A Machine Learning Approach to Automated Negotiation and Prospects for Electronic Commerce

Jim R. Oliver[1]

Abstract

We show how a system of artificial adaptive agents, using a genetic algorithm based learning technique, can learn strategies that enable it to effectively participate in stylized business negotiations. The negotiation policies learned are evaluated on several dimensions, including joint outcomes, nearness to the efficient frontier, and the similarity to outcomes of human negotiations. The results are promising for integrating such agents into practicable electronic commerce systems. We discuss what a system might look like and ways in which particular classes of business negotiations could be supported or even entirely automated.

1 Introduction

Even in simple negotiations, people often reach sub-optimal negotiations thereby "leaving money on the table" [1] [2]. While many factors contribute to missing out on gains from trade—overconfidence, falsely assuming fixed pies, and the framing of the situation (e.g. [3])—the end result is that parties fail to find agreements which would make each better off. This well documented fact has led researchers to develop tools to help people prepare for and participate in negotiations. This paper looks toward future electronic marketplaces and investigates not just supporting negotiators but also the possibility of fully automated business negotiations.

The challenge of negotiation arises, in part, from the fact that each side has private information about their own utility function, but is ignorant of the other's values and strategies. Exacerbating this situation is the incentive that negotiators have to misrepresent their preferences. Finding superior agreements in this dynamic environment of mutual mistrust is extremely challenging. Given the difficulty of the search, and the failings of humans at this task, wouldn't it be nice if an information system could search the possibilities effectively for us? We show how a system of autonomous agents can achieve this goal and learn effective negotiation strategies. Furthermore, based on our results, we argue that these agents should be integrable into practical electronic commerce systems which would not

1 Please use the following mail addresses:
Email: oliver27@opim.wharton.upenn.edu
Snail mail: 8013 Glenmore Spring Rd., Bethesda, MD 20817
Gab mail: 800-338-4079

only leave less money on the table, but would enable new types of transactions to be negotiated cost effectively, electronically, and automatically.

To clarify both the types of negotiations we consider and the opportunities for automated systems, we present a simple example. Although it is very specific, it illustrates many general issues. Consider the following scenario. A manager is leaving tomorrow morning on yet another sudden business trip. She has only visited this client once before and is not very familiar with her destination. She needs to reserve a hotel room—a different one from last time, as it was not satisfactory. Generally she prefers a hotel close to her client, so she has time both to work out in the morning, at a hotel gym, and grab a bite to eat before her meeting. Other amenities, such as room service, a pool, and laundry, are not as important. However, the cost of phone calls is of some interest as she dials in for both voice mail and e-mail. The only unusual aspect of this trip is there is a chance that she can finish her business in one day and leave in the evening, rather than staying in a hotel; a hotel with a liberal cancellation policy would be worth extra to her. Wouldn't it be nice if there were an automated system which could make a satisfactory reservation for her?

Now consider the same scenario from the point of view of a prospective hotel. Suppose this hotel is close to the manager's client. Suppose further that the hotel was recently renovated, including upgraded fitness facilities and installation of some minimal food service facilities that enables the hotel to inexpensively serve a continental breakfast. Also of note, the hotel is near an airport and often people arrive in the evening and want a hotel nearby, but haven't made any reservations. Thus, if the hotel is nearly booked with advanced reservations, a few no shows generally do not lead to any lost income. Wouldn't it be nice if there were an automated system that would help the hotel find the guests that value its characteristics the most?

Currently, this type of routine business transaction typically does not involve any negotiation between the parties. There are many reasons, ranging from convention, to agency issues, to the costs associated with negotiating with multiple parties over multiple issues. Ignoring the first two reasons, if negotiation were very cheap, then a dialog between the hotel and the customer would be valuable because options could be explored, in real-time, for mutual gain. The ability to create additional value is the essence of integrative bargaining (as opposed to the zero-sum distributive bargaining, such as dividing a dollar). An automated system could facilitate this value-creating dialog and therefore benefit all the parties involved. Furthermore, these automated assistants would be even more valuable if they could learn good negotiation strategies with minimal supervision. In sum, effective automated agents could not only help with existing negotiations, but new opportunities for more efficient exchange seem possible.

Although the hotel example is simple and specific, it has properties which are very general. As is usually the case, neither side knows the other's utility function: the prospective customer does not know the hotel's costs, and the hotel does not know how much the manager values each feature. While in this case, the electronic dialog would be about hotel rooms, it should be clear that this type of dialog generalizes to many other negotiations. That is, the promise of machine support is not a sophisticated hotel reservation system, but a general purpose system that creates value by making better deals for each party. Such a system would be especially valuable in any situation where the product offerings are very dynamic or substantially customized.

2 Research Goals

This stream of research is motivated by two broad questions:

> *Can automated agents learn strategies which enable them to effectively participate in typical, semi-structured, multi-issue, business negotiations?*
>
> *What is required and how does it work?*

These general questions beg many, more specific questions. While many avenues of inquiry could be undertaken, this paper focuses on just two aspects. We create a system of artificial adaptive agents (AAAs), test them in a variety of negotiation contexts, and evaluate their performance in two ways: 1) the extent to which they can learn to achieve effective outcomes for the specific games, and 2) their performance compared to published human data. These investigations are our initial, but necessarily incomplete, efforts at answering our open ended research questions. Based on what we learn, we propose a more complete system for automated negotiation in electronic commerce.

3 Background and literature review

We review four streams of research that inform the design of a system of automated negotiating agents. The first is game-theoretic models of bargaining and negotiation. Next are negotiation support systems (NSS) and distributed artificial intelligence (DAI), which address computer support of human agents and the design and study of distributed, computing agents respectively. Lastly we review evolutionary computation approaches to decision and search situations related to negotiation.

3.1 Game theory

The study of bargaining and negotiation has long attracted economists because it is fundamental to exchange and markets. Early foundations were laid by Nash [4, 5], and the area is still very active. Despite significant effort and progress, bargaining is incompletely understood; Radner and Satterthwaite [6] note that "adequate theories of bargaining exist only for the degenerate, polar cases of perfect competition and monopoly" (pg. 1).The following, more specific objections regarding game-theoretic models of bargaining are raised by Linhart and Radner [6]:

> *Common knowledge*. In particular, most models assume a common prior for the valuation of the negotiated object to the buyer and seller; yet, "ordinary experience seems to indicate that what makes horse races is variation among priors" ([6], pg. 216).

Multiple equilibria. In sealed-bid bargaining under uncertainty, there is a continuum of equilibria, even if one only considers pure strategies. Even in the simple case of complete information, two prominent axiomatic solutions, the Nash and the Kalai-Smorodinsky, can predict different outcomes.

Single object. In most models bargaining occurs over a single dimensions, such as price, but in real negotiations there are frequently many other issues such as quantity, quality parameters, delivery, and so forth.

These objections suggest that game-theoretic models of bargaining will be difficult to apply to natural situations. This point is also made by Raiffa [2] who notes, "...I never really used the techniques of game theory—concepts and ideas, yes, but techniques, no—in my roles as negotiator... The qualitative framework of thought was repeatedly helpful—not its detailed, esoteric, quantitative aspects. Simple back-of-the envelope analysis was all that seemed appropriate" (pg. 3). Although it is not yet clear the extent to which artificial agents can learn simple quantitative models that can be appropriately applied in new situations, à la back-of-the envelope, we are excited by our results in which automated agents discover rules of thumb for particular situations and which suggest that agents can appropriately modify these rules of thumb over time.

In sum, game-theoretic models have provided great insights into competitive decision-making, however they fall short of informing the specific design of computer models—in particular machine learning models—of negotiation. Put another way, game theory tells us about outcomes we can expect when rational agents bargain, whether these are artificial or not. The models do not tell, in all cases, a given agent which of many strategies to use in a given bargaining situation.

3.2 NSS

A disturbing research finding is the extent to which negotiators fail to reach the frontier of negotiation possibilities. In fact negotiators often leave money on the table, even in relatively simple negotiations. For example, in an experiment by Rangaswamy and Shell [7] only 4 of 34 dyads made a key integrative tradeoff. The experiment was a simulated international supply contract which had four negotiable issues, each issue having only four distinct options.

The challenges of negotiation and the shortcomings of human negotiators have prompted researchers to pursue computer support of negotiation, known generally as negotiation support systems (NSSs). Although NSSs typically emphasizes support, rather than automation, the implementations and the computational approaches they employ are relevant to and suggestive of possibilities for artificial agents. One example of the use of computational techniques is a concession model of Matwin, Szapiro, and Haigh [8], which hard codes a general strategy of concession in multi-issue negotiation. A very different NSS of Rangaswamy and Shell [7] employs a computer-based method to elicit a conjoint representation of preferences. Once the parties have a better understanding of their preferences, they make proposals electronically. In controlled experiments, the supported users reach better agreements. An additional feature of the system is it observes the offers made by each party and, by knowing the preferences of both, can suggest Pareto improving solutions. This feature raises two issues. One regards incentive compatibility and possible strategic behavior. Users who are aware that a computer will make suggestions based on the utility assessment might attempt to game the system by mis-representing their preferences. The second issue is satisfaction; users must be comfortable with a central system knowing their preferences and observing their offers. One can imagine a secure electronic marketplace that minimizes this issue, but some users might still be hesitant.

Negotiation support systems (NSSs) are designed to facilitate the various phases of the bargaining process. Because negotiations are considered complex and unstructured [9], NSS functional requirements have emphasized support capabilities which are very general; as such, these systems neither lend themselves to nor are intended to be fully automated. The tools for support are varied; many emphasize mathematical support tools, such as decision trees, forecasting, and so forth. However, Jelassi and Foroughi [10] have called for tools which address behavioral characteristics and cognitive perspectives of negotiators.

Woo [11] uses speech act theory to formalize the negotiation process so that machine transmission of messages in possible. Automation would result from combining this with the appropriate domain knowledge and other benefits could accrue from repetitive, similar negotiations.

Although NSS research has different goals from this research, we share some of its prescriptive aims and the area offers important guidance and ideas for more automated negotiators, particularly in the areas of system architecture, functional requirements, and user interface.

3.3 DAI

Like NSSs, distributed artificial intelligence (DAI) systems provide examples of computational approaches to decision making. Bond and Gasser [12] characterize the scope of DAI as considering "how the work of solving a particular problem can be divided among a number of modules ... that cooperate at the level of dividing and sharing knowledge about the problem and about the developing solution" ([12], pg. 3). The challenges are to coordinate the modules, with limited communication, in the face of possibly inconsistent knowledge. Most DAI research has assumed cooperation, or "collaborative reasoning"; conflicts have been limited to issues such as bidding for shared resources, typically computation. This non-strategic, primarily cooperative approach, typical of most DAI research, clearly can not be applied to all (competitive) business situations. However, recent research has begun to explore less cooperative paradigms and is more promising for strategic interaction (see, e.g. Rosenchein and Zlotkin [13]).

3.4 Evolutionary Computation

The most immediately relevant stream of research explicitly investigates machine models of competitive situations by using the techniques of evolutionary computation in systems of artificial agents. Genetic algorithms (GAs) are probably the most common evolutionary technique. We describe the algorithm in more detail in §4, but briefly GAs are techniques inspired by evolution, in particular the concepts of variation and natural selection. In an optimization context, a population of candidate solutions is generated and evaluated; the best solutions are assigned the highest fitness and preferentially chosen to be "parents" and combined to create new candidate solutions that comprise the next generation. These "children" are just new possible solutions, which are evaluated and the cycle continues.

Using an evolutionary computation approach called genetic programming, Dworman, Kimbrough, and Laing [14] offer support for the idea that autonomous learning agents can discover particular and attractive equilibria in certain classes of games. Another example is that of Marimon, McGrattan, and Sargent [15] who study a simple exchange economy in which agents must use a commodity or fiat money as a medium of exchange if trade is to occur. The artificially intelligent agents are modeled using classifier systems to make decisions. For most of the economies simulated, trading and consumption patterns converge to a stationary Nash equilibrium even if agents start with random rules. The simulations show that multiagent systems of classifiers can eventually learn to play Nash-Markov equilibria.

There are many other cases of evolutionary algorithms (EAs) being applied to decision situations, such as the prisoners' dilemma by Ho [16], Miller [17] and Axelrod [18], sequential decisions by Oliver [19], and double auctions by Rust, Miller, and Palmer [20]. This broad stream of literature, including much else not mentioned, suggests that the goal of practical, automated negotiating agents is ambitious but attainable.

4 Implementation

The success of evolutionary algorithms in diverse domains, but especially in decision and search problems [21], make them an appropriate machine learning approach for automated agents discovering effective negotiation strategies. In our first phase of research, which is reported here, we use only genetic algorithms (GAs). We overview GAs here, but an excellent introduction is Goldberg [21].

4.1 Genetic Algorithms

In a GA, candidate solutions to the problem are encoded into "chromosomes," which are a representation of a solution or instance of the problem at hand. While there are no hard and fast rules of representation, a specific encoding of a problem into binary strings is often done. The GA then operates on the binary (base-2) string analogously to the way genetic processes operate on our own base-4 chromosomes. While a binary coding is common, this is not required by the GA.

The basic GA begins with a randomly generated population of candidate solutions. We describe here the standard binary case; the non-binary case is analogous. The population is the set of chromosomes, which begin as a random set of ones and zeroes. Typically, each chromosome is evaluated. For example, in the case of maximization, the chromosome is the input to the objective function, and the fitness of the chromosomes could be the value of the objective function.

A new population is created by selecting individuals to be parents for the new population. The basic selection strategy is to choose parents proportional to their fitness. Thus an individual chromosome that has twice the fitness of another has twice the chance of being a parent for the new population. Sometimes results are improved by scaling the fitness according to a non-linear function before the selection routine is performed.

The selected parents are used to create the next generation of the population. While in some cases, new parents are simply preserved in the next generation, new parts of the problem space are explored by creating new chromosomes. The evolutionary inspired operators of crossover and mutation are most commonly used. Single point crossover is one particular approach and works as follows. Assuming the sixth position is the randomly chosen crossover point, then one child is composed of the first six bits of one parent and bits after position six from the other parent. This method of crossover can be used to create two offspring from each parent. The other child gets the first six bits from the second parent and the final bits from the first.

The other primary mechanism of generating variance in the population is mutation. The following is an example of mutation. If the fifth bit of a child mutates, it changes from a zero to a one or from a one to a zero. Generally every bit in a chromosome has a small chance of mutating. The probability of a mutation occurring for any given bit is controlled by a system parameter called the mutation rate. Thus the number of mutations per chromosome depends on the length of the chromosome and the mutation rate.

After the new population is created, the cycle begins again. Each new chromosome is evaluated for fitness, a new population is created and so forth. This loop is repeated until a specified stop condition is met. One that is often used is a stable average fitness. If the fitness of the population has remained rather stable for a number of generations, then continuing for more generations is not likely to yield better results. Experience with GAs and understanding the problem being solved are a significant help to creating appropriate stopping conditions.

4.2 Negotiation as Search

Negotiation is a search process. For example, two player, integrative bargaining can be viewed as two negotiators jointly searching a multi-dimensional space and then agreeing to a single point in the space. Each dimension can be discrete or real valued, although in our implementation, only discrete values are used. Each party has a multi-attribute value function over the space of possible agreement points. In the bargaining space, each dimension corresponds to an issue to be negotiated; each issue has two or more alternatives which are indexed by elements of a set. For illustration, consider a stylized business negotiation. Suppose a purchasing agent needs to obtain quickly a particular part from a supplier. The agent might be interested in the issues of price, quantity, and delivery. With regard to delivery, the buyer might have decreasing utilities for next morning, next afternoon, 2nd day air, and (several day) ground transport. For the price dimension, the part-worth would be monotonically decreasing in price. The part-worth for quantity might be an ideal point model. In contrast to the purchaser's values, a supplier would have different, partially opposing, utilities. The space of bargaining outcomes is {price in [0, P_{max}]} X {quantity in

[0, Q_{max}]}X {delivery from {next morning, next afternoon, 2nd day, ground}}, although some points might not be feasible.

4.3 Value Functions

In simulations reported here, simple, additive preference models are typically employed; the part-worths for each alternative are typically independent, although specific interactions are accommodated in some cases. The part-worths are implemented as a simple table look up. The set corresponding to a dimension is arbitrarily indexed and a corresponding payoff set is encoded in the system. In our example, the set for the delivery dimension is {ground, 2 day, next day afternoon, next day morning} which has payoffs of, perhaps, {0, 7, 8, 10} to the customer and {10, 2, 1, 0} to the supplier (the maximum part-worth of 10 is arbitrary). The part-worths are implemented with a non-associative array, i.e. delivery [1] = ground, and payoff_customer[1,1] = 0. In general, the value, V, of a particular option $X = (x_1, x_2, \ldots, x_n)$ is,

$$V(X) = w_i \, v_i(x_i)$$

where v_i is the part-worth function for the alternatives for issue i, and w_i is a weighting factor that may or may not be necessary depending on the scaling of the part-worths. Using the previous example, the utility of the outcome (price = \$2.75, qty = 150, delivery = next morning) is V(\$2.75, 150, next morning) = v_1(\$2.75) + v_2(150) + v_3(next morning).

4.4 Strategies

A set of random, feasible, initial strategies are created. The strategies currently in use by the system are sets of simple, sequential, threshold rules. The rules are meant to be intuitive, straightforward to explicate, and not too difficult to elicit. An example of the type of strategy we use is the following: consider the seller of an item, such as a car. The strategy might be to initially, accept any offer whose value is greater than a threshold T_1. If the prospective buyer's offer does not meet that threshold, then make a counter offer. If this counter offer is not accepted, and the buyer comes back with a new offer, the seller will compare this with another, typically different, threshold, T_2. Again, if the threshold is not met, a counter offer could be made. At any point, either party might choose to discontinue bargaining. This type of rule structure can be extended for an arbitrary number of rounds, but for practical purposes the real strategies are limited in depth. The current system only uses strategies with this simple structure, but more general strategies could be used.

4.5 Messages

Offers made by each party are communicated via messages, which are sent privately to the other party. This approach can be extended to more than two parties, so as to handle the cases of multi-party negotiation or sealed bid auctions, in future work. In posted price auctions, the messages could be broadcasted, or posted electronically, to all participants.

4.6 Operation

Figure 4.1 summarizes system operation. Each of the two artificial players is initialized with a set of random strategies, and each randomly chooses a strategy to test. Within the bargaining cycle, one player is randomly chosen to start the bargaining; this player sends an offer message. The other player's strategy evaluates the message and either accepts the offer or makes a counter-offer. This continues until agreement is reached or one of the players exhausts the strategy being tested and sends a quit message. The system continues to test strategies in this manner, for a specified (a system parameter) number of times. Then the genetic algorithm is run and new a population of strategies is created. This outer loop is also continued until an exit condition is met.

4.7 Comments

Earlier we cited three limitations of typical game theoretic models of bargaining. The bargaining games played by our artificial agents are designed with explicit awareness of these limitations. The negotiations are not over a single object, but like most business negotiations are inherently multi-dimensional. All games have multiple equilibria and which is selected is specifically explored. Lastly, the usual assumption of common knowledge is not required in this computational model: the artificial agents, are initially devoid of any explicit knowledge about other agents, and they do not even explicitly know their own payoff function, which is provided by nature. This information structure is in the spirit of Young [22] (p. 5) who notes: "Typically, the parties do not know each other's utility functions with any degree of accuracy... Usually they do not know each other's BATNAs [best alternative to a negotiated agreement]." However, each agent does have a strategy, and the agent's population has a distribution of strategies, which changes over time as strategies co-evolve. Natural selection shapes this distribution by culling strategies that are less useful given the current distribution of opposing strategies.

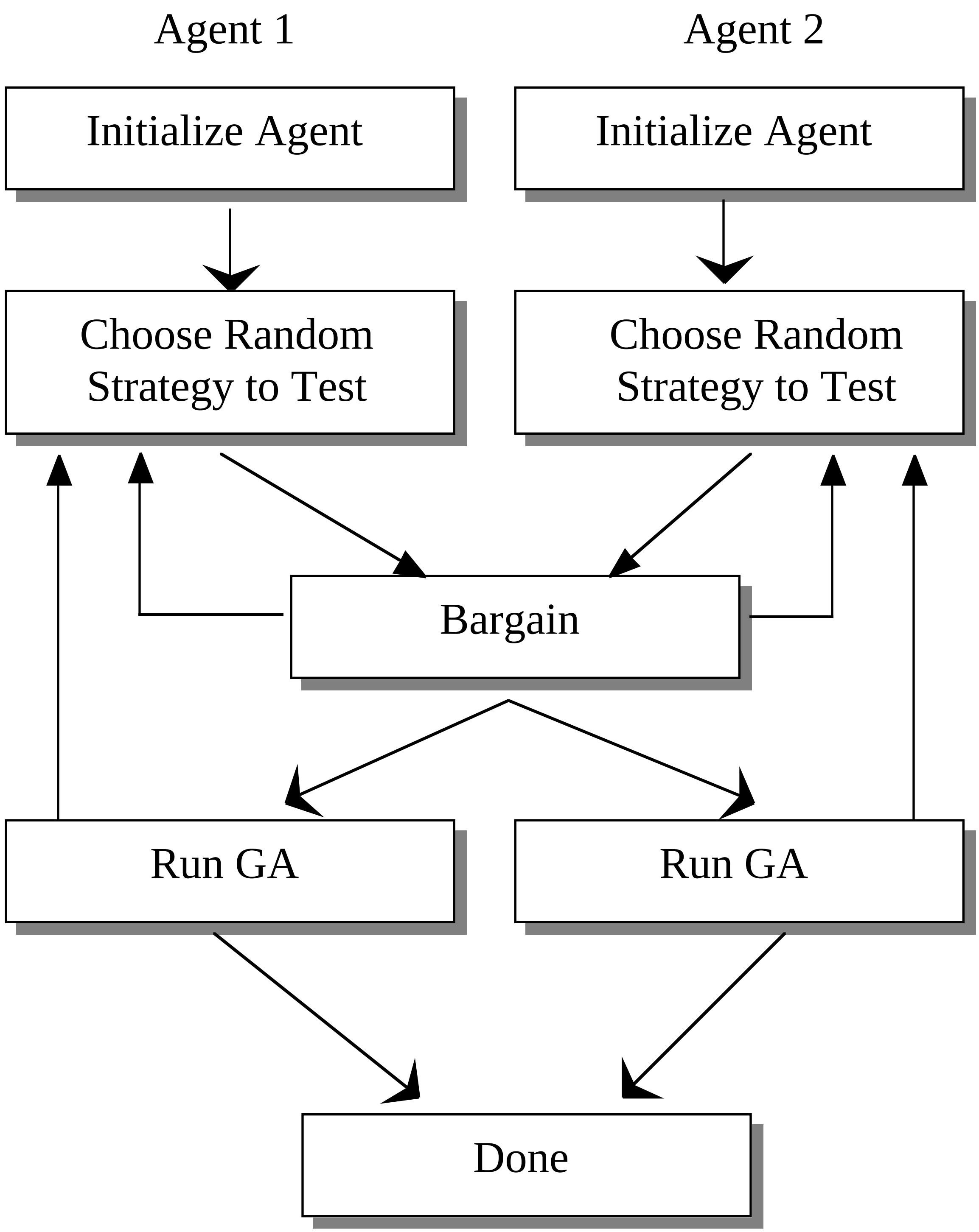


Figure 4.1. System operation.

5 Experiments and Results

We investigate the performance of the AAAs across five types of games. We operationalize performance with several standard performance measures, as described below. In addition, each particular run requires specific system parameter values, also described below.

5.1 Game types

The first of the five game types, called no conflict (§5.4), has no competitive aspect, but provides a useful benchmark. The next focuses on a pure distributive bargaining problem (§5.5) in a reduced-dimension environment. The third is a simplified integrative bargaining problem (§5.6), a low dimension, stylized business negotiation. The fourth (§5.7) involves a larger dimensionality example from Raiffa (1984), for which human results are published; the outcomes of artificial, adaptive agents compares favorably with that of the human subjects in this stylized, but fairly realistic, labor negotiation. The fifth (§5.8) is a stylized international business negotiation by Rangaswamy and Shell [7].

5.2 Performance Measures

Several statistics and measures are tracked regarding the performance of individual chromosomes (strategies), the population as a whole, and features that characterize the bargaining sessions.

The most important measures characterize the performance of individual agents and the dyads. AAAs should achieve high payoff outcomes individually. We measure the individual payoffs in terms of the agents' endowed value functions, which we know precisely; this information enables statistical tests of agent learning by comparing individual payoffs achieved after a training period with (1) agreements in the first generation that arise from agreements by the random strategies the agents are initially endowed with, and (2) the expected payoff to each agent of a randomly selected point in the bargaining space.

Agents should not only achieve excellent individual payoffs, but they should achieve excellent outcomes collectively as well. Following Bottom and Studt [23] and Foroughi, Perkins, and Jelassi [24] we measure joint payoffs as the sum of the individual payoffs. We use the same statistical tests as above to compare the joint payoff with agreements in the first generation of bargaining, and the expected payoff from a random point in the feasible space.

Human negotiators often choose agreements that are below the pareto frontier. We compute nearness to the frontier as follows. For each agreement, a list of all pareto superior agreements on the frontier, is generated, and the improvement in payoffs, for each agent, associated with these points is calculated and averaged. The averaging is both over the set of dominating points and over all the agreements in the generation. This inferiority measure has two benefits. Besides being more encompassing than just Euclidean distance to frontier, keeping the measures independent for each agent means that no interpersonal tradeoff measures are made.

5.3 System parameters

Each run of the system requires several parameters to be specified as follows.

Population size This is set to 20. These 20 chromosomes make up the strategy set for each agent. Each chromosome is a threshold decision rule that can begin and end a single bargaining session. Twenty was chosen as a small population size, yet one that consistently yielded results. Pilot tests with smaller population sizes were not consistent in performance. Optimizing the population size is not simple, and is not the goal of this work.

Number of sessions This is set to 20. Each generation consists of 20 bargaining games. Because there are 20 chromosomes per agent, each chromosome will participate, on average, in one bargaining game per generation. The strategies for testing are selected randomly (uniformly), with replacement, so typically, in each generation, some strategies are tested more that once and others not at all. The random selection was used to insure that no cycles are created in which, for example, strategy one of agent one always plays the same strategy of agent two.

Number of generations This is set to 20. Each run consists of 20 generations. This was adequate time for the agents to learn reasonable strategies.

Crossover rate This is set to 0.5, which favors neither parent in crossover. Each part of a child's chromosome is equally likely to come from either parent.

Mutation rate This is set to 0.05, which implies that, on average, one in 20 elements of the offspring will be different from both parents. Mutation occurs on individual threshold values and on individual components of the offer vectors. This mutation rate is perhaps a little higher than some GA applications, but it is not unusual either. Like population size, this parameter was chosen based on published GA applications and on brief pilot studies; optimizing this system parameter is left to later research.

Number of offers This is set to 3 for the experiments reported here. The value of 3 allows each side to make up to 3 offers before the game is terminated, assuming neither side has agreed. This parameter, like other individual parameters, could be asymmetric between agents, but this is not explored in this research.

5.4 Experiment 1: system verification

For simply verifying the operation of the system, the artificial agents are endowed with the same value functions. Consequently, the agents should converge to the unique, joint maximum, i.e. the unique pareto optimum. The absence of conflicting interests makes this test essentially two, interrelated, discrete optimizations — the difficulties of competitive, co-evolution are deliberately removed.

Agent value functions.

Figure 5.1 shows the value functions for the agents for six alternatives, one through six, for each of two issues, A and B. Both parties are happiest with the option (A=6, B=1).

Results.

Figure 5.2 shows the results for a single population pair (each with 20 chromosomes, or strategies), initially endowed with random strategies, after 20 generations. The smallest dots, on the line, are the possible agreement points, and the slightly larger dots are noisy plots of the actual agreements. The noise is intentionally added to clearly show multiple agreements at the same point. The pareto optimum is in the upper right corner. In this run, most all of the agreements are the maximum possible, except one which is in the center of the graph. This poor agreement could have resulted from a mutated strategy.

Figure 5.3 shows another run in which the average payoff is actually similar to that in Figure 5.2, but the population converged early, prior to reaching the maximum. Most of the agreements are just below the maximum in the Northeast corner. Early convergence is a general problem of GA based approaches. While there is no absolute solution, there are strategies that help to minimize it [21], but these are not the focus of this investigation.

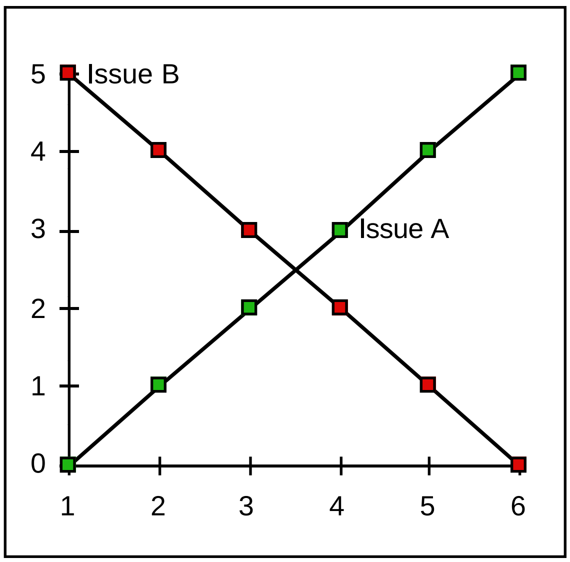


Figure 5.1. Value (y-axis) for each agent for two issues. Each issue has six alternatives (x-axis).

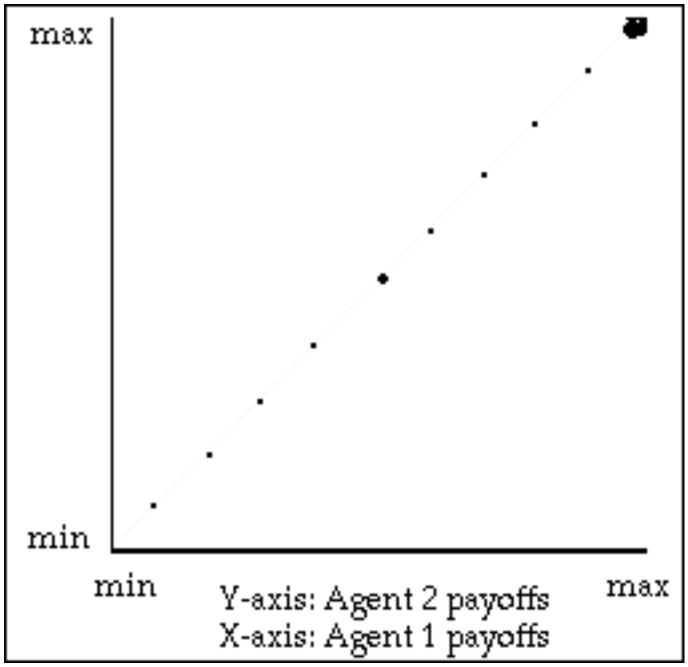


Figure 5.2. One run of the no conflict game.

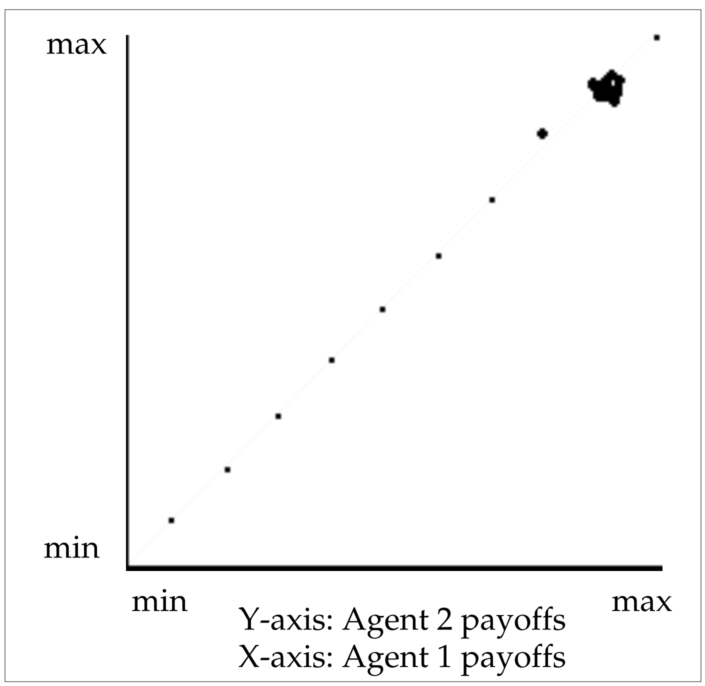


Figure 5.3. Another run of the no conflict game.

Figures 5.2 and 5.3 visually show bargaining agents agreeing to points that are much better than if they were just agreeing to random points in the bargaining space. To test more formally whether the agents are learning, we use several statistical tests, which are summarized below.

Individual learning. We perform a one-sided *t*-test comparing the payoff to each agent in the first five generations, averaged over ten runs, with those of the last five generations, assuming a pooled variance.

Individual comparison with a random point. We perform a one-sided *t*-test of the hypotheses that the last generation payoffs are greater than the expected value of a random point in the bargaining space.

Dyad learning. We perform a one-sided *t*-test for the hypotheses that the dyad learns from the first five to the last five generations, as measured by the joint payoff.

Dyad comparison with a random point. We perform a one-sided *t*-test of the hypotheses that the joint payoffs for the dyad in the last generation are greater than the expected value of a random point in the bargaining space.

Nearness to the frontier. We perform a one-sided *t*-test comparing the nearness to the frontier for each agent in the first five generations with the nearness in the last five generations, assuming a pooled variance.

Table 5.1 summarizes the results of the *t*-tests. The agents exhibit significant learning behavior; all of the hypotheses were strongly supported.

Test	P-Value
Individual learning (Agent 1)	6.5×10^{-5}
Individual learning (Agent 2)	6.5×10^{-5}
Individual payoff better than random (Agent 1, expected random payoff = .5)	1.1×10^{-7}
Individual payoff better than random (Agent 2, expected random payoff = .5)	1.1×10^{-7}
Dyad learning	6.4×10^{-5}
Dyad payoff better than random (Expected Joint Payoff From Random Agreement = 1.0)	1.1×10^{-7}
Nearness to frontier (Agent 1)	6.3×10^{-6}
Nearness to Frontier (Agent 2)	6.3×10^{-6}

Figure 5.1. Results of statistical tests for the no conflict game.

5.5 Experiment 2: pure distributive bargaining

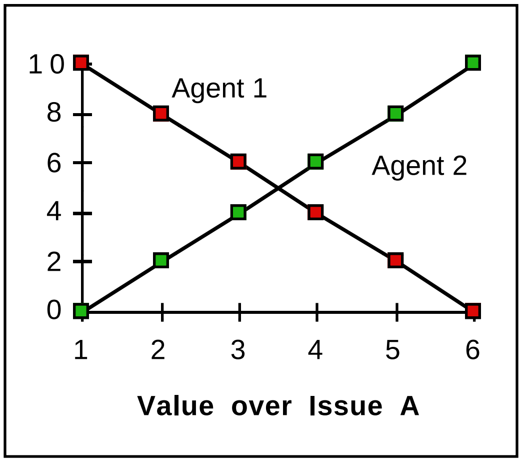
<u>Agent value functions.</u>

Figures 5.4 and 5.5 show value functions for the agents for six alternatives, one through six, for each of two issues, A and B. Player 1 prefers (1,1), i.e. alternative one on issues A and B, while player 2 wants (6,6). Like games in later sections, this game has multiple Nash equilibria; however, unlike most of the games reported later, there is no

prominent equilibria — the frontier is flat as shown in Figure 5.6. While cooperative game theory deals explicitly with such situations, classical non-cooperative game theory does not predict which equilibrium will be selected.

Results.

Figure 5.7 shows the results for a single run. In this case, as in other runs of the same game, there is some clumping of the agreements, due to convergence of the population of strategies. The particular location of convergence tended to vary and we defer further investigation of pure distributive bargaining to future research; the focus in this paper is on integrative bargaining because of the opportunity for mutual gain and the relevance to typical, commercial exchange situations.



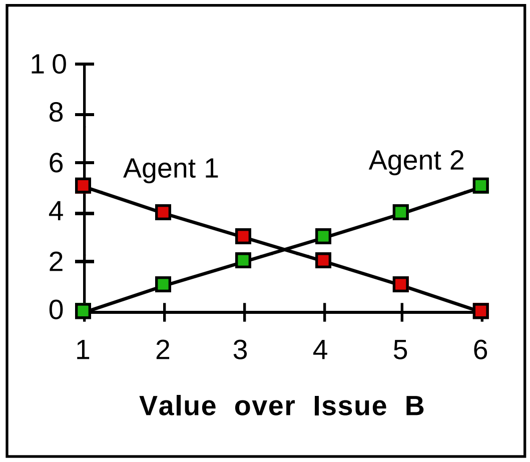
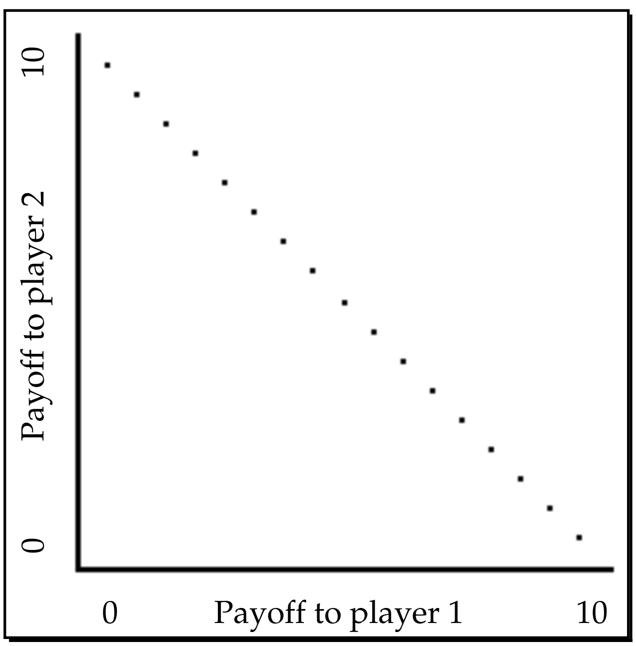


Figure 5.4. Values for six issue A options. Figure 5.5. Values for six issue B options.



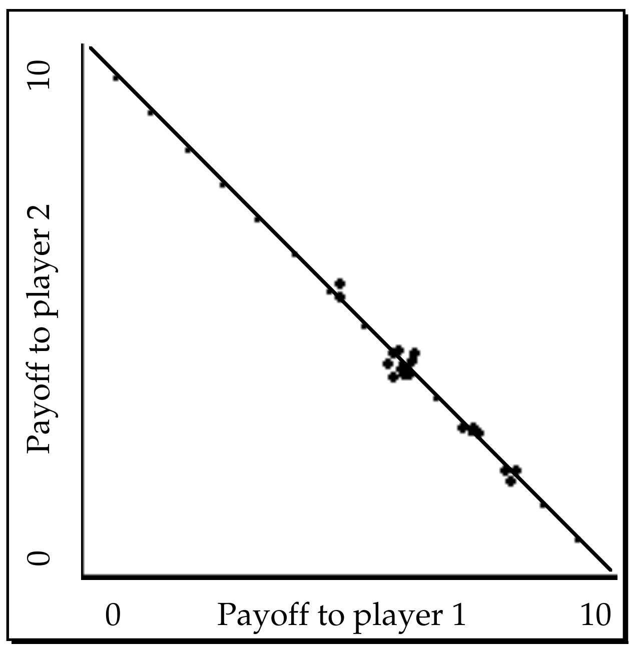


Figure 5.6. Feasible payoffs graph. Figure 5.7. Population after 20 generations.

No statistical results are given because the nature of the game makes all measures meaningless: all agreements are always on the frontier, and all joint values are the same. The only test of interest would be individual tests to see if either agent is consistently outperforming the other. Such a result would suggest an error in the system as our agents are symmetric. No significant differences were found.

5.6 Experiment 3: simple integrative bargaining

This game is a stylized business negotiation that captures a simplified purchasing situation in which the agents bargain over price, quantity, and delivery. The agents have opposing value functions within each issue, but the importance across issues varies such that rather than compromising on each issue, each player should give up everything on the issue of least importance and get the maximum on the issue of greater importance.

Agent Value Functions.

Table 5.2 shows the value functions for the agents. For agent 1, the most important issue is price, whereas for agent 2 the most important issue is quantity. The expected joint payoff, the sum of the both agent's payoffs, from a random point in the bargaining space is 0.98. The joint maximum is 1.36.

Results.

An example of AAA agreements and the entire bargaining space is shown in figure 5.8. The large dots are the agreement points for the final generation of one run of bargaining, and the smaller dots are all the possible agreements. The larger dots have noise added intentionally to clarify the number of agreements at a discrete point. In the figure, the average payoff to agent 1 is 0.66 and to agent 2 is 0.53, for a joint payoff of 1.18.

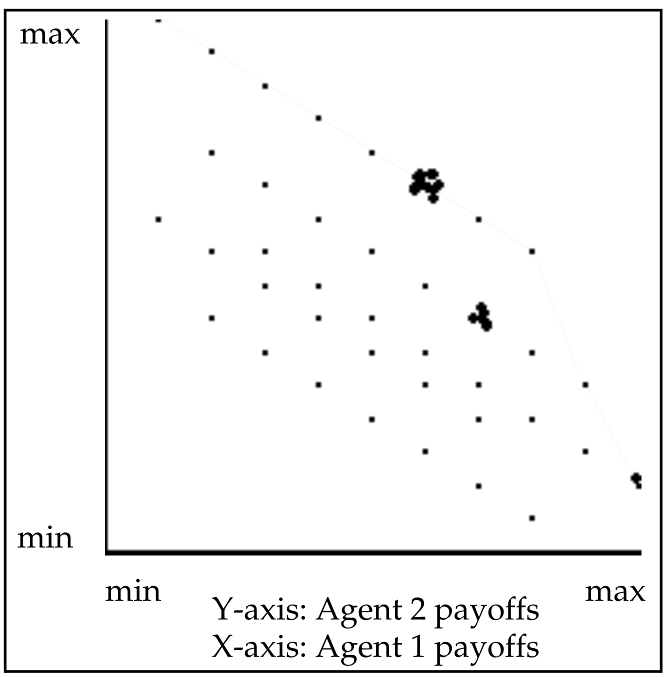


Figure 5.8. Results for one run of the integrative game.

		Values	
Issues	**Alternatives**	**Agent 1**	**Agent 2**
Price	low	0	.25
		.11	.19
		.22	.12
		.33	.06
	high	.44	0
Quantity	few	.22	0
		.17	.12
		.11	.25
		.06	.38
	many	0	.56
Delivery	next morning	.33	0
	next afternoon	.22	.06
	2nd day	.11	.12
	7-10 days	0	.19

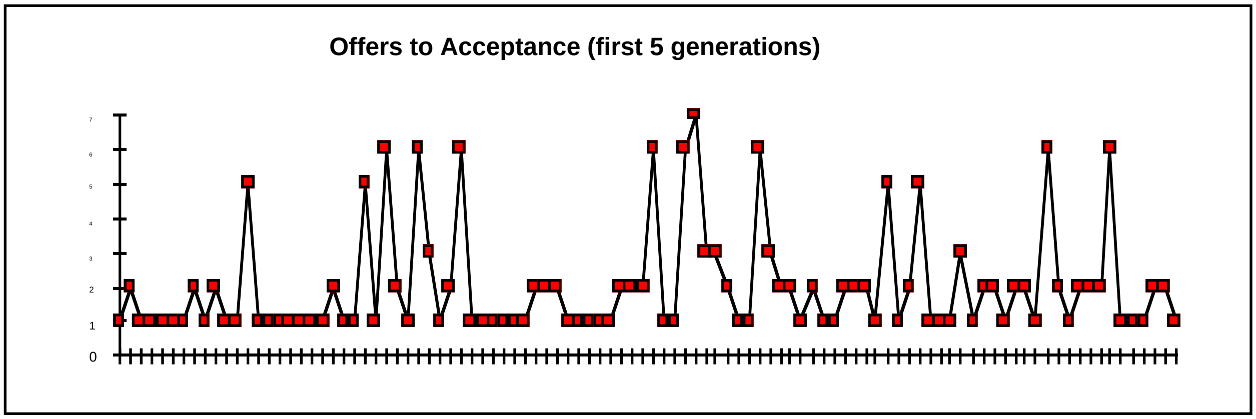
Table 5.2. Value over three issues, price, quality, and delivery, for two players.

The same statistical tests, show in Table 5.3, are applied to the agents as in the first game. On balance, the agents have no problem learning in this simplified business negotiation context.

Test	P-Value
Individual learning (Agent 1)	0.00066
Individual learning (Agent 2)	0.00014
Individual comparison with a random point (Agent 1, expected random payoff = .51)	0.014
Individual comparison with a random point (Agent 2, expected random payoff = .47)	0.056
Dyad learning	1.8×10^{-9}
Dyad comparison with a random point (expected joint payoff from random agreement = 0.98)	0.00047
Nearness to Frontier (Agent 1)	6.4×10^{-8}
Nearness to Frontier (Agent 2)	2.4×10^{-8}

Table 5.3. Statistical test results for integrative game.

The GA methodology facilitates a detailed analysis of the evolution of high performing strategies. Figure 5.9 shows that, agents learn not to accept an offer too soon because they might come across a better offer later. The strategy cannot wait too long as the bargaining will be cut off. The top of Figure 5.9 shows the population starts out by accepting many initial offers; however in the last five generations (out of 20), many agents do not accept until the middle of the bargaining session.



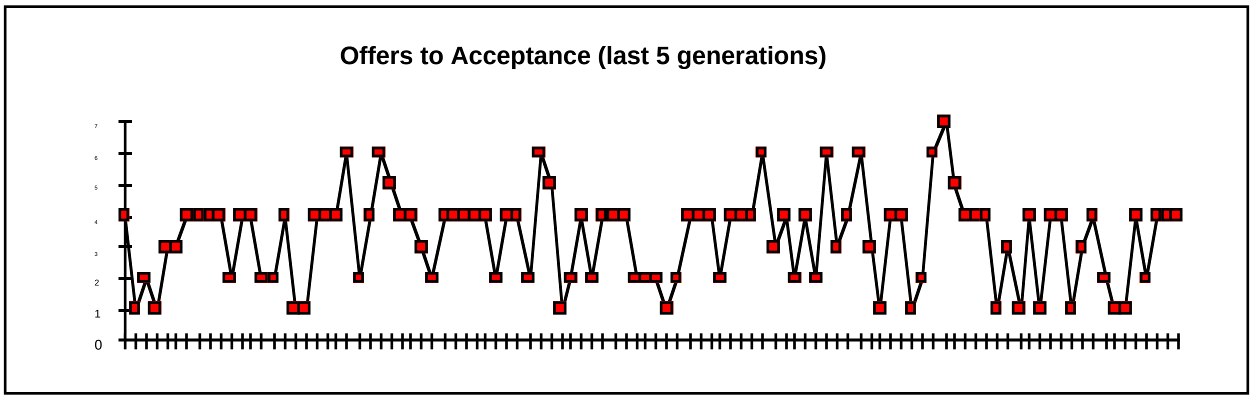


Figure 5.9. Offers to acceptance early in the run (top) and late in the run (bottom).

5.7 Experiment 4: City vs. AMPO

The negotiation games reported so far have been contrived and stylized, both for simplicity and to expose the operation of the automated agents in a basic but crisp manner. Real negotiations, even simple ones, are more complicated than those of the previous sections. One important factor is complexity, especially in terms of dimensionality; high dimensionality creates a challenging negotiation environment and human subjects are more likely to accept inferior offers.

Agent value functions.

In this section, we recreate an experimental game reported in [2]. It is a stylized labor negotiation between an city and a police union, called AMPO. The size of the bargaining space, as implemented, is 13,219,200 possible points of agreement. There are 11 dimensions, each with anywhere from 2 to about a dozen alternatives. The original problem had one dimension that had a (nearly) continuous interval -- the payoff was linear over a salary increase range of 500 to 750 dollars. Given that fractional amounts were unlikely, this effectively made the dimension discrete, but we more coarsely quantized the entire dimension into 12 points. As in other games, we scale the payoffs to the range of 0 to 1.

Results.

The result of one run is shown below in Figure 5.10. A visual comparison of the agreements reached by human agents versus agreements reached by the artificial agents shows no performance differences. Both the humans and the AAAs occasionally reach agreements very close to the frontier, but typcially both were a similar distance away[2], although the humans varied more than the AAAs. The results of a longer series of ten runs suggest that neither group outperforms the other, but there were some differences. The humans representing the City, on average, achieved higher average payoffs for themselves than did the AAAs, 0.70 versus 0.65. A t-test of the average AAA payoffs with the mean payoff for the humans representing the city had a *p*-value of 0.06. Opposite results held for the agents representing AMPO. In this case, the AAAs outperformed the humans, earning mean payoffs of 0.58 versus 0.53. This difference was mildy significant ($p = 0.04$). Although these performance differences are interesting and deserve further investigation, a more important comparison is the joint payoff; on average the joint payoff was the same for the human agents and the AAAs, 1.23. Thus neither group outperformed the other.

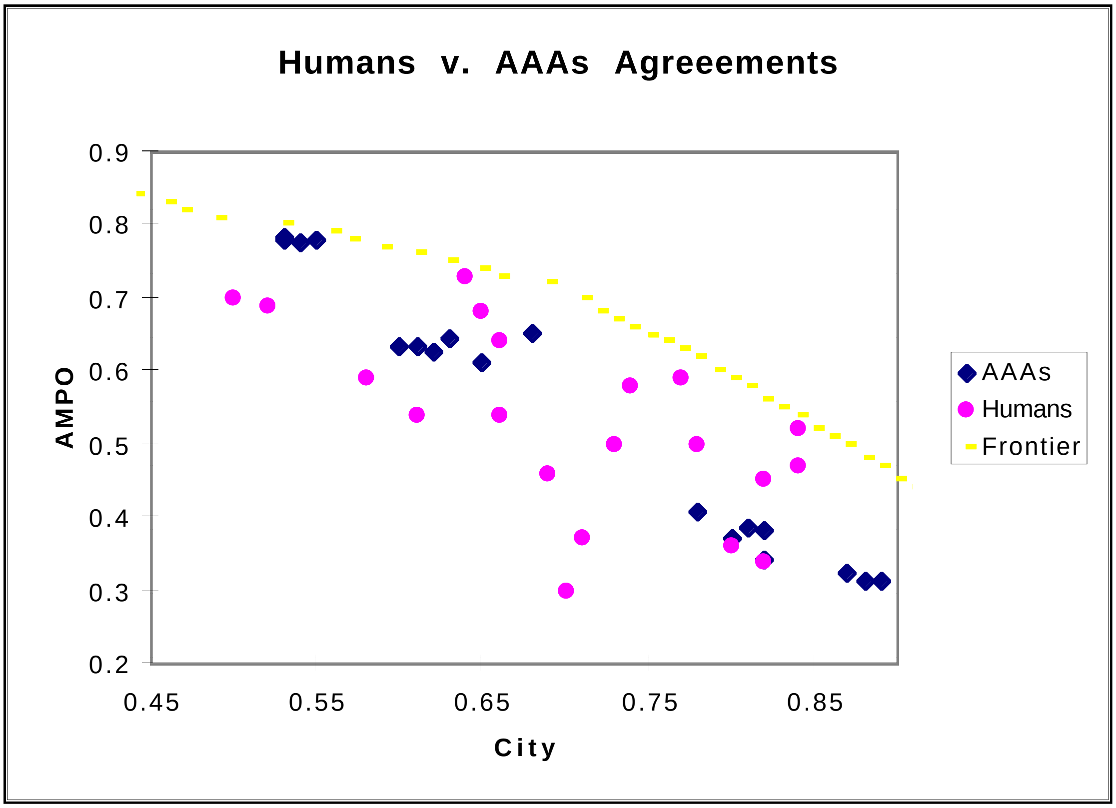


Figure 5.10. Human agreements and AAA agreements for one run. Neither outperforms the other.

2 Two features of Figure 5.10 deserve mention. First, the scales do not cover the entire payoff range of zero to one, which highlights differences in agreements. Second, only some of the frontier points are shown, for clarity.

5.8 Experiment 5: Stylized International Business Negotiation

Rangaswamy and Shell [7], report on a four dimensional, stylized international business negotiation. The game only has 256 discrete, possible points of agreement, but unaided humans were not effective at making integrative tradeoffs: only 4 out of 34 pairs did so in a series of experiments. Using an NSS of their design, the human subjects in the assisted case performed significantly better, but even in this case fewer than 50% [7] achieved the main integrative solution. This game has the characteristic that only ordinal preferences are induced in the laboratory. This is recreated for the AAAs by having the agents play against multiple instances of value functions which fit the ordinal preferences.

Agent Value Functions.

The experiment is a simulated international business negotiation between an American healthcare company and an Eastern European medical equipment supplier. There are four issues to be negotiated: price, delivery, the currency of payment, and the location to adjudicate disputes. The options for each issue are,

Price:	180, 195, 210, 225 K$
Delivery:	6, 8, 12, 14 months
Currency:	US$, Euro$, other hard, Hungarian
Dispute:	US, London, ICC, Hungary

A key element of the preference structures in this game is the built in integrative tradeoff of Hungarian currency and 14 month delivery. This tradeoff is the one that humans have difficulty finding, but is the single most valuable.

Results.

The human results reported by Rangaswamy and Shell [7] are measures of the frequency that key integrative tradeoffs are made. Table 5.4 shows the frequency of key integrative tradeoffs, in the international game, for humans and AAAs. Recall that the primary integrative tradeoff is Hungarian currency and 14 month delivery. A 12 month delivery is the second most preferred alternative, on the most important issue, for the supplier. Similarly, European currency is the second most preferred alternative, on the most important issue, for the customer. While the Hungarian-14 month combination is the most integrative, the others are also beneficial, those shown in table 5.4 are the ones reported by Rangaswamy and Shell [7]. The AAAs results are summed for 5 types of customers. The results of the experiments are that the AAAs learn to achieve outcomes that are in between the unaided and the aided humans. In general, while the AAAs found the best integrative tradeoff frequently, they settled on many more "other" agreements, as opposed to the two nearly as good tradeoffs, when compared with the human case. Presumably this result is due, in part, to that fact that the GA implemented for this system is very simple. Manipulation of population size, different crossover strategies, diversity management, and other techniques could probably improve performance at least slightly, and possibly substantially.

	Human		Artificial Adaptive Agents
	Face-to-Face (Unaided)	Using Negotiation Analyst	Total for 5 customer types
Hung—14 months	4	15	358
Hung—12 months	9	3	63
Euro—12 months	15	6	22
Other	6	8	464
No Agreement	0	2	93
Total	34	34	1000

Table 5.4. Frequency of integrative tradeoffs made in the international business negotiation.

5.9 Relevance and Discussion

While the results presented here are preliminary, these initial investigations strongly suggest that artificial agents can learn effective strategies for specific negotiation games. As argued before, given the difficulties humans have at low dimensionality negotiation games, it is not obvious that cognitively simple artificial agents would be successful at such a task. The incomplete information environment that is challenging for humans is, of course, similarly challenging for artificial agents. Further hindering the agents, the genetic learning approach operates in a dynamic, co-evolutionary environment, rather than the static world of a traditional optimization3. We submit that the ability of such simple agents to learn the same games that humans find challenging is promising on many fronts, but especially for electronic commerce.

The performance of artificial agents compares quite favorably with that of human subjects. The performance of AAAs certainly does not dominate the human case, rather AAAs sometimes performed better and sometimes performed worse, when measured in terms of ex-post efficiency and ability to make integrative tradeoffs. The circumstances in which one outperforms the other is an important area of future research, but we submit that the current successes of the AAAs are encouraging, exciting, and promising.

6 Architecture of a System for Electronic Commerce

We have argued that the results presented here are promising, but neither the system of AAAs developed nor the studies undertaken, are the same as fielding a practical system. However, the results of the previous section inform

3 There is an important distinction between this game environment and the application of GAs to optimization. In optimization, there are no external dynamics for the GA, i.e. typically the environment does not change. Thus a given chromosome always has the same raw fitness (algorithms to adjusts the raw fitness based on population characteristics might be used for the purpose of encouraging population diversity). The competitive environment of coevolution is fundamentally different and more difficult.

the design of a more complete system. In this section, we briefly consider what it would take to get this ability into practice, how it might work, and what would have to be learned first.

6.1 System Outline

Creating a working system for electronic commerce needs to address user, marketplace, and functionality issues that we have, until now, ignored. Figure 6 outlines a possible system architecture. We discuss each element of the system in turn.

6.2 User Interface

This is point of interaction with the following system functions,

1. eliciting preferences from user,
2. eliciting, browsing and manipulation of strategies,
3. updating the database of products and services, their negotiable issues, and so forth,
4. reporting.

Points (1), (2), and (3) are described below. The reporting function (4) is primarily intended to inform a manager or principal as to how the agents are performing. The average length of negotiations, changes over time, and profitability of the agents are just three aspects of bargaining behavior that might be valuable for management.

6.3 Preferences

Until now, the precise objects of negotiation have been abstract and stylized. But a practical system needs to buy and sell real products and services. This requires the system to know about the specific attributes of the products of interest, information that can be stored in a local or remote database.

The utility functions for these products need to be elicited, and these functions will be stored, most likely, locally—this is the type of information people want the ability to keep private. Many researchers, as well as commercial concerns, have developed viable computer-based approaches to utility function elicitation. Rangaswamy and Shell [7] report on one such elicitation procedure that could be used.

If new products, or new features for an existing product, become available, both the new attribute data and the user's value over that attribute need to be incorporated into the system. The PC industry offers two examples. First, manufacturers have recently started to bundle modems into new PCs, rather than the consumer purchasing these separately. If the consumer's DSS already knows about both PCs and modems, then relatively minor changes are required; the bargaining space must be modified to include this new possibility. A second, more complicated case is processor speed. The introduction of, say, 200 MHz Pentium chips presents an alternative on the processor dimension that the user will not have placed a value on if the option did not exist when values were elicited.

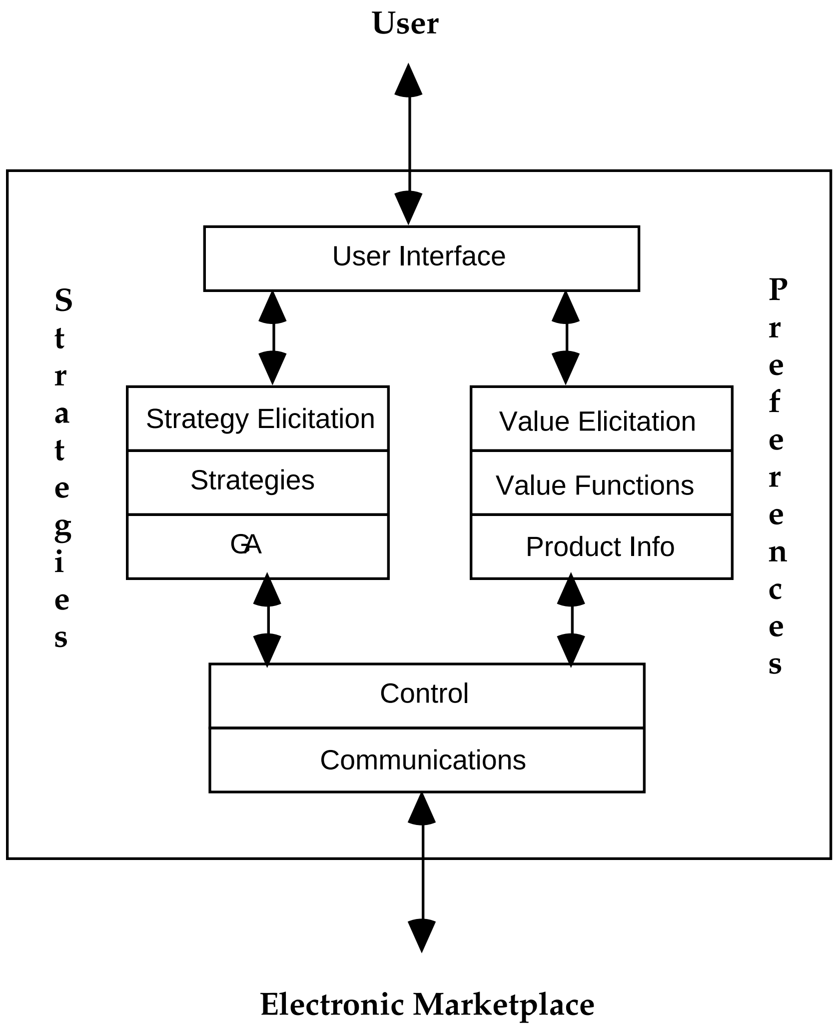


Figure 6. Architecture for automated negotiation system.

6.4 Strategies

The bargaining strategies and their modification are the centerpiece of a system of negotiating agents. In this research simple threshold rules have been used. A practical system needs the flexibility to support different types of strategies. Which additional types of strategies should be used or allowed by the AAAs is an important, open question.

In the system created for the experiments reported above, the initial populations of rules were generated randomly. In a practical system, depending of the environment the agents are to exist in, an alternative to random initialization might be desired. Humans rarely approach a novel situation de novo, and there is no reason for AAAs to do so either. Strategies could come from other agents or from humans. In the latter case, we might elicit strategies directly, or a system could watch the user and infer a strategy based on observed negotiations. The following discusses the possibilities in more detail.

1) *Random initialization*. If there is an effective learning algorithm, and an appropriate venue for learning, then random starting strategies might be the most cost effective.
2) *Obtain strategies from other AAAs*. Strategies of AAAs in similar situations should make a good starting point. This approach requires a way for the system, or the user, to understand which agent strategies would be appropriate for a novel situation.
3) *Direct strategy elicitation*. The simple threshold rules used in this work have a natural appeal because they are similar to the way human agents are sometimes told to negotiate for principals. For example, a human agent might be told, "Don't settle for less than X," or "Try to get Y, but it's OK if you can only get Z," and so forth. These types of strategies should be fairly straightforward to elicit from a user. To elicit other types of strategies could require a new representation scheme; this deserves further investigation.
4) *Bootstrapping*. This might also be called passively seeding the system, or apprenticeship mode. A useful, but possibly difficult undertaking, would be to design an algorithm that observes actual negotiations and infers which strategies are being used by one side, or even both sides. If both sides are estimated, simulations could be run to predict the results of alternative strategies. If an effective learning algorithm is available, then even a quick and dirty bootstrap model could be useful, as a starting point.

Depending on how the initial strategies are arrived at, additional modifications, prior to fielding agents in the marketplace, might be desired. We consider three possibilities.

1) *No learning.* If the initial strategies are believed to be of a high enough caliber—perhaps they were elicited from an experienced negotiator—then modifications to the initial rule set might not be required.
2) *Off-line learning with simulated opponent.* If a credible model of the opponent can be built, then a simulation approach can be taken to learn strategies that hopefully will be effective. An extension to a static opponent model would be to posit an adaptive procedure for the opponent. (See Sebenius [25] for an overview of this line of research.)
3) *Off-line learning in a practice forum.* Another possibility is to conduct practice games—with other market participants and not models of them—and have agents learn in this forgiving environment. The problem with this approach is that if the negotiations are non-binding then there is additional incentive to misrepresent preferences, because principals might not want to reveal information, or they might wish to establish a particular reputation. Anonymous

practice sessions would remove any reputation building incentive, but such a scenario is still not equivalent to real, live, binding negotiations[4].

Finally, when the user has confidence in the strategies available for use by the system, she will "field" the agents to the marketplace. Whether and how these strategies continue to be modified, based on the negotiation experiences, needs to be decided by the user. There are three possibilities.

1) *Fixed strategies*. One approach is to field static agents. Recall that the simple GA approach to learning occasionally created new strategies that agreed to low-value points in the bargaining space. While the effectiveness of the GA could undoubtedly be improved, users still might wish the agents to remain static, especially if the performance of the agents is stable. This is the if-it-ain't-broken-don't-fix-it philosophy. Static agents might also be justified in marketplaces which are relatively static, or in cases in which the downside of a poor strategy is large. Many financial transactions have a significant downside and current automated financial trading systems are carefully controlled; significant strategy changes in most of these systems are programmed from the outside the system, rather than the system selecting radically new strategies on its own.
2) *GA suggests*. A GA, or other learning algorithm, might *suggest* new strategies, but be prevented from acting on them without human approval.
3) *GA modifies*. In the most automated system, a GA—or any other learning algorithm—discovers and tries new strategies. The inconsistent performance of the GA approach might restrict this to less important transactions, or to games which have less of a downside. As an example, compare the divorce game to the no conflict game. The average difference between agreements on the pareto frontier and agreements from the set of points that dominate no others is much less in the divorce game than it is in the no-conflict game.

6.5 Control and Communication

All of the system activities must be coordinated, and this is the job of the control module. This module could also handle exceptions. An example is a supplier sends a customer a message about a product and the message includes an attribute that is not in the product database of the receiving system, or for which value function information has not been elicited.

The final module of the system is the communications module; its job is to interface to the electronic marketplace.

4 "Cheap talk" is the game theoretic term for low-cost, nonbinding, nonverifiable communication, and it has been an active research area. See, for example, Farrell [25] for an example of how cheap talk can achieve partial coordination among market participants.

6.6 Comment

This above list is focused on local systems and neglects services and features that are likely to be the domain of a centralized system, or will be distributed throughout the system. Security features, such as authentication and certification, secure payment mechanisms, and measures to protect the systems from unauthorized intrusions are all needed. These security features are needed in any electronic commerce networks and are not unique to the case of negotiating agents. Significant academic and commercial efforts are currently directed toward these problems and we leave this important area for others.

7 CONCLUSIONS

We showed how AAAs, using simple satisficing rules can, from a random start, learn to play negotiation games under the "direction" of a basic GA. While refinements to the GA, such as improved crossover and better diversity management, likely would improve effectiveness, the performance of these basic agents stands on its own merits. Systematic, statistical comparison with humans shows the ability for AAAs to perform similarly to humans, and even exceed their performance. That this level of success would be achieved was not a priori obvious, and the results hint at exciting possibilities for electronic commerce.

The success of the AAAs illustrates the power of the adaptive approach. One might not expect such simple agents to exhibit such complex behavior and to perform so well. For example, one might wonder, how do the agents know the unknown about the other agent? In fact, the agents do not have explicit models of the other agents, yet they have strategies that are adapted to their environment, which is created by the other agents.

This research reinforces the idea that computational science, in general, and evolutionary algorithms, in particular, provide a rich tool for the study of bargaining and negotiation. The set of programs developed for this research allows us to put negotiation dynamics under a microscope. GAs were chosen as a robust and general learning mechanism, and we are encouraged by the success of the simple GA in a difficult, dynamic, coevolutionary environment. In well defined situations, such as many optimization contexts, it is possible to find a better performing, specialized algorithm. Similarly, it might be possible that specialized negotiation learning algorithms will be developed; insights from this and related work should inform such efforts as the field matures.

REFERENCES

1. Camerer, C., *Behavioral Game Theory*, in *Insights in Decision Making: A Tribute to Hillel J. Einhorn*, R.M. Hogarth, Editor^Editors. 1990, Univ. of Chicago Press: Chicago, IL. p. 311-336.
2. Raiffa, H., *The Art and Science of Negotiation*. 1982, Cambridge, MA: Harvard University Press.
3. Bazerman, M.H. and M.A. Neale, *Negotiator Rationality and Negotiator Cognition: The Interactive Roles of Presecriptive and Descriptive Research*, in *Negotiation Analysis*, P.H. Young, Editor^Editors. 1992, Univ. of Michigan Press: Ann Arbor, MI. p. 109-129.

4. Nash, J., *The Bargaining Problem.* Econometrica, 1950. **18**: p. 155-162.
5. Nash, J.F., *Two Person Cooperative Games.* Econometrica, 1953. **21**: p. 128-140.
6. Linhart, P.B., R. Radner, and M.A. Satterthwaite, *Bargaining with Incomplete Information*. 1992, San Diego: Academic Press.
7. Rangaswamy, A. and R. Shell, *Using Computers to Realize Joint Gains in Negotiations: Toward and Electronic Bargaining Table*, . 1995.
8. Matwin, S., T. Szapiro, and K. Haigh, *Genetic Algorithms Approach to a Negotiation Support System.* IEEE Transactions on Systems, Man, and Cybernetics, 1991. **21**(1 (Jan/Feb)): p. 102-114.
9. Bui, T. *Building DSS for Negotiators: A Three-Step Design Process.* in *Hawaii International Conference on System Sciences*. 1992. Kauai, Hawaii: IEEE.
10. Jelassi, M.T. and A. Foroughi, *Negotiation Support Systems: An Overview of Design Issues and Existing Software.* Decision Support Systems, 1989. **5**(2): p. 167-181.
11. Woo, C.C. and M.K. Chang, *An Approach to Facilitate the Automation of Semistructured and Recurring Negotiations in Organizations.* Journal of Organizational Computing, 1990. **2**(1): p. 47-76.
12. Bond, A.H. and L. Gasser, *Readings in Distributed Artificial Intelligence*. 1988, San Mateo, CA: Morgan Kaufmann.
13. Rosenchein, J.S. and G. Zlotkin, *Designing Conventions for Automated Negotiations.* AI Magazine, 1994(Fall): p. 29-46.
14. Dworman, G.O., S.O. Kimbrough, and J.D. Laing. *On Automated Discovery of Models Using Genetic Programming in Game-Theoretic Contexts*. in *Twenty-Eighth Annuyal Hawaii International Conference on System Sciences*. 1995. Maui, HI: IEEE Computer Society Press.
15. Marimon, R., E. McGrattan, and T.J. Sargent, *Money as a Medium of Exchange in an Economy with Artificially Intelligent Agents.* Journal of Economics, Dynamics and Control, 1990. **14**: p. 329-373.
16. Ho, T.-H., *Finite Automata Play Repeated Prisoner's Dilemma with Information Processing Costs*, . 1991, Operations and Information Management, The Wharton School.
17. Miller, J.H., *The Coevolution of Automata in the Repeated Prisoner's Dilemma.*, . 1989, Santa Fe Institute.
18. Axelrod, R., *The Evolution of Strategies in the Iterated Prisoner's Dilemma*, in *Genetic Algorithms and Simulated Annealing*, L. Davis, Editor^Editors. 1987, Morgan Kaufman Publishers: Los Altos, CA. p. 32-41.
19. Oliver, J.R., *Finding Decision Rules with Genetic Algorithms.* AI Expert, 1994. **9**(3): p. 32-39.
20. Rust, J., J.H. Miller, and R. Palmer, *Behavior of Trading Automata in a Computerized Double Auction Market*, in *The Double Auction Market*, D. Friedman and J. Rust, Editor^Editors. 1992, Addison-Wesley: Reading, MA. p. 155-198.
21. Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*. 1989, Reading, MA: Addison-Wesley.
22. Young, P., ed.^eds. *Negotiation Analysis*. . 1991, University of Michigan Press: Ann Arbor, MI.

23. Bottom, W.P. and A. Studt, *Framing Effects and the Distributive Aspect of Intgrative Bargaining*. Organizational Behavior and Human Decision Processes, 1993. **56**: p. 459-474.
24. Foroughi, A., W.C. Perkins, and M.T. Jelassi, *An Empirical Study of an Interactive, Session-Oriented Computerized Negotiation Support System (NSS)*. Group Decision and Negotiation, 1995. **4**: p. 485-512.
25. Sebenius, J.K., *Negotiation Analysis: A Characterization and Review*. Management Science, 1992. **38**(1 (January)): p. 18-38.