids, which, as a result, makes it much more difficult for them to reach a jointly beneficial outcome. Figure 5 shows, however, that a high fitness level is still reached when p_{com} is as small as 0.5. This indicates that the EA generates bargaining automata which can deal successfully with opponents who sometimes ignore the last bid and remain in the same state.

To summarize, these results show that the co-evolving system of automata yields near-optimal outcomes (from a utilitarian point of view) for a multi-issue bargaining problem which involves multiple opponents (with different preferences). Moreover, performance remains high in case of stochastic environments (i.e., stochastic breakdown or noisy communication between the automata). In the next section, we consider the performance of the individual populations of evolving automata.

7.2 Performance of the individual populations

In the previous section, we analyzed the performance of the whole system of 12 co-evolving populations (we reported the mean fitness across all strategies). We now consider the performance of the individual populations during the evolutionary experiments. The performance (fitness) is therefore defined in this section as the mean fitness across all (25) strategies in a population.

It is important to note at this point that an optimal performance of the complete system does not necessarily imply that the actual outcomes inside the system should be "fair" (i.e., symmetric). Stated otherwise, there exist asymmetric outcomes which are still optimal from a global (utilitarian) point of view (see the analysis in the appendix). An example is given in Fig 6. This figure shows the fitness of each of the 12 populations during one run of the EA (for $p = 0$ and $p_{com} = 1.0$). Note the evolution of asymmetric outcomes in the long run.

The mean fitness level reached in Fig 6 (across all populations) is ≈ 0.727 after 1500 generations. This means that the system's performance is very high in the long run (only 3% below the optimum of 0.75). In the long run, all populations also perform better than non-discriminating players (who receive at most ≈ 0.49, see Section 7).

Note the complex dynamic behavior of the co-evolving system of automata. The fitness of the populations increases rapidly in the early stages of the evolutionary process. On a longer time scale, the system goes through periods of stasis (metastable states) interrupted by fast transitions to unstable dynamic behavior or to new periods of stasis. This behavior resembles the complex dynamics encountered in other (computer-based) ecological models, such as Thomas Ray's Sierra model [19] or Lindgren's simulations of the IPD [13]. The differences between the individual populations (see Fig. 6) are due to stochastic factors in the initial generations. The "lock-in" behavior on longer time scales is caused by convergence of the separate EAs.

We investigated in a series of additional experiments whether more symmetric outcomes of the bar-

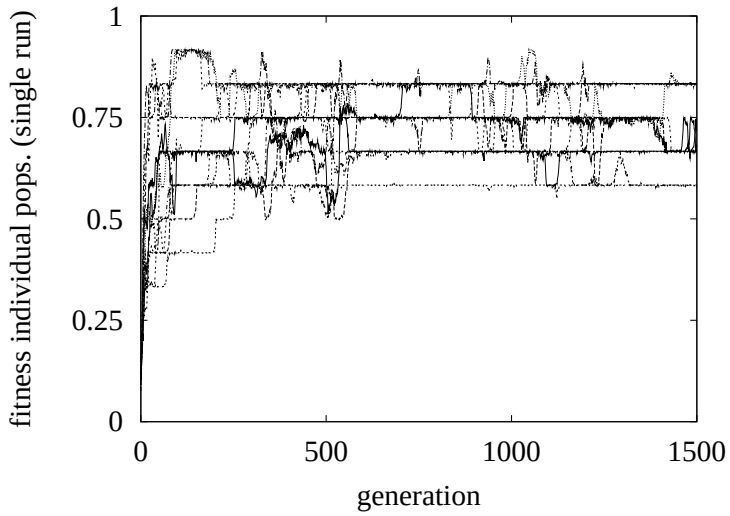

Figure 6: Fitness of the individual populations during a single run of the EA. Note the complex dynamic behavior of the co-evolving system. The performance of the complete system is very high in the long run (close to the utilitarian optimum).

gaining process can be reached in the course of evolution. We experimented for instance with a variety of different EAs (with different selection schemes, mutation operators, selection intensities, etc.). In addition, several methods for competitive co-evolution (like fitness sharing, shared sampling, the hall of fame, or balanced co-evolution) [21, 8, 20, 18] were implemented and evaluated. However, neither the alternative EAs nor the methods for competitive co-evolution yield more symmetric outcomes in the long run.

It thus appears to be difficult to generate "fair" automata by competitive co-evolution for the bargaining problem that we consider here. An explanation for the consistent evolution of asymmetric outcomes may be that only $\approx 3.49 * 10^4$ out of a total of $\approx 1.07 * 10^9$ optimal allocations yield the same fitness for all 12 players (see the analysis in the appendix). Stated otherwise, only a tiny fraction ($\approx 0.003\%$) of all possible optimal allocations is symmetric. The probability that an optimal and symmetric allocation is generated by the multi-population EA is thus very small.

7.3 Analysis of the evolving automata

We start the analysis of the evolving automata with some remarks about their complexity. We then address an important issue in the field of co-evolutionary learning – the generalization issue. This issue concerns the question whether the strategies developed by co-evolution are generally applicable, i.e., if they also work well against a variety of unseen opponent strategies. We here test the generalization ability of the evolved automata in two different ways. First, we let the evolved automata compete against opponents that were generated in independent evolutionary simulations (i.e., experiments with different seeds for the random number generators). This gives an indication of how strongly the performance of the evolved strategies depends on stochastic processes which occur in the course of evolution. We then give an example of an evolved automaton which performs well against different opponent types. To conclude, we investigate the performance of the evolved automata when playing against a test suite of non-evolutionary opponents.

Complexity of the evolving automata
A frequently-used complexity measure is the number of states that is accessible by an automaton.[6] A state is accessible if, given the automaton's starting state, there is some possible combination of moves by the opponent that will result in a transition to the state. The mean number of accessible states across all (300) automata in the evolutionary system is equal to 11 ± 1 (out of a possible 16).[7,8] The complexity of the evolving automata can vary significantly, however, in the course of the evolutionary process and from population to population.

Performance against unseen evolutionary opponents
Table 1 gives an overview of the performance of the top-performing automata (after 250 and 1000 generations) against the best automata from the same and other EA runs. We collected the fittest

training length	opponents	fitness
250 generations	same EA run	0.71 ± 0.29
250 generations	other EA run	0.29 ± 0.39
1000 generations	same EA run	0.73 ± 0.26
1000 generations	other EA run	0.30 ± 0.40

Table 1: Performance of the top-performing automata (after 250 and 1000 generations) when competing against the top performers from the same and other EA runs (for $p = 0$ and $p_{com} = 1$). Standard deviations are calculated across all outcomes of the two-player games.

automata of 15 independent EA runs, so each automaton plays against 6 opponents from the same EA run and $14 * 6 = 84$ (unseen) opponents from the other EA runs. Table 1 shows that the performance of the top-performing automata is very low when they are matched with the top performers of other EA runs. Stated otherwise, the performance against the unseen opponents is very poor.

The poor performance against opponents from other evolutionary runs can be explained as follows. In the bargaining game considered here (with $p = 0$ and $p_{com} = 1$), the automata can exchange offers "for free" in the first 9 rounds. That is, only the offers made in the 10^{th} (and last) round determine the payoffs received by the players. Consequently, the sole purpose of the offers submitted in the first 9 rounds is to arrive at the proper state (by a series of transitions) once the final round is reached. When automata co-evolve, they can coordinate their bids in the first 9 rounds in such a way that they reach the proper state at the end of the game.

However, when the evolved automata are matched with unseen opponents (from a different EA run), the probability that the joint moves made by two competing automata lead to the proper end state becomes smaller (because their moves are probably not the same due to stochastic factors). So, stated otherwise, the pre-negotiation phase (the first 9 rounds) becomes less effective because the automata no longer exchange the proper "signalling moves".

Even when the appropriate end state is reached, coordination problems can occur between two competing automata from different EA runs. For example, when two automata both value the same issue, one of the two should concede that issue to the opponent (otherwise the deal is incompatible). Two automata from different EA runs do not necessarily coordinate their final demands in a proper way, so many bids will be incompatible. Note also that an automaton can (in principle) demand an issue which is not valuable for him, as long as its co-evolved opponent does not claim the same issue. Claiming an issue which is not valuable can, on the other hand, lead to incompatible deals when playing against unseen opponents from other EA runs (who may claim the same issue).

[6] Other measures of complexity do exist. See [15] for a further discussion of this topic.

[7] For $p = 0$ and $p_{com} = 1$. The complexity of the evolving automata does not change significantly if we consider variations of the basic game (e.g., with p_{com} equal to 0.5 or with p equal to 0.1).

[8] These statistics were determined after 1000 generations. The mean and standard deviation were calculated across 15 EA-runs.

A partial solution for the above problems is to make it more difficult for the automata to condition their behavior precisely on the moves made by their co-evolving opponents. This way, a premature lock-in to the idiosyncratic behavior of the co-evolving players can be avoided to a certain extent. We already mentioned experiments in which the communication between the competing automata was disturbed (in a stochastic fashion). In these experiments, an automaton only receives the last move of the opponent with a probability $p_{com} < 1$. Table 2 gives an overview of the performance of the top-performing automata (after 250 and 1000 generations) if we generate the automata with the setting $p_{com} = 0.5$ (instead of 1 as in Table 1). The results shown in Table 2 are obtained for the

training length	opponents	fitness
250 generations	same EA run	0.69 ± 0.28
250 generations	other EA run	0.44 ± 0.39
1000 generations	same EA run	0.71 ± 0.27
1000 generations	other EA run	0.44 ± 0.41

Table 2: Performance of the top-performing automata (after 250 and 1000 generations) when competing against the top performers from the same and other EA runs. These automata were trained in a game with noisy communication ($p_{com} = 0.5$ and $p = 0$).

same game as in Table 1 (i.e., $p = 0$ and $p_{com} = 1$). Note that the performance against the unseen opponents (from other EA runs) increases (from ≈ 0.30 to ≈ 0.44), while the performance against the opponents from the same EA run remains high (≈ 0.70). It thus appears to be possible to generate more robust automata when, during the training phase, the automata sometimes ignore the last move made by the opponent and remain in the same state. It is important to note here that this stochastic training method does not produce "random" strategies. This becomes clear when we compare the performance levels in Table 2 with the (much lower) fitness level that is reached by random players (0.11 ± 0.24). An example of a high-performance automaton (produced by the stochastic training method) is also given below.

In an additional series of experiments, we also studied the generalization ability of the evolved automata in bargaining games with premature breakdown (i.e., with $p > 0$). These experiments show that the performance against unseen automata (from other EA runs) improves in this case. For example, for a breakdown probability p equal to 0.1 (and $p_{com} = 1.0$) the performance against the top performers of other EA runs is equal to 0.45 ± 0.40 (instead of only 0.30 ± 0.40 when $p = 0$, see Table 1).[9] The performance against the top performers from the same EA run also remains high in this case: 0.69 ± 0.29.

The improved generalization ability for $p > 0$ is due to the fact that successful automata should claim one or more valuable issues in each round in this case (otherwise a player may receive nothing if the game is terminated prematurely). This makes the behavior of the automata more predictable than in the game without premature breakdown ($p = 0$), where the automata can in principle exchange arbitrary moves as long as the proper issues are claimed in the very last round. Interestingly, the automata that were generated with the setting $p = 0$ and $p_{com} = 0.5$ also perform satisfactory in a game with a small risk of breakdown ($p = 0.1$ and $p_{com} = 1.0$): the score against the top-performers from the same EA run is equal to 0.64 ± 0.33, whereas the score versus the top performers from the other EA runs is 0.40 ± 0.41. (The corresponding numbers for the automata trained with the settings $p = 0$ and $p_{com} = 1.0$ are much lower: 0.46 ± 0.40, respectively 0.26 ± 0.37.) These results indicate that automata trained in a game with noisy communication may also perform well in a bargaining game with a premature risk of breakdown.

As we argued above, a natural measure of an automaton's generalization ability (against evolutionary opponents) is its performance against the unseen top-performers from other EA runs. Using

[9] After a training period of 1000 generations.

this criterion, we selected one automaton (with the highest score against the unseen opponents) for each population. The fitness statistics across these 12 automata are given in Table 3. To be expected

training length	training setting	min. fitness	av. fitness	max. fitness
250 generations	$p_{com} = 1.0$	0.35	0.40	0.47
1000 generations	$p_{com} = 1.0$	0.30	0.40	0.45
250 generations	$p_{com} = 0.5$	0.51	0.56	0.59
1000 generations	$p_{com} = 0.5$	0.44	0.54	0.67

Table 3: Performance of the 12 automata (one for each population) with the highest score against the top performers from the other EA runs. We consider the best automata trained in a game with noisy communication ($p_{com} = 0.5$) and the best automata trained in a game with perfect communication ($p_{com} = 1$). The generalization ability of the automata (as reported in this table) is evaluated in a game with perfect communication (i.e., with $p_{com} = 1$).

(given the results shown in Tables 1 and 2), the automata trained with the setting $p_{com} = 0.5$ appear to generalize best across a variety of unseen opponents.

Example of an evolved automaton
An example of an evolved automaton is given in Fig. 7. This automaton was drawn from population

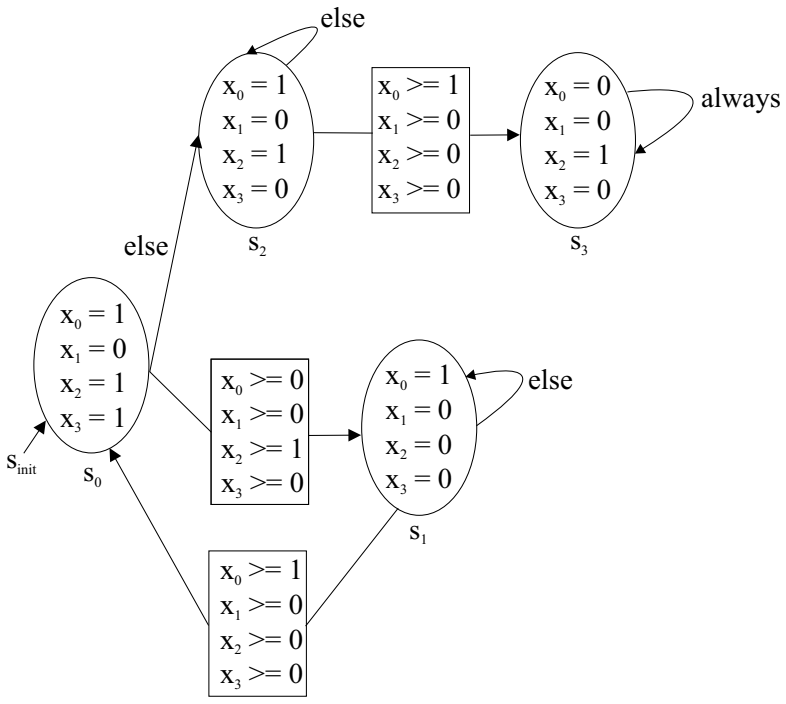

Figure 7: Structure of an evolved automaton. Issues 0 and 2 are valuable for this automaton.

11 after a training period of 250 generations (with the settings $p_{com} = 0.5$ and $p = 0$). The valuable issues for this automaton are issues 0 and 2 (cf. Fig. 2). This automaton can only reach 3 other states from the initial state; the 12 other (inaccessible) states are not shown in Fig. 7.

Despite its simple structure, the automaton reaches a high score (0.56 ± 0.32) against the unseen top-performers from other EA runs (i.e., its generalization ability appears to be good). The automaton

starts by claiming issues 0, 2, and 3 in the initial state (s_0). The automaton then makes a transition to state s_1 if the opponent also claims issue 2. It then concedes issues 2 and 3 to the opponent and only demands issue 0. Note that conceding issue 3 is not a "real" concession for the automaton, since it attaches no weight to this issue. When the opponent also claims issue 0 in state s_1, the automaton returns to the initial state. If, starting from the initial state, the opponent does not claim issue 2, the automaton shifts to state s_2 and responds by demanding issues 0 and 2. If the automaton is in state s_2 when the opponent demands issue 0, the automaton switches (permanently) to state s_3 and concedes issue 0 to the opponent (while standing firm on issue 2).

The automaton in Fig. 7 has thus clearly adapted its transition thresholds to detect whether the opponent is claiming an issue which is important for the automaton as well. In this case, a transition is made to another state and a proper concession is made on one or more issues. This behavior results in mutually beneficial deals against a variety of opponents: depending on which issues are claimed by the opponent, the automaton is able to rapidly switch to the proper state and avoid the occurrence of incompatible bids.

Performance against non-evolutionary opponents
We now consider the generalization ability of the evolved automata when competing against a test suite of non-evolutionary opponents. For this purpose, we generate test opponents in a similar fashion as done by Darwen and Yao [24, 8]. They propose to choose the test strategies from a large random sampling of all possible strategy genotypes. Choosing the test strategies this way means that there is no human input about what a "good" strategy ought to be, i.e., any human favoritism is avoided. Strategies which are selected from a large sampling of genotypes also have the advantage that they perform acceptably against a broad range of opponents (i.e., they are not brittle).

We select the test strategies in the following way. First, we generate 12 groups of test strategies. Each group of test strategies consists of 250 randomly initialized automata. The test strategies from the different groups then compete with each other (given the topology shown in Fig. 2). After evaluating the performance of all test strategies, the best automaton of each of the 12 groups is stored. This procedure is repeated 15 times with different seeds for the random generators, yielding a total of $15 * 12 = 180$ test strategies in total.

The performance of the top-performing automata (from 15 independent EA runs) against these test opponents is summarized in Table 4. Table 4 shows that the top-performing automata perform well

training length	training setting	fitness automata	fitness test opponents
250 generations	$p_{com} = 1.0$	0.39 ± 0.41	0.32 ± 0.34
1000 generations	$p_{com} = 1.0$	0.39 ± 0.42	0.32 ± 0.34
250 generations	$p_{com} = 0.5$	0.41 ± 0.39	0.36 ± 0.35
1000 generations	$p_{com} = 0.5$	0.42 ± 0.41	0.35 ± 0.34

Table 4: Performance of the top-performing automata (after 250 and 1000 generations) when competing against the suite of (non-evolutionary) test opponents. The test opponents reach a fitness level of 0.37 ± 0.35 when they compete against each other.

against the collection of non-evolutionary test opponents. In fact, their mean performance level exceeds the fitness level reached by the test opponents when they compete against each other (0.37 ± 0.35).

8. Asymmetric bargaining problems

Figure 8 shows the performance of the co-evolving automata for the asymmetric bargaining problem shown in Fig. 3. The performance of the co-evolving automata is again compared with the theoretical upper bounds for discriminating players (≈ 0.772) and non-discriminating players (≈ 0.528). Note that the results shown in Fig. 8 are very similar to the results for the symmetric bargaining problem

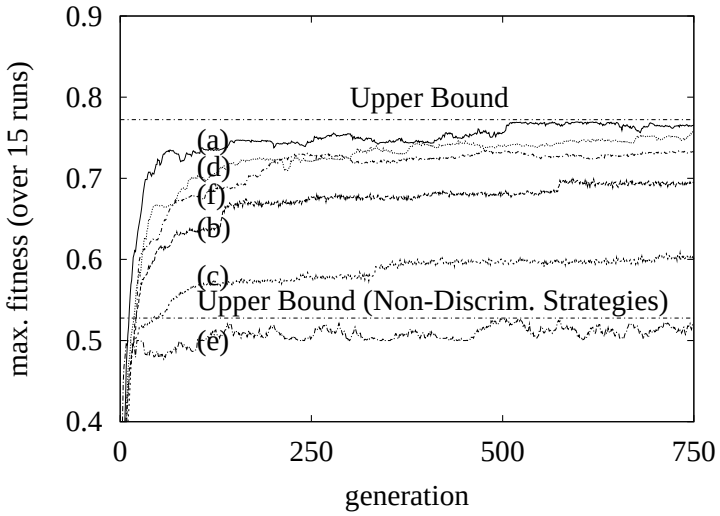

Figure 8: Maximum fitness level reached for an asymmetric bargaining problem. As in Fig. 5, results are shown for (a) $p = 0$, (b) $p = 0.1$, (c) $p = 0.5$, (d) $p = 0.5$ (after 5 rounds), (e) $p = 1$, and (f) $p_{com} = 0.5$ (and $p = 0$).

(cf. Fig. 5). In particular, performance is again high (in the long run) for the corresponding curves (a,d,f). We performed additional experiments for other (asymmetric) bargaining problems. The long-run performance of the evolving automata remained consistently high in these experiments. We can thus conclude that our approach works well for both symmetric and asymmetric bargaining problems.

9. Time-dependent preferences

Figure 9 shows the performance of the co-evolving populations in case the players' preferences are changing abruptly at certain points in time (the scenario with "sudden shocks" described in Section 5.1). Results are shown for $p = 0$ (and $p_{com} = 1.0$). From generation 0 to generation 200, the fitnesses of the evolving populations increase rapidly to a high level. At generation 200, the issue weights of populations 1 and 2 are interchanged. This leads to an immediate drop in fitness for these populations. After generation 200, the automata in populations 1 and 2 are re-optimized given the modified weights attached to the different issues. Figure 9 clearly shows that this adjustment process leads to a rapid increase in fitness of populations 1 and 2. It is important to note that the fitnesses of the other populations remain approximately the same after the change at generation 200. This indicates that the co-evolving system is rather robust: a local perturbation does not undermine the (high) performance of the other parts of the system.

At generation 300, the preferences of populations 10 and 11 are interchanged. Again, the fitness of these populations drops significantly initially, but then returns to high values afterwards (while the performance of the other populations remains high). At generation 400, the preferences of populations 5 and 6 are switched. Note that this leads to a very large drop in fitness for these two populations (compared with the fitness collapses after 200 and 300 generations). This can be explained by the fact that the issue weight vectors for these two populations do not overlap. That is, an issue which is valuable for the automata in one population is not valuable at all for the automata in the other population. Consequently, the evolved automata in populations 5 and 6 claim the wrong issues when the issue weights are interchanged at generation 400. Note, however, that the fitnesses of populations 5 and 6 increase rapidly after generation 400.

After the next perturbations at generations 500, 600, and 700, the issue weights remain constant.

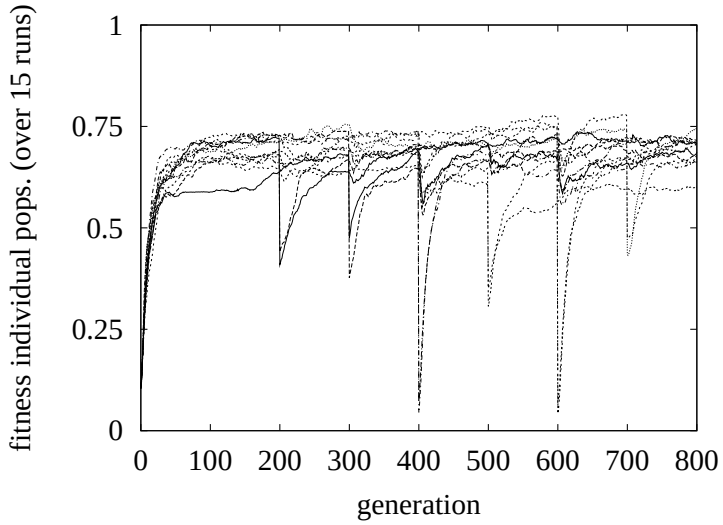

Figure 9: Performance of the co-evolving populations when the players' preferences are suddenly changing at certain points in time (at generation 200, 300, ...,700). (Standard deviations are omitted for clarity.)

In the long run, the co-evolving system reaches very high performance levels. In some runs, optimal outcomes (yielding a global fitness level of 0.75) were in fact reached. The co-evolving system thus appears to be able to "recover" successfully from a series of sudden shocks.

We now consider the scenario where the players' preferences are gradually changing over time (see Section 5.1). Until generation 500, the preferences of the players are as shown in Fig. 2. From generation 500 to generation 600 we then gradually adjust the issue weights in such a way that after 600 generations the populations have the issue weights as shown in Fig. 3. Figure 10 shows the performance of the co-evolving system of automata for this scenario. We set $p = 0$ and $p_{com} = 1.0$

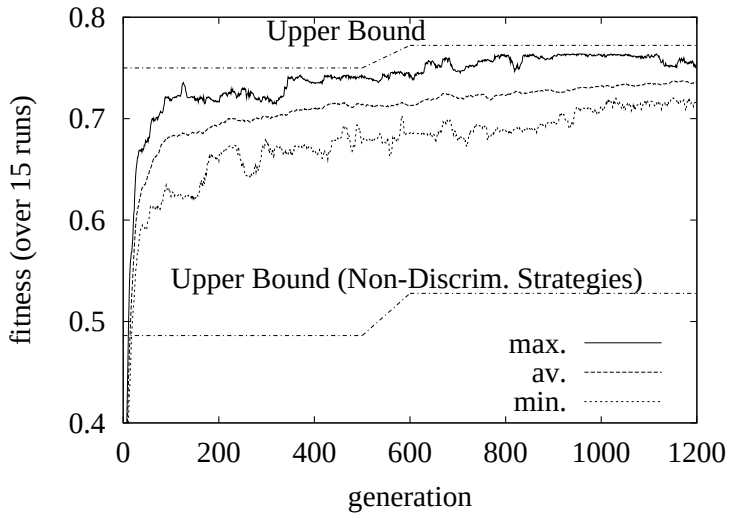

Figure 10: Performance of the co-evolving system of automata when the players' preferences are gradually changing from generation 500 to generation 600.

in these experiments. Note that the evolution towards higher fitness levels is not disturbed by the gradually changing preferences (from generation 500 to generation 600). The co-evolving system of automata thus continues to operate at a high performance level when the players' preferences are gradually changing over time.

10. Automata in 20 populations negotiate over 5 issues

We now investigate the scalability of our approach by considering a much more complex bargaining problem. This bargaining problem involves 20 co-evolving populations, where each automaton negotiates with automata from 10 different populations over 5 issues (see Fig. 4). We set $p = 0$ and $p_{com} = 1$.

The theoretical upper bound for this bargaining problem is equal to 0.7. The upper bound for non-discriminating players cannot be determined quickly by means of exhaustive search for this network [there are 7 alternatives for each of the 20 populations yielding a total of 7^{20} ($\approx 8.0 * 10^{16}$) possible bids]. We therefore estimated this upper bound based upon simulations with our multi-population EA.[10] The best outcome (for non-discriminating players) that we generated this way has a fitness level of ≈ 0.49.

The maximum fitness level that is reached by the system of co-evolving automata (in 15 independent EA-runs, with each run lasting 1200 generations) is much higher (≈ 0.673, i.e. only 4% below the theoretical upper bound of 0.7). We can therefore conclude that our evolutionary approach continues to work very well in case of highly complex multi-issue/multi-opponent bargaining problems.

11. Conclusions

We have shown in this paper how bargaining strategies can be generated that are able to distinguish successfully between different opponents (with different preferences). We represent the bargaining strategies by a special kind of finite automata, which require only two transitions per state. Such automata (with a limited complexity) are a suitable representation in a computational setting. Computational experiments indeed show that our modelling choice for the transition function makes it possible for an EA to rapidly find highly-fit solutions. In particular, we show in this paper that our approach (automata with a limited complexity which are optimized by an EA) yields very efficient bargaining strategies for complex bargaining problems involving multiple issues and multiple opponents (with different preferences).

Results obtained with the multi-population EA are in fact very close to the (utilitarian) optimum, even though the automata receive no explicit information about the identity or preferences of their opponents. Moreover, performance of the evolving automata remains high in case the bargaining problem is (i) stochastic (e.g., because of stochastic breakdown of the bargaining process or because of noisy communication between the negotiating parties), (ii) asymmetric (e.g., because the players' preferences for the valuable items are not the same), or (iii) time-dependent (e.g., because the players' preferences are changing over time). Furthermore, our evolutionary approach appears to scale well for more complex bargaining problems. We generated for instance very efficient automata for a problem with 20 co-evolving populations, where each automaton negotiates with automata from 10 different populations over 5 issues. The evolutionary techniques developed in this paper thus appear to be important for the further development of evolving automata for real-life agent system applications.

APPENDIX

1. Further Analysis of the Bargaining Problem

This appendix further analyzes the bargaining problem shown in Fig. 2 (involving 12 players and 4 issues). We determine the total number of allocations for this problem in Appendix 1.1. An allocation

[10] By setting $p = 1$ in the bargaining model.

is defined here as a distribution of the different issues over the various players (for each encounter between two competing players). Two allocations are considered to be similar if they differ only in the allocation of an issue to a player who has a zero weight for that issue. We do not distinguish between such (payoff-equivalent) allocations, and, instead, count them as only one distinct allocation.

We then calculate the total number of optimal allocations in Appendix 1.2. An allocation is said to be optimal (in a utilitarian sense) if the outcome of an encounter between a pair of competing players maximizes the sum of their utilities. We furthermore require that both parties receive a positive payoff (to avoid optimal solutions that are evolutionarily unstable). Appendix 1.3 reports the total number of optimal allocations which are symmetric (i.e., yield the same payoff to all players).

1.1 Number of Allocations

Assume that player 1 reaches a deal with player 7 (see Fig. 2). Now consider the issues valued by these two players (issues 0 and 1). There exist 3 possible allocations of each of these two issues (either player 1, or player 7, or none of the two players receives the issue), so we have 9 possible allocations of issues 0 and 1 in total (the allocation of issues 2 and 3 is not considered, since both players attach a zero weight to them). When player 1 meets player 8, 12 allocations are possible: 2 possible allocations of issue 0 (which is only valued by player 1), 2 allocations of issue 2 (which is only valued by player 8), and 3 for issue 1 (which is valued by both players). Similarly, the number of possible allocations when player 1 meets player 10, 11, or 12 is also equal to 12. There exist 16 possible allocations when player 1 meets player 9 (each issue is valued by only one of the two players). The total number of possible allocations involving player 1 is thus equal to $9 * 16 * 12^4 = 2985984$. By repeating this procedure for players 2-6, it becomes clear that the total number of allocations is equal to $2985984^6 \approx 7.09 * 10^{38}$.

1.2 Number of Optimal Allocations

Assume that player 1 reaches a deal with player 7 which maximizes the sum of their utilities. Assume furthermore that both players receive a positive payoff (we ignore optimal solutions that are evolutionarily unstable). In that case, either player 1 receives issue 0 and player 7 receives issue 1, or vice versa (again, the allocation of issues 2 and 3 is not considered, since both players attach a zero weight to them). Similarly, assume that player 1 reaches an optimal deal with player 8. Again, two possibilities exist: player 1 receives issue 0 and player 8 issues 1 and 2 or player 1 receives issues 0 and 1 and player 8 issue 2. Similarly, two optimal allocations exist when player 1 meets player 10, 11, or 12. When player 1 meets player 9, however, only one optimal allocation exists (player 1 receives issues 0 and 1 and player 9 receives issues 2 and 3). The total number of optimal allocations involving player 1 is thus equal to 2^5. By repeating this procedure for players 2-6, it becomes clear that the total number of optimal allocations is equal to $(2^5)^6 = 2^{30} \approx 1.07 * 10^9$. The total number of optimal allocations is thus a very small fraction of the total number of allocations.

1.3 Number of Symmetric Optimal Allocations

We determined the number of symmetric optimal allocations (34944 in total) by generating all optimal allocations (on a computer) and filtering out the symmetric ones. Note that the total number of symmetric optimal allocations is a very small fraction of the total number of optimal allocations.

References

1. D. Ashlock. GP-automata for dividing the dollar. In *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 18–26, Stanford University, CA, USA, 13-16 July 1997. Morgan Kaufmann.
2. D. Ashlock and C. Richter. The effect of splitting populations on bidding strategies. In *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 27–34, Stanford University, CA, USA, 13-16 July 1997. Morgan Kaufmann.
3. T. Bäck. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, New York and Oxford, 1996.
4. K. Binmore. *Fun and Games.* D.C. Heath and Company, Lexington, MA, 1992.
5. D. Carmel and S. Markovitch. Learning models of intelligent agents. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Portland, Oregon, 1996.
6. D. Carmel and S. Markovitch. Exploration strategies for model-based learning in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 2(2):141–172, 1999.
7. A. Chavez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. In *Proceedings of the 1st International Conference on the Practical Appication of Intelligent Agents and Multi-Agent Technology*, London, April 1996.
8. P.J. Darwen and X. Yao. Speciation as automatic categorical modulatization. *IEEE Transactions on Evolutionary Computation*, 1(2):100–108, 1997.
9. D.B. Fogel. The evolution of intelligent decision-making in gaming. *Cybernetics and Systems*, 22:223–236, 1991.
10. D.B. Fogel. Evolving behaviors in the iterated prisoner's dilemma. *Evolutionary Computation*, 1(1):77–97, 1993.
11. L.J. Fogel. *Intelligence through Simulated Evolution.* Wiley, New York, 1999.
12. J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Reading, MA, 1979.
13. K. Lindgren. Evolutionary phenomena in simple dynamics. In C.G. Langton, C. Taylor, J.D. Farmer, and S. Rasmussen, editors, *Artificial Life II, Vol. X*, pages 295–312. Addison-Wesley, 1991.

14. P. Maes, R.H. Guttman, and A.G. Moukas. Agents that buy and sell. *Communications of the ACM*, 42(3):81–91, 1999.
15. J.H. Miller. The coevolution of automata in the repeated prisoner's dilemma. *Journal of Economic Behavior and Organization*, 29(1):87–112, 1996.
16. M. Mitchell. *An Introduction to Genetic Algorithms.* The MIT Press, Cambridge, MA, 1996.
17. J.F. Nash. Two-person cooperative games. *Econometrica*, 21:128–140, 1953.
18. J. Paredis. Toward balanced coevolution. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo, and H.-P. Schwefel, editors, *Proceedings of the 6th Conference on Parallel Problem Solving from Nature – PPSN VI*, volume 1917 of *Lecture Notes in Computer Science*, pages 497–506. Springer, Berlin, 2000.
19. T.S. Ray. An approach to the synthesis of life. In C.G. Langton, C. Taylor, J.D. Farmer, and S. Rasmussen, editors, *Artificial Life II, Vol. X*, pages 371–408. Addison-Wesley, 1992.
20. C.D. Rosin and R.K. Belew. New methods for competitive coevolution. *Evolutionary Computation*, 5(1):1–29, 1997.
21. R.E. Smith, S. Forrest, and A.S. Perelson. Searching for diverse, cooperative populations with genetic algorithms. *Evolutionary Computation*, 1(2):127–149, 1993.
22. D.D.B. van Bragt, C.H.M. van Kemenade, and J.A. La Poutré. The influence of evolutionary selection schemes on the iterated prisoner's dilemma. *Computational Economics*, 17(2/3):253–263, 2001.
23. E. van Damme. *Stability and Perfection of Nash Equilibria.* Springer-Verlag, Berlin, 1991.
24. X. Yao and P.J. Darwen. An experimental study of N-person iterated prisoner's dilemma games. *Informatica*, 18:435–450, 1994.