Co-evolutionary auction mechanism design: a preliminary report

Steve Phelps[1], Peter McBurney[1], Simon Parsons[123], Elizabeth Sklar[4]

[1] Department of Computer Science, University of Liverpool
Chadwick Building, Liverpool L69 7ZF, UK.
{sphelps,p.j.mcburney}@csc.liv.ac.uk

[2] Center for Coordination Science, Sloan School of Management,
M.I.T., 3 Cambridge Center, NE20-336, Cambridge, MA 02142, USA
sparsons@mit.edu

[3] Department of Computer and Information Science, Brooklyn College
2900 Bedford Avenue, Brooklyn, New York, NY 11210, USA
parsons@sci.brooklyn.cuny.edu

[4] Department of Computer Science, Columbia University
1214 Amsterdam Avenue, Mailcode 0401, New York, NY 10027, USA
sklar@cs.columbia.edu

Abstract. Auctions can be thought of as a method for resource allocation. The economic theory behind such systems is *mechanism design*. Traditionally, economists have approached design problems by studying the analytic or experimental properties of different mechanisms. An alternative is to view a mechanism as the outcome of some *evolutionary process* involving *buyers*, *sellers* and an *auctioneer*, and so automatically generate not just strategies for trading, but also strategies for auctioneering. As a first step in this alternative direction, we have applied genetic programming to the development of an auction pricing rule for double auctions in a wholesale electricity marketplace.

1 Introduction

Much recent work in the field of Multi-Agent Systems (MAS) has focused on resource allocation problems, for example [8, 15]. These problems are especially difficult to solve efficiently in an open system if the values which agents place on resources, or the values of their human principals, are private and unobservable. In such a situation, the difficulty facing somebody wishing to allocate the resources to those who value them most highly is that participating agents cannot necessarily be relied upon to report their private values truthfully; there is nothing to prevent "greedy" agents from exaggerating their resource requirements. Auction mechanisms attempt to overcome this difficulty by having agents support their value-claims with hard cash. This has two advantages. First it punishes greedy agents by making them pay for the resources that they have oversubscribed to (alternatively one can think of this as preventing agents from oversubscribing by forcing them to pay a higher price than they would have to pay for the resources they actually need). Second, it allocates resources to the agents who pay the most, which should be the agents who value the resources most highly.

Designing mechanisms to achieve specific economic requirements, such as achieving market efficiency or maximising social welfare, against self-interested intelligent traders, is no trivial matter as can be seen from accounts of the auction design process for the recent radio spectrum auctions in the UK [16] and the US [7, 18]. The economic theory of mechanism design approaches the task of designing efficient resource allocation mechanisms by studying the formal, analytical properties of alternative mechanisms [14, 28]. However, for some kinds of mechanisms, including continuous double auctions [10], the mechanisms are too complex to admit analytical solutions. Because of these complexities, economists are increasingly turning to computational methods in an attempt to take what has been called an "engineering approach" [25, 27] to problems in microeconomics. We follow the computational approach in this paper. However, we also take what we believe is a step beyond existing work by using computational methods to design the auction rules themselves, rather than just explore their behaviour.

2 Co-evolution

One approach to computational microeconomic design is to use techniques from machine learning to explore the space of possible ways in which agents might act in particular markets. For example, reinforcement learning has been used to explore bidding patterns in auctions [22, 25] and establish the ways in which price-setting behaviour can affect consumer markets [30]. Another approach is to use techniques from evolutionary computing, that is from genetic algorithms [13] and genetic programming [17].

Inspired by the biological metaphor of evolution, genetic algorithms code aspects of a solution to a problem in an artificial "chromosome" (typically a binary string) and then breed a population of chromosomes using techniques like crossover (combining bits of the strings from different individuals) and mutation (flipping individual bits). Genetic programming extends this approach by evolving not a bit-string-encoded solution to a problem, but an actual program to solve the problem itself. Programs are encoded as s-expressions and modelled as trees (nodes are function names and branches arguments of those functions); and these trees are subject to crossover (swapping subtrees from different programs) and mutation (replacing subtrees with random subtrees). Whichever approach is used, the best individuals, evaluated using a *fitness* function, are kept and "bred"; and bad individuals are rejected. However, deciding which individuals are the best is a hard problem.

Evolutionary approaches perform a search through a space of solutions with the theoretical advantage that random jumps around the search space—created by crossover and mutation—can prevent the system from getting stuck in local optima, unlike other machine learning techniques. Unfortunately, in practice this is not always the case at least partly because what constitutes the best fitness measure can change over time. To overcome this problem, some researchers have turned to *co-evolution* [1, 12, 20], and the aim of our work is to apply co-evolution to economic mechanism design.

In co-evolution, simultaneously evolving populations of agents interact, providing each other with a fitness measure that changes as the agents evolve. In successful applications, an "arms race" spiral develops wherein each population spurs the other(s) to advance and the result is continuous learning for all populations. However, this has been notoriously difficult to achieve. Often populations settle into a *mediocre stable state*, reaching a local optimum and being unable to move beyond it. Consequently, there is a growing body of work examining the dynamics of co-evolutionary learning environments in an attempt to identify phenomena that contribute to success [2, 6, 9, 23]. The following aspects are of particular importance (some of which are relevant for both evolutionary and co-evolutionary techniques):

1. choice of representation for individuals within each population;
2. definition of a fitness function for determining which individuals in a population will reproduce;
3. choice of operators and proportion of population(s) to be used for reproduction;
4. selection of learning experiences for individuals (i.e., who interacts with whom, how many times and how frequently);
5. size of population and number of populations;
6. avoidance of collusion wherein members of different populations can work together to make non-optimal moves that may produce better short-term results for each but may cause the populations as a whole to get stuck in local optima; and
7. a clearly defined vision of the fitness landscape and how to measure progress so that one can even recognize if a local (or indeed global) optimum has been reached.

Note that the notion of collusion in co-evolution is not necessarily the same as the notion of collusion in auction theory (although the concepts are related). Collusion in co-evolution is where members of the co-evolving populations help each other to score high fitness, the by-product being that the populations as a whole settle into a local optimum. Collusion in auction theory is where several bidders work together to ensure that they trade at a higher profit than they would otherwise have obtained were they not working together. For example, a group of buyers might band together to purchase goods for less than they would otherwise pay [19].

We see efficient mechanisms evolving through repeated interactions between participants who may also be evolving individually—thus we believe that the co-evolutionary methodology is highly appropriate for our problem. Thus it is our long term aim to understand the above aspects for the evolution of trading strategies and auction rules.

In our work, we are using genetic programming (GP) [17] to represent individuals with different roles in an auction: the auctioneer, and the two types of traders (buyers and sellers). Through the interactions of the traders, individual and group trading strategies evolve, as well as auction mechanisms themselves. We view the mechanisms as "hosts" and the trading strategies as "parasites". If greedy, non-truthful strategies were to emerge, then we would hope that the auctioneer population would adapt defenses, and that strategy-proof, incentive-compatible mechanisms would evolve. Investigation of such an approach is the long-term aim of our research, and to our knowledge we are the first to apply genetic programming and co-evolution to mechanism design (though [4] describes similar work—this is discussed in more detail in Section 5).

Here, we report our initial work towards this aim. In Section 3, we describe the scenario we are studying. Section 4 then describes our use of genetic programming to co-evolve trading strategies for buyers and sellers in these auctions, and some of our preliminary results in using genetic programming to evolve auction pricing rules. Section 5 discusses how these results fit into our overall plan of work, and describes some future directions. Finally, Section 6 concludes with a brief summary.

3 The Experimental Scenario

To provide a multi-agent test-bed for such an approach we have adopted the wholesale electricity market auction simulation model of Nicolaisen *et al.* [22]. In this section, we provide a detailed description of the market model used in [22] as a preliminary to understanding both the application of our approach and the results we have obtained.

In the scenario explored in [22], a number of traders buy and sell electricity in a discriminatory-price[1] continuous double auction. Every trader has a *private value* for the electricity that they trade; for buyers this is the price that they can obtain in a secondary retail market and for sellers this reflects the costs associated with generating the electricity. Here, as in most work in auction theory, this value is considered private to individuals—because the traders are always trying to make a profit themselves (what we are calling "local profit"), sellers are not willing to reveal how little they might accept for units of electricity and buyers are not willing to reveal how much they might pay for units of electricity.

Trade in electricity is affected by capacity constraints; every trader has a finite maximum capacity of electricity that they can generate or purchase for resale. The *generating capacity* that each individual buyer, B_i can resell is written as GC_{B_i}, and the *generating capacity* that each seller S_j can generate is written as GC_{S_j}. The market proceeds in rounds of trading. In each round, all the traders

[1] In any auction that trades multiple items and so has several traders buying and/or selling simultaneously, there is a choice to be made about the price at which trades occur. In *uniform* price auctions, all trades happen at the same price. In *discriminatory* price auctions, trades take place at prices which are determined by the values that the traders have indicated they prefer.

are given the opportunity to bid in a random order. Each trader selects a price and submits a bid or an ask—in double auction terminiology, buyers make "bids" and sellers make "asks", bids are offers to buy at a given price and asks are offers to sell at a given price—at that price and with a quantity equal to their generating capacity. Traders may "pass" if they do not wish to submit a bid or ask. The auctioneer then matches bids and asks, and sets the *trade price* at which units of capacity are traded. The market proceeds until a maximum number of auction rounds is reached.

Traders seek to maximise their own *local profit*, and clearly this depends on the price that they buy/sell their capacity for. In particular, the local profit of an individual buyer, B_i, is calculated as:

$$Profit_{B_i} = \sum_{k=1}^{NT_{B_i}} private\_value_k - trade\_price_k$$

where NT_{B_i} is the number of goods traded by buyer B_i. Similarly, the local profit of a seller S_j is:

$$Profit_{S_j} = \sum_{p=1}^{NT_{S_j}} trade\_price_p - private\_value_p$$

where NT_{S_j} is the number of goods traded by seller S_j. Thus for both buyers and sellers, their profit on a given trade is just the difference between the amount for which they buy or sell and their private value.

In addition to the local profit, we can also define what we call the *global profit*, which is the total profit across all the buyers and sellers:

$$GlobalProfit = \sum_{i=1}^{NB} Profit_{B_i} + \sum_{j=1}^{NS} Profit_{S_j}$$

where NB is the number of buyers and NS is the number of sellers in the market. It is the job of the market to maximise the global profit. The maximum value of global profit is where every buyer and seller bids or asks their private value. This is not typically the same as sellers selling at their private value and buyers buying at their private value—otherwise the global profit would be zero—since the auctioneer typically sets the trade price somewhere above the ask price and below the bid price. If the global profit falls below the maximum, it means that more profit could be generated if the goods were distributed differently (to buyers that valued them more highly, for example).

It is possible to measure how well the market as a whole is operating by calculating how close the profit generated is to the maximum global profit that is possible. In real life, of course, this cannot be done, since there is no way of knowing the private values of the participants. However, in a simulation we know what the private values are, and so can compute local profits, and then global profit. It is also possible to calculate the *theoretical profit* TP, which is

	Relative Capacity								
	1/2			1.00			2.00		
			stdev			stdev			stdev
	Buyer MP	-0.13	(0.09)	Buyer MP	-0.15	(0.09)	Buyer MP	0.10	(0.30)
	Seller MP	0.55	(0.38)	Seller MP	0.38	(0.33)	Seller MP	-0.10	(0.25)
2									
	Efficiency	99.81	(0.02)	Efficiency	96.30	(0.05)	Efficiency	99.88	(0.06)
Relative	Buyer MP	-0.22	(0.12)	Buyer MP	-0.13	(0.10)	Buyer MP	0.13	(0.33)
Concentration	Seller MP	0.80	(0.53)	Seller MP	0.28	(0.35)	Seller MP	-0.10	(0.26)
1									
	Efficiency	92.13	(0.09)	Efficiency	94.59	(0.07)	Efficiency	100.00	(0.00)
	Buyer MP	-0.21	(0.12)	Buyer MP	-0.14	(0.08)	Buyer MP	0.09	(0.24)
	Seller MP	0.67	(0.46)	Seller MP	0.30	(0.31)	Seller MP	-0.07	(0.19)
1/2									
	Efficiency	91.84	(0.09)	Efficiency	94.24	(0.07)	Efficiency	100.00	(0.00)

Table 1. Market power and efficiency outcomes for the best-fit MRE algorithm with 1000 auction rounds and parameter values s(1) = 9.00, r= 0.10, and e = 0.20. Refer to [22] for a detailed description of the MRE parameters: r the recency parameter; e the experimentation parameter and $s(1)$ the scaling parameter.

just the value of the global profit when all traders bid or ask at their private values. The usual way to measure how well a market is operating, is to calculate *market efficiency*, ME:

$$ME = \frac{GlobalProfit}{TP}$$

expressed as a percentage. There are two parts to the theoretical profit, TP_B, the theoretical profit made by the buyers, and TP_S, that made by the sellers.

Market efficiency, of course, looks at both buyer and seller profits together. Thus it can obscure asymmetries in the market where, for example, buyers are making lots of profit at the expense of sellers—when the market is skewed it is still possible to get high efficiency, but there is obviously something wrong with the market in terms of fairness. Two other measures, measures of market power MP, identify how close the buyers and sellers come to sharing the profits equally. Buyer market power is the difference between the actual profits earned by the buyers and the profits they would earn were all buyers and sellers trading at their private values, expressed as a fraction of the theoretical profits available to the buyers:

$$BuyerMP = \frac{TP_B - \sum_{i=1}^{NB} Profit_{B_i}}{TP_B}$$

Seller market power is definined in similar way:

$$BuyerMP = \frac{TP_S - \sum_{j=1}^{NS} Profit_{S_j}}{TP_S}$$

Clearly market efficiency and market power will depend on a number of factors, and the ones investigated in [22] are *relative concentration* (RCON) and *relative capacity* (RCAP). RCON is the ratio of the number of buyers to the number of sellers:

$$\text{RCON} = \frac{NS}{NB}$$

and RCAP is the relative generating capacity of each group:

$$\text{RCAP} = \frac{\sum_{i=1}^{NB} GC_{B_i}}{\sum_{j=1}^{NS} GC_{S_j}}$$

The main results from [22] are summarised in Table 1. Each cell of the table corresponds to particular values for RCON and RCAP. These are obtained for agents that use a myopic reinforcement learning algorithm (which is a modification of the Roth-Erev algorithm [26]) to choose prices at which to trade. Using this approach each agent chooses to bid or ask at a value other than its private value, selecting the deviation from the private value by picking one of K possible "mark-ups", and the agent then receives a reward directly proportional to the profits that result from this bid or ask. The agent makes the choice by generating random numbers according to a probability distribution built up linearly from the cumulative rewards for each possible action. The modified Roth-Erev algorithm (MRE) has three main parameters: r the recency parameter; e the experimentation parameter and $s(1)$ the scaling parameter.

Because traders use stochastic strategies, the sensitivity of these outcomes to particular values of the pseudo-random number generator seed was tested by running the experiment 100 times with different seeds on each iteration. For each variable we reproduce the average result, followed (in Table 1) by the standard deviation in parentheses. These results are significant because they indicate that there are market biases inherent in the discriminatory-price auction rules. For example, one would expect that Seller MP should increase as RCAP increases, but this is not what is found by experimentation. Nicolaisen *et al.* [22] conclude that the inherent market-structure is responsible for failure of this hypothesis.

This electricity scenario was selected for our research because it focuses on market power and this seems an appropriate focus for investigating the evolution of auction rules. As agents evolve successful extra-marginal strategies, their market power indices will increase. For example, if sellers were able to employ collusive price-fixing strategies, we should expect to see their market power indices grow. Different auction rules may have differing abilities to counter this kind of tactic; hence, market power outcomes are an important quantative metric to consider in assessing auction designs.

4 Co-evolution using Genetic Programming

Having outlined the scenario we are using, we now describe our work with this scenario. There are five parts to this description. In the first part we detail the functioning of the auction mechanism. In the second part we explain how we use genetic programming to evolve traders and auctioneers. In the third and fourth parts, which include the main results of the paper, we describe two sets of experiments using this set-up. Finally the fifth part gives some pointers to our more recent work.

4.1 The market set-up

We can think of the electricity market in its most general form as an iterated game between three types of agents:

- Sellers
- Buyers
- Auctioneers

Each iteration of the game involves three steps. First the traders, that is the set of all buyers and sellers, make a move. These moves collectively indicate how many units of generating capacity they want to trade and at what price they wish to trade. The auctioneer then moves, matching traders based on the moves that the traders made previously. Finally the traders either accept or reject the matches suggested by the auctioneer.

In the first step of the game, sellers have the choice of making one of two moves—they either issue an ask, `ask(units, price)`, or they pass, `pass()`. Similarly, buyers have the choice between bidding, `bid(units, price)`, or passing, `pass()`. In the second step, Auctioneers have a choice between matching buyers and sellers (or more accurately the bids and asks issued by buyers and sellers), `match(buyer, seller, price, quantity)`, or they can `pass()`. Note that during an auctioneer's turn, she can make either multiple `match` moves or a single `pass` move. The key to the operation of the market is the auctioneer's job of matching buyers and sellers, based on their current `bid` and `ask` moves. In our work, the matching is carried out using the 4-heap algorithm [32] and works as follows.

We can represent the current bids and asks in two tables. The `bid` table is sorted in descending order, from highest price to lowest. The `ask` table is sorted in ascending order, from lowest price to highest. For each good for which there exists either a bid or an ask, there is an entry in each table, indicating the trade price and the buyer or seller. The matching process simply matches corresponding rows in the two tables, until the price in the bid row falls below the price in the ask row. As an example, consider the following moves for a market containing three buyers and three sellers:

B_0	`ask(2, 10)`
B_1	`ask(1, 20)`
B_2	`ask(2, 5)`
S_0	`bid(2, 15)`
S_1	`bid(2, 8)`
S_2	`bid(1, 9)`

The matching algorithm then matches the bids that offer above the ask price, matching the highest bid with the lowest ask to generate the largest surplus. Note that bids and asks for multiple units are broken down into multiple bids and asks for single items:

	bids		asks	
B_1	20	$\rightarrow$	8	S_1
B_0	10	$\rightarrow$	8	S_1
B_0	10	$\rightarrow$	9	S_2
B_2	5		15	S_0
B_2	5		15	S_0

Here the first three pairs of entries match, the last two do not. The auctioneer's response would then be to make all the matches (the auctioneer will only pass when there are no matches):

$$\texttt{match}(B_1, S_1, \mathit{trade\_price}_0, 1)$$
$$\texttt{match}(B_0, S_1, \mathit{trade\_price}_1, 1)$$
$$\texttt{match}(B_0, S_2, \mathit{trade\_price}_2, 1)$$

Any unmatched bids or asks remain in the market. Part of the auctioneer's job is to determine the trade price, and it is evolving the rule for making this decision that is the overall goal of the experiments we report here.

For the experiments described below, we used our own Java implementation of the 4-heap algorithm. This software is available under an open-source license at `http://jasa.sourceforge.net/`. All price information was encoded using double-precision floating point variables and all quantity information was encoded using integers.

4.2 The genetic programming setup

We are trying to evolve agents as players in the market game described above. To do this, we need to evolve strategies for each type of player, given the types of moves that each player can make. For a buyer, this is whether to pass in a given round or to make a bid; and if a bid is made, how many goods to bid for at what price. For a seller, this is whether to pass in a given round or to put in an ask; and if an ask is made, how many goods to ask for at what price. For an auctioneer, this is how to match bids and asks and to determine the trading prices. For the moment, we are using the 4-heap algorithm, so here we are only evolving a strategy for determining the trading prices based on the matched ask and bid prices.

To allow for a variety of trading strategies we consider each game to take place between seven players—1 auctioneer (A), 3 buyers (B_0, B_1, B_2) and 3 sellers (S_0, S_1, S_2). Each player is represented by a population of agents—here we use 100 as illustrated in Table 2. To implement the agents, we made use of a Java-based evolutionary computation system called ECJ.[2] ECJ implements a strongly-typed GP [21] version of Koza's [17] original system. For all of the GP experiments in this paper, the standard Koza parameters were used in combination with the standard Koza GP operators. The functions given in Tables 3, 4 and 5 were used as the GP function-set, and the initial populations were generated randomly using these functions. As is usually the case with GP, individuals are tree structures composed of these functions.

[2] `http://www.cs.umd.edu/projects/plus/ec/ecj/`

A	B_0	B_1	B_2	S_0	S_1	S_2
A_0	$B_{0,0}$	$B_{0,1}$	$B_{0,2}$	$S_{0,0}$	$S_{0,1}$	$S_{0,2}$
A_1	$B_{1,0}$	$B_{1,1}$	$B_{1,2}$	$S_{1,0}$	$S_{1,1}$	$S_{1,2}$
A_2	$B_{2,0}$	$B_{2,1}$	$B_{2,2}$	$S_{2,0}$	$S_{2,1}$	$S_{2,2}$
$\vdots$	$\vdots$					$\vdots$
A_{99}	$B_{99,0}$	$B_{99,1}$	$B_{99,2}$	$S_{99,0}$	$S_{99,1}$	$S_{99,2}$

Table 2. Each player is a population of 100 agents.

We begin with random initial populations, we establish the fitness of each population, then pick individuals to breed the next generation and repeat. The fitness of traders is given by their local profit over a number of iterations of the game, and the fitness of the auctioneer is given by the global profit. Since we are dealing with 100-strong populations for each player in the game we need to play a number of rounds of the game for each generation. The algorithm for handling a single generation is shown in Figure 1 (the rows and columns mentioned there refer to Table 2), where K determines the number of games played to establish the fitness of each set of strategies. Once fitness is established, individuals are selected for breeding using tournament selection [11], with a tournament size of 7. The crossover operator is applied with a probability of 0.9, and the reproduction operator is applied with a probability of 0.1, as per standard Koza GP [17].

4.3 Co-evolution of Trading Strategies

In the first experiments we carried out, we evolved separate populations of strategies for each trader in the electricity market scenario, but did not try to evolve the auctioneer. Instead, the auctioneer used a standard discriminatory price rule for establishing the trading prices, the aim being to calibrate this work against the results provided by [22]. Thus the scenario is basically that described in Section 3, the only difference being that instead of using the modified Roth-Erev algorithm to select prices, buyers and sellers select prices by evaluating functions that were evolved using genetic programming.

```
randomly shuffle the 100 population members within each column
for i ← 0 to 99 {
  for j ← 1 to K {
    play one market game
  }
  calculate the fitness for row i
}
perform reproduction and selection
```

Fig. 1. The algorithm for establishing the fitness of players in the market game.

Function	*Arguments*	*Return-type*	*Description*
$+$	($+$ `number number`)	`number`	Addition
$-$	($-$ `number number`)	`number`	Subtraction
/	(/ `number number`)	`number`	Division
$*$	($*$ `number number`)	`number`	Multiplication
1	none	`number`	1
DoubleERC	none	`number`	A double precision floating point ephemeral random constant in the range (0..1).
QuoteBidPrice	none	`number`	The current bid quote
QuoteAskPrice	none	`number`	The current ask quote

Table 3. GP functions common to all function-sets

Function	*Arguments*	*Return-type*	*Description*
$<$	($<$ `number number`)	`boolean`	Less-than function
$=$	($=$ `number number`)	`boolean`	Equals function
$>$	($>$ `number number`)	`boolean`	Greater-than function
True	none	`boolean`	True
PrivateValue	none	`number`	The agent's private valuation for electricity
Nand	(Nand `boolean boolean`)	`boolean`	Not-and boolean operator
IfElse	(IfElse `boolean number number`)	`number`	Return 2nd argument if condition is true, otherwise return 3rd argument.

Table 4. Additional GP functions used in evolving trading strategies

The use of a separate co-evolving population for each trader allowed us to explore the potential emergence of collusive (in the auction sense) tactics between self-interested traders; each population attempts to maximise its own profits, but in certain situations populations may be able to increase their profits by co-operating with rival populations. This could not be modeled, by, for example, representing all of the buyers as a single population, since this optimization problem would not account for self-interested behaviour of individual traders.

Since the trading strategies are evolving, there are times when the function determining what price to set gives impossible results. Our method for handling the necessary error correction is as follows—wherever evaluation of this function resulted in a negative price, or in a division by zero exception, the price was set to 0 and this was used as the requisite bid or ask.

Initially, we were interested in whether high-efficiency outcomes are sustained in this experiment. As with the original experiments, high levels of market efficiency indicate that overall, traders are successfully "discovering" profits that

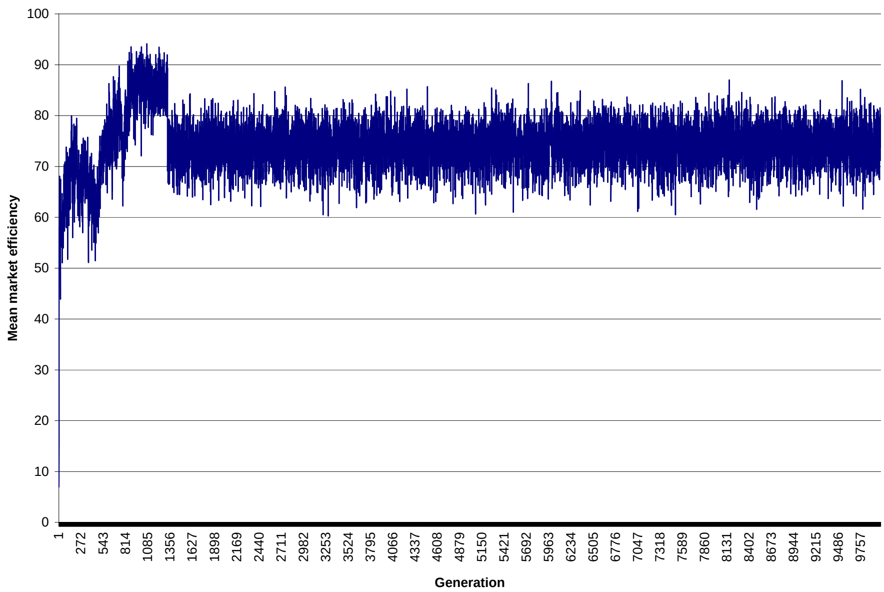

Fig. 2. Evolution of mean efficiency for RCON=1 and RCAP=1 over 10,000 generations using a fixed discriminatory-pricing auctioneer, and 6 sub-populations of co-evolving strategies each of size 100.

are available in the market. We would not necessarily expect to see stability, or gradual improvement, of each strategy's individual profits in this co-evolutionary scenario. However, if overall market efficiency were to decline temporarily, we would expect the co-evolving strategy set as a whole to adapt and reacquire the "lost" profits. Thus if strategy sub-populations were to successfully adapt to new market conditions, we would expect to see mean market efficiency remain stable at around 100% since mean market efficiency measures the performance of the different varieties of buyer and seller as a whole.

Figure 2 shows the evolution of the mean market efficiency for each generation of the experiment in the case RCAP=1 and RCON=1 over 10,000 generations. Note that by generation 2000, the market efficiency has become relatively stable, though it still fluctuates over a narrow range. The mean of this fluctuating value is 74.3%. This seems to suggest that the genetic programming approach permits the evolution of relatively efficient markets.

The use of co-evolution to evolve trading strategies is not new in experimental economics; for example, see [24]. Our interest in co-evolving strategies was mainly to verify that the genetic programming approach worked for this scenario, which it appears to. The work described in this section was also a step towards the use of co-evolutionary techniques to evolve trading strategies and auction rules—in

Function	*Arguments*	*Return-type*	*Description*
AskPrice	none	`number`	The price of the ask (offer to sell) currently being matched in the auction
BidPrice	none	`number`	The price of the bid currently being matched in the auction

Table 5. Additional GP functions used in evolving auctioneer pricing rules

other words to evolve mechanisms along with the best way to trade within them. This is the main focus of our research, and our preliminary work towards doing this will be the subject of the next section.

4.4 Co-evolution of Auction Pricing Rules

An additional population of *auctioneers* was introduced into our experiment giving the seven populations of players described in Section 4.2. These agents were derived from the auctioneer classes that we implemented for our previous experiments, but instead of using the standard code to set the clearing price for a given transaction, they used a function that was evolved using GP. The set of functions used for the auction pricing rule are those functions in Tables 3 and 5. The space of possible pricing rules thus encompasses, but is not restricted to, both uniform-price and discriminatory-price versions of the k-double auction[3]. Pricing rules which make use of the values *AskPrice* or *BidPrice* correspond to discriminatory-price auctions since *AskPrice* and *BidPrice* are the current prices that the auctioneer is considering as possible matches—the trade price is thus a function of what the traders have been offering. Pricing rules not making use of these functions correspond to uniform-price auctions since the trade rules are independent of what the traders in a particular match have been offering. Whereas the trading-strategy populations' fitness is, as in the last experiment, taken to be proportional to their individual local profits, the fitness for the auctioneer population is taken to be proportional to the global profits earned in the market.

Intuitively, the auctioneer population can be thought to be "learning" auction-pricing rules that maintain market efficiency in the face of co-evolving buying and selling strategies. Our hypothesis is that in this version of the experiment, in which there are a small number of traders with fixed private values, the most robust auction pricing rule is the one that sets the price for electricity at the

[3] As described in Section 4.1, it is the job of the auctioneer to choose the price at which trades take place. Given a matched pair of bid b and ask a, such that $a < b$, the auctioneer has to pick a trade price in the interval $[a, b]$. A k-double auction, $k \in [0, 1]$, is a double auction in which the auctioneer sets the trade price at $kb + (1 - k)a$ [29].

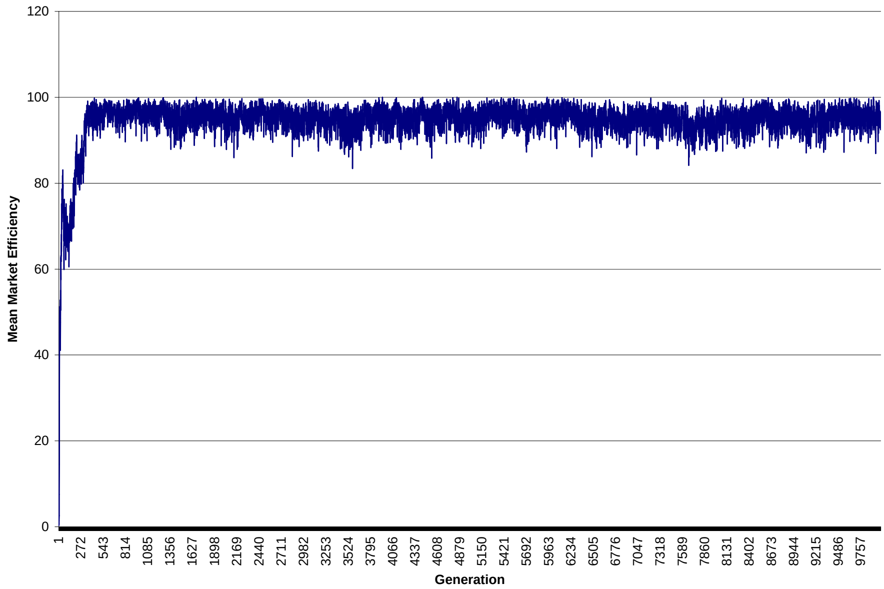

Fig. 3. Evolution of mean efficiency for RCON=1 and RCAP=1 over 10,000 generations using an auctioneer with a GP-evolved pricing rule, and 6 additional populations of co-evolving strategies.

equilibrium price, regardless of what traders actually bid. We believe that the auctioneer population should discover this rule—it should discover the equilibrium price for the market. It should do this because private values are fixed, and the auctioneer population has indirect access to meta-information about the market—market efficiency—that is based on the (in-practice unobservable) private values. Of course, this pricing rule would not work in practice, because in practice private values are not from a fixed, predefined set. However, by considering the hypothesis that the most robust pricing rule is the one that sets prices at the equilibrium level, we will be able to assess the validity of our underlying assumption.

Figure 3 shows the evolution of the mean market efficiency for each generation of this version of the experiment in the case RCAP=1 and RCON=1 over 10,000 generations. The graph seems to indicate that the adaptive auctioneers are able to respond to the changing strategies used by the traders. Again the market reaches stability relatively quickly, though once again the efficiency fluctuates across a narrow range—the average of this range is 94.5%.

Table 6 shows the stable pricing function evolved for the auctioneers' pricing rule under different market conditions. In all cases the pricing rule is a linear function of the Bid and Ask Prices and the function only uses either Bid Price

		RCAP		
		$\frac{1}{2}$	1	2
	2	$QuoteBidPrice - 0.39$	$QuoteBidPrice$	$QuoteBidPrice$
$RCON$	1	$\approx QuoteAskPrice$	$\approx QuoteAskPrice$	$\approx QuoteAskPrice$
	$\frac{1}{2}$	$35.47 - 35.47AskPrice$	$BidPrice$	$QuoteBidPrice$

Table 6. GP-evolved auction pricing rules at generation 1000 for different market conditions, i.e. different values of $RCON$ and $RCAP$

Seller 1	$QuoteAskPrice$
Seller 2	$QuoteBidPrice$
Seller 3	$QuoteBidPrice$
Buyer 1	$QuoteAskPrice$
Buyer 2	$QuoteAskPrice$
Buyer 3	$QuoteBidPrice$

Table 7. The set of trading strategies at generation 1000 for $RCON = 1$, $RCAP = 1$

or Ask Price[4]. When the number of buyers and sellers is equal, the pricing rule is only determined by the Ask Price, suggesting that the sellers control the market whatever the relative capacity. Table 7 shows the trading strategy-set for the auction after 1000 generations in the case $RCON = RCAP = 1$. These expressions are simplifications of the s-expressions generated by the GP. Most reduce exactly to the expressions given, but several seem to resist simplification—these were plotted against *QuoteAskPrice* and were found to be approximately equal to it. They are thus given as $\approx QuoteAskPrice$.

4.5 Further Experiments

One important issue that arises from the previous experiments concerns the private values of the traders. The strategies of the traders and, indirectly, the auctioneers, depend upon the private values of the traders. Thus if private values are fixed, it is possible for the populations to converge on constant values which approximate the private values rather than sensible functions. Thus when deciding on the market clearing price for a buyer with fixed private value of 20,

[4] Some of the pricing rules also use *QuoteBidPrice* or *QuoteAskPrice* which are the values made public by the auctioneer to give buyers and sellers an idea of what the current trading price is. It is the equivalent of the prices displayed on stock exchange tickerboards.

```
randomly shuffle the 100 population members within each column
for i ← 0 to 99 {
  for j ← 1 to K {
    randomly pick private values for row i
    for k ← 1 to N {
      play one round of the market game
    }
  }
  calculate the fitness for row i
}
perform reproduction and selection
```

Fig. 4. The revised algorithm for establishing the fitness of players in the market game (refers to Table 2)

and a seller with fixed private value 10, the traders may learn to bid at the constant values 20 and 10 respectively rather than at a function of *QuoteBidPrice*, *QuoteAskPrice*, or *PrivateValue*. Similarly, the auctioneer may learn to settle at 15 rather than:

$$\frac{BidPrice + AskPrice}{2}$$

The problem is that the rules used by traders and auctioneer are then overfitted to the specific private values, and will not produce efficient markets when private values change. This overfitting seems to be happening in the experiments reported above, and explains the relatively high efficiencies we obtained.

To reduce this effect we have recently been carrying out experiments in which we randomly pick private values for the traders for each one of the K games we play to determine the fitness. In addition, we play N times with each of the random private values, terming a single auction a "round of the game", and a set of N with fixed private values a "game". Thus the values of N and K represent the number of rounds to play in a game and the number of games to play in a generation, respectively. The revised algorithm is given in Figure 4.

Another issue that arises is that it is common for the markets to descend into bad solutions. In particular, traders can learn to bid below their profit margin, which, when the auctioneer tries to develop a strategy that gives good profits despite this, leads to a strategy that prevents the traders evolving to use more sensible strategies. In other words co-evolution is working to lock the system into a local optimum and prevent it from reaching a global optimum, just as described in Section 2. To try to combat this, we have reduced the search space for all the agents. Traders are restricted to use linear fuctions of their private values as bids or asks, and auctioneers are restricted to use linear functions of the ask and bid prices.

With these restrictions and the randomised private values we find that we can obtain market efficiencies around 86% without having the pricing rules degenerate into constant values. For example, we find buyer rules like:

```
if(not(QuoteBidPrice < (PrivateValue * 0.081675285)) {
  PrivateValue}
else {
  PrivateValue * 0.081675285 }
```

and the auctioneer uses rules like:

$$BidPrice - constant$$

and

$$AskPrice + constant$$

Results of these latest experiments are forthcoming.

5 Discussion

In terms of the seven aspects of any application of co-evolution that were raised in Section 2, we believe that the work described here has provided an adequate start to dealing with the first four—how to represent individuals, what counts as fitness, how to carry out reproduction, and how to perform selection. We can claim this because the choices we have explained above lead to the evolution of reasonable individuals as evidenced by the high level of market efficiency obtained in our experiments. However, there is still much work to be done. Even the results obtained so far have raised some interesting questions, such as how to interpret the different auction rules that can be evolved for each of the combinations of RCAP and RCON, and how to incorporate market-power metrics into the fitness function for auction rules. Clearly we also have to address the final three issues mentioned in Section 2 as well—correct population size, how to detect and avoid collusion, and how to measure progress.

This latter is a particularly important question since we need to be able to track the *adaptive progress*, as opposed to the instantaneous fitness, of the auctioneers verses the trading strategies. We are currently investigating the possibility of using CIAO (Current Individual vs. Ancestral Opponents) metrics as proposed in [6], in order to gain insights into the co-evolutionary dynamics of these experiments, and using pareto co-evolution [31] in order to ensure that auction designs are robust in the face of a diverse range of strategies.

Finally we should discuss the relation of our work with that of Cliff [4], which is the only other work that we are aware of in which the auction mechanism itself evolves. Cliff's work in this area builds on his Zero-Intelligence-Plus [3] traders, and first used genetic algorithms to determine the parameters that control the bidding behaviour of the agents [5]. This work is analagous to our use of genetic programming to decide how buyers and sellers bid. The next stage of Cliff's work, which was undertaken concurrently with, but independently of, ours was to add an extra parameter into the genetic algorithm representing the probability with which a buyer or seller is selected to make a Bid or Ask. The experiment thus explores a continuum between auctions in which only buyers act, like an English auction, and auctions in which only sellers act. This work, then, is only concerned with tuning one, admittedly important, parameter rather than constructing the auction rules from scratch. Furthermore, since Cliff's work involves just a single population of chromosomes—which capture the parameters which determine buyers, sellers and auctioneers—it is an evolutionary but not a co-evolutionary approach.

6 Summary

In this paper we have reported on the preliminary stages of work aiming to explore the evolution of economic auction mechanisms. In our initial work, we have adopted a multi-agent systems test-bed involving auctions in an electricity marketplace. We first described work in which buyer and seller strategies are co-evolved using genetic programming. The genetic programming approach was able to produce reasonably high efficiency outcomes in this case. Next we presented some of our preliminary work on evolving *auction designs* using genetic-programming which again was able to produce relatively high efficiency outcomes and was able to reach stability quicker than when the buyer and seller strategies evolved alone. Finally, we discussed a little of the work we have recently undertaken to overcome limitations inherent in these first experiments. We believe that this is the first attempt to evolve auction mechanisms, and, though far from complete, makes it possible to frame further research in this area.

Acknowledgements This work was partially supported by the EU IST Programme through the SLIE Project (IST-1999-10948). We also thank Larry Bull, Sam Hough, Michael Phelps, Alvin Roth, Leigh Tesfatsion and Peter Wurman for their assistance, and the anonymous referees for their helpful comments.

References

1. P. J. Angeline and J. B. Pollack. Competitive environments evolve better solutions for complex tasks. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 264–270, 1993.
2. A. D. Blair, E. Sklar, and P. Funes. Co-evolution, determinism and robustness. In *Proceedings of the Second Asia-Pacific Conference on Simulated Evolution and Learning*, pages 389–396. Springer-Verlag, 1999.
3. D. Cliff. Minimal-intelligence agents for bargaining behaviours in market environments. Technical Report HPL-97-91, HP Labs, 1997.
4. D. Cliff. Evolution of market mechanism through a continuous space of auction-types. Technical Report HPL-2001-326, HP Labs, 2001.
5. D. Cliff. Evolutionary optimization of parameter sets for adaptive software-agent traders in continuous double auction markets. Technical Report HPL-2001-99, HP Labs, 2001.
6. D. Cliff and Geoffrey F. Miller. Tracking the red queen: Measurements of adaptive progress in co-evolutionary simulations. In *European Conference on Artificial Life*, pages 200–218, 1995.

7. P. Cramton and J. Schwartz. Collusive bidding: Lessons from the FCC spectrum auctions. *Journal of Regulatory Economics*, 17:229–252, 2000.
8. S. S. Fatima and M. Wooldridge. Adaptive task and resource allocation in multi-agent systems. In *Proceedings of the 5th International Conference on Autonomous Agents*, pages 537–544. ACM Press, 2001.
9. S. G. Ficici and J. B. Pollack. Challenges in coevolutionary learning: Arms-race dynamics, open-endedness, and mediocre stable states. In *Proceedings of the 6th International Conference on Artificial Life"*, pages 238–247, 1998.
10. D. Friedman and J. Rust, editors. *The Double Auction Market: Institutions, Theories and Evidence*. Santa Fe Institute Studies in the Sciences of Complexity. Perseus Publishing, Cambridge, MA, USA, 1993.
11. D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, San Mateo, CA, 1991.
12. W. D. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. In *Proceedings of the Second International Conference on Artificial Life*, pages 313–324, 1992.
13. J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
14. M. O. Jackson. Mechanism theory. In *The Encyclopedia of Life Support Systems*. EOLSS Publishers, 2000.
15. N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, M. Wooldridge, and C. Sierra. Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, 10(2):199–215, 2001.
16. P. Klemperer. How (not) to run auctions: the European 3G telecom auctions. *European Economic Review (forthcoming)*, 2002.
17. J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
18. R. P. McAfee and J. McMillan. Analyzing the airwaves auction. *Journal of Economic Perspectives*, 10:159–176, 1996.
19. P. Milgrom. Auctions and bidding: A primer. *Journal of Economic Perspectives*, 3:3–22, 1989.
20. G. F. Miller and D. Cliff. Protean behavior in dynamic games: Arguments for the co-evolution of pursuit-evasion tactics. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 411–420, 1994.
21. David J. Montana. Strongly typed genetic programming. Technical Report #7866, Bolt Beranek and Newman, 10 Moulton Street, Cambridge, MA 02138, USA, July 1993.
22. J. Nicolaisen, V. Petrov, and L. Tesfatsion. Market power and efficiency in a computational electricity market with discriminatory double-auction pricing. *IEEE Transactions on Evolutionary Computation*, 5(5):504–523, October 2001.
23. J. B. Pollack and A. D. Blair. Co-evolution in the successful learning of backgammon strategy. *Machine Learning*, 32:225–240, 1998.
24. T. C. Price. Using co-evolutionary programming to simulate strategic behaviour in markets. *Journal of Evolutionary Economics*, 7:219–254, 1997.
25. A. E. Roth. The economist as engineer: Game theory, experimentation, and computation as tools for design economics. *Econometrica*, 70:1341–1378, 2002.
26. A. E. Roth and I. Erev. Learning in extensive form games: experimental data and simple dynamic models in the intermediate term. *Games and Economic Behavior*, 8:164–212, 1995.

27. J. Rust, J. Miller, and R. Palmer. Behavior of trading automata in a computerized double auction market. In D. Friedman and J. Rust, editors, *The Double Auction Market: Institutions, Theories and Evidence*, Santa Fe Institute Studies in the Sciences of Complexity, pages 155–198. Perseus Publishing, Cambridge, MA., 1993.
28. T. W. Sandholm. Distributed rational decision making. In G. Weiss, editor, *Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence*, pages 201–258. MIT Press, Cambridge, MA, USA, 1999.
29. M. A. Satterthwaite and S. R. Williams. The bayesian theory of the k-double auction. In D. Friedman and J. Rust, editors, *The Double Auction Market: Institutions, Theories and Evidence*, Santa Fe Institute Studies in the Sciences of Complexity, pages 99–123. Perseus Publishing, Cambridge, MA., 1993.
30. G. Tesauro and J. O. Kephart. Pricing in agent economies using multi-agent Q-learning. *Autonomous Agents and Multi-Agent Systems*, 5(3):289–304, 2002.
31. R. A. Watson and J. B. Pollack. Symbiotic combination as an alternative to sexual recombination in genetic algorithms. In *6th International Conference on Parallel Problem Solving from Nature*. Springer Verlag, 16-20 2000.
32. P. R. Wurman, W. E. Walsh, and M.P. Wellman. Flexible double auctions for electronic commerce: theory and implementation. *Decision Support Systems*, 24:17–27, 1998.