Applying Multi-Objective Evolutionary Computing to Auction Mechanism Design

S. Phelps, S. Parsons, E. Sklar, P. McBurney

Abstract

The mechanism design problem in economics is about designing rules of interaction for market games which aim to yield a globally desirable result in the face of self-interested agents who may attempt to take advantage of the mechanism in order to maximize their own individual outcomes. This problem can be extremely complex. Traditionally, economists have used game theory and other formal methods to construct mechanism rules. In this paper, we report on an alternative approach which we hope will eventually yield more robust solutions than the present analytical counterparts. Our methodology views mechanism design as a multi-objective optimisation problem and addresses the problem using genetic programming. This paper reports on preliminary work in this direction where we evolve an auction-pricing rule for a continuous double auction using a multi-objective fitness function.

1 Introduction

The auction mechanism design problem has attracted much interest in recent years, and economists have had considerable success in applying techniques from game theory to the design of auction-based markets for deregulated commodity markets (e.g., California's deregulated electricity market [2; 9]) and the sale of government assets (e.g., auctions of electromagnetic spectrum for mobile phones [11; 10]). Alvin Roth [16] has suggested that this is akin to an engineering process in which economists design the rules of a market mechanism in order to meet particular socio-economic requirements (e.g., maximising the efficiency of allocating commodities in a market).

The engineering of auction mechanisms is of particular importance to agent-based electronic commerce and multi-agent systems in general. E-commerce has enabled consumers to act as price-makers instead of just price-takers in large auction-based markets and has stimulated the use of personalised bidding agents to empower those consumers even more. In addition, auction mechanisms are seen as a promising means of solving many distributed resource-allocation problems in multi-agent systems and grid technology.

One approach to computational economics 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 [13; 16]. Another approach is to use techniques from evolutionary computing, e.g., from genetic programming (GP) [8]. Earlier work has explored the use of co-evolutionary GP to determine auction mechanism rules automatically [14; 15].

In that work, mechanism rules and bidding strategies were encoded and co-evolved in ways that sought to maximise overall market efficiency and the profits of individual agents. Here, we extend this work by focusing on the multi-objective optimisation issues inherent in the mechanism design problem.

The rest of the paper is as follows. In Section 2, we describe the standard view of n-player games and introduce the perspective we will take in this paper. Then in Section 3, we discuss in detail the scenario that we have been investigating. We present two sets of data: first, an attempt to map the fitness landscape using a standard class of auction pricing rules — the k-double auction (see section 4); and second, an experiment in which we try to find alternative to the k-double auction results by using genetic programming to evolve auction rules (section 5). We close with a discussion and summary.

2 Equilibria for n-player games

When evaluating a mechanism design, the designer must take into account the set of trading strategies that are likely to be played by agents trading in the mechanism under consideration. Deriving the set of the strategies likely to be played for a particular market game, that is "solving" the game, is a non-trivial problem in the general case. This is because there is often no clear *dominant* strategy which constitutes best play; rather the best strategy to play depends entirely on the strategies played by other agents. Nash defined a solution concept in which the strategy adopted by any given agent is a best-response to the best-response strategies adopted by all other agents, and proved that all n-player, non-zero-sum games admitted solutions so defined.

Nash's solution concept is widely adopted in theoretical economics. Thus when evaluating an economic mechanism, the designer computes the Nash equilibria of strategies for the given mechanism; and this forms the basis of predictions about how people will actually behave under the rules of this mechanism. The designer can then analyse market outcomes in equilibria and quantitively assess, for example, the likely affect on overall market-efficiency that a given change in the mechanism rules will yield. Thus the role of the designer is to ensure that the Nash equilibria correspond to situations in which high market efficiency is obtained.

Another approach focuses on mechanism design as a *multi-objective optimization* problem. We consider as a separate dimension each problem variable we are interested in maximising (for example, market efficiency, seller revenue and so on), and the difficulty lies in simultaneously maximising as many dimensions as possible.

We view the mechanism design problem as a multi-objective optimisation problem and the task is to choose mechanism rules which pareto-optimise different market variables when traders play Nash-equilibrium strategies. However, there are a number of problems beginning with computing the Nash equilibria:

1. Agents with limited computational power (i.e., "bounded rationality" constraints) may be unable to compute their Nash-equilibrium strategy;
2. Even with vast amounts of computational and analytic power, many games defy solution; e.g., in the case of the k-double-auction, analytical techniques have yet to yield a solution;
3. Empirical evidence shows that human agents often fail to coordinate on Nash-equilibria for very simple games whose solution is easily derivable under bounded-rationality assumptions [6]; and
4. Often a given game will yield a multitude of Nash solutions and there is little guidance for practitioners on choosing plausible subsets thereof as predicted outcomes.

These difficulties with the standard theory of games have led to the development of a field known as *cognitive game theory* [4], in which models of learning play a central role in explaining and predicting strategic behaviour. Erev and Roth [17] show how simulations of agents equipped with a simple reinforcement learning algorithm can explain and predict the experimental data observed when human agents play a diverse range of trading games. Such *multi-agent reinforcement learning* models form the basis of our solution concept for optimising mechanism designs. Rather than computing the theoretical equilibria for a given point in the mechanism search space, we run a number of multi-agent simulations using agents equipped with a learning algorithm that determines their bidding strategies.

Note that we are not attempting to find theoretically optimal strategies for our agents[1]. Rather, we are attempting to predict how boundedly-rational agents, who have no prior knowledge of an equilibrium solution nor the means to calculate one, might actually play against the mechanism we are (automatically) designing. For this reason, we chose to use the Roth-Erev algorithm[17], since it forms the basis of a *cognitive model* of how people actually behave in strategic environments. In particular it models two important principles of learning psychology:

- *Thorndike's law of effect* — choices that have led to good outcomes in the past are more likely to be repeated in the future; and
- *The power law of practice* — learning curves tend to be steep initially, and then flatter.

It is also important to note that the Roth-Erev algorithm models *self-interested* behavior. Agents' strategies are not explicitly chosen for globally desirable outcomes, such as achieving high overall market efficiency. Rather, agents attempt to maximise their own utility, under bounded-rationality constraints, possibly at the expense of global profit. It is only by adjusting the mechanism that we are able to improve global outcomes. This is in contrast to the related work of Cliff [1], in which both the mechanism and the strategies are simultaneously adjusted according to a fitness function that selects *only* for globally desirable outcomes.

3 Experimental setup

Our scenario stems from [13] (hereafter referred to as *NPT*). A more detailed description of our interpretation can be found in [14]. In this scenario, a number of traders buy and sell electricity in a discriminatory-price[2] continuous double auction [5]. Every trader assigns a 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 this value is considered *private*; because traders are always trying to make a profit themselves, 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.

The key to the operation of the market is the auctioneer's job of matching buyers and sellers, based on their current bids and asks, and setting the *trade price* at which units of capacity are traded. In our work, the matching process is carried out using the 4-heap algorithm [19]. The rule for determining the trade price is what we are trying to evolve.

In our experiments, the number of sellers, NS, is 30, the number of buyers, NB, is also 30, and there is one auctioneer. All traders have a capacity of 10 units. Traders are equipped with the modified version of the Roth-Erev learning algorithm reinforcement learning algorithm (MRE) described in [13]. The MRE algorithm is calibrated with four parameters: a scaling parameter $s(1)$, a recency parameter r, an experimentation parameter e and a parameter k representing the number of possible actions that can be taken by the learner. Each action represents a possible mark-up over the agent's limit price. In these experiments actions are scaled by a factor of 100 and then either added to, or subtracted from, the agent's private value in order to arrive at a bid or an ask

[1] In other words Nash equilibrium strategies

[2] In *uniform* price auctions, all trades in any given auction round happen at the same price. In *discriminatory* price auctions of the kind we have here, different trades in the same auction round occur at different prices.

Parameter	value
k	10
r	0.1
e	0.2
$s(1)$	1

Table 1: Parameters for the modified Roth-Erev learning algorithm

depending on whether the agent is a buyer or seller respectively. Table 1 summarises the parameter values used in our experiments.

As the basis of our multi-objective optimisation problem, we have adopted the three variables used in NPT, namely: *market efficiency*, *seller market-power* and *buyer market-power*. Here we present a brief summary of these variables (refer to [13] for details). Market efficiency, EA, is defined as the ratio (expressed as a percentage) of the total profits earned by all traders in the market, $PBA+PSA$, to the profits theoretically available to them in competitive equilibrium, $PBE + PSE$.

$$EA = 100\left(\frac{PBA + PSA}{PBE + PSE}\right) \tag{1}$$

Buyer market-power, MPB, is defined as the difference between the actual profits of buyers, PBA, and the potential equilibrium profits $PBCE$ for buyers, expressed as a ratio of the equilibrium profits.

$$MPB = \frac{PBA - PBE}{PBE} \tag{2}$$

Seller market-power is computed in the same way as buyer market power:

$$MPS = \frac{PBS - PSE}{PSE} \tag{3}$$

Market efficiency, EA, tracks how good our mechanism is at generating *global profit*, whereas the market-power indices, MPB and MPS track to what extent each group is better or worse off compared to a theoretical ideal market where traders bid truthfully, and an optimal allocation is made.

Our design objective is to increase market efficiency, whilst simultaneously keeping the market-power of both buyers and sellers to a minimum; we want to increase global profit but without giving unfair advantage to either buyers or sellers. We normalise each variable by mapping it onto the range $[0, 1]$, where 1 represents the optimal value of a variable and 0 represents the worst value. Variables are mapped using the following functions:

$$\widehat{EA} = \frac{EA}{100} \tag{4}$$

$$\widehat{MPB} = \frac{1}{1 + MPB} \tag{5}$$

$$\widehat{MPS} = \frac{1}{1 + MPS} \tag{6}$$

Given these, our design objective is then to pareto-optimise the vector:

$$\vec{F} = (\widehat{EA}, \widehat{MPB}, \widehat{MPS}) \tag{7}$$

by adjusting the auction mechanism. For the experiments in this paper, we weight market-power and efficiency equally and combine the three objectives in a linear sum:

$$F = \frac{\widehat{MPB}}{4} + \frac{\widehat{MPS}}{4} + \frac{\widehat{EA}}{2} \tag{8}$$

thus transforming the problem into an optimisation problem where we attempt to maximise the single scalar F.

For now, we have restricted our search of the mechanism design space to the *transaction pricing rule*, which sets the price of any given transaction as a function of the *bid* and *ask* prices submitted by buyers and sellers respectively. NPT uses a discriminatory-price k-double-auction transaction pricing rule [18], in which a different transaction price is awarded for each matched bid-ask pair in the current auction round. The price is set according to the following function:

$$p_t = kp_a + (1 - k)p_b \tag{9}$$

where p_t is the transaction price, p_a is the ask price, p_b is the bid price and k is a parameter that can be adjusted by the auction designer. In the original NPT experiments k is taken to be 0.5.

Our ultimate aim is to investigate if there are alternatives to the k-double-auction rule that perform well, not necessarily under equilibrium conditions, but when agents play Roth-Erev derived strategies; that is, adaptive strategies derived from a *cognitive* model of human game playing.

In our experiments, we consider the space of all possible pricing rules that are functions of p_a and p_b. We represent each function as a Lisp s-expression, and we use Koza genetic-programming [8] to search this space.

One might ask why we are using genetic-programming to search such a vast space, when we could simply restrict attention to the k-double-auction pricing rule, and search for pareto-optimal values of k. The reason we use genetic-programming, is that we see this as a general method of representing *arbitrary* mechanism rules, not just those that can be neatly parameterised. In this particular case, we have chosen an aspect of the auction design that can be so simply parameterised, so that we can compare the performance of the genetic-programming search against a brute-force search of different values of k. In the following section we use a brute-force search of k to get an approximate view of the fitness landscape that our genetic-programming search will encounter. In future work, we will use genetic-programming to search for additional rules governing the auction mechanism, for example rules governing allowable bids, and rules governing the matching mechanism.

4 Mapping the landscape

We ran the auction with particular values of k for 100 rounds with 10,000 different supply and demand schedules constructed by assigning each agent a random private value from a uniform distribution in the range $[30, 1000]$. The market variables under observation are averaged over these 10,000 different schedules. We carried out these 100 rounds with 10,000 different supply and demand schedules for 100 values of k at increments of 0.01. Figure 1 shows the mean fitness for each k value.

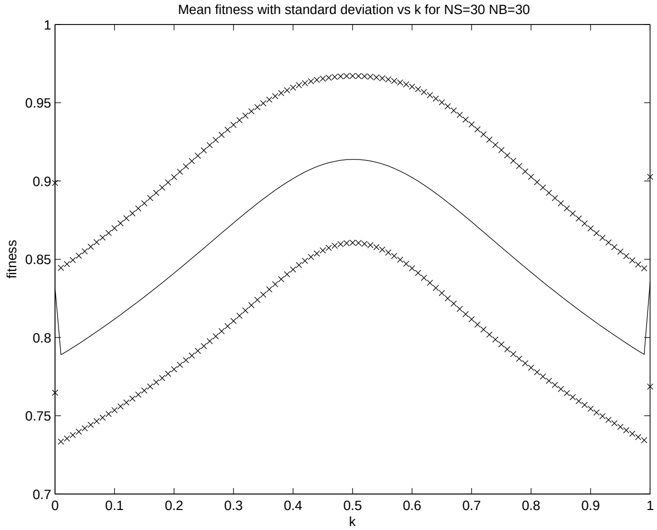

Figure 1: Mean fitness plotted against k for a large market with 60 traders

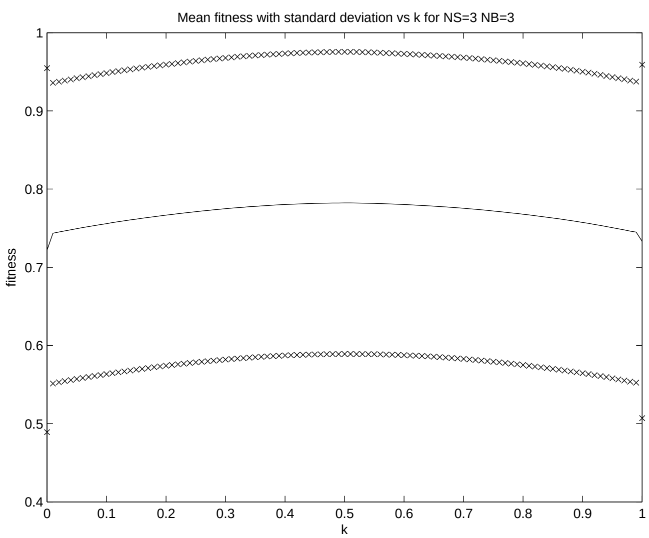

Figure 2: Mean fitness plotted against k for a small market with only 6 traders.

This mapping of our optimisation variables at different values of k gives us an idea of the fitness landscape in which the genetic programming technique is evolving the auction rule. A qualitative interpretation of this data would suggest that pricing rules corresponding to values of k close to 0.5 should be selected by the GP experiment.

5 Evolving pricing rules

We represented each function as a Lisp s-expression, and we used Koza's basic genetic programming [8] with the parameters given in Table 2 to search this space. We made use of a Java-based evolutionary computation system called ECJ.[3] ECJ implements a strongly-typed GP [12] version of Koza's [8] original system. For the GP experiments in this paper, the standard Koza parameters were used in combination with the standard Koza GP operators, with the addition of a small amount of parsimony pressure (applied with probability 0.005) in order to counter the effects of GP code bloat.

Parameter	value
Population size	4000
Selection	Parsimony Binary Tournament
Cross-over probability	0.9
Reproduction probability	0.1
Parsimony size probability	0.005
Cross-over maximum tree depth	17
Grow maximum tree depth	5
Grow minimum tree depth	5

Table 2: Koza GP parameters

Our function-set consisted of the terminals $ASKPRICE$ and $BIDPRICE$, representing p_a and p_b respectively, together with the standard arithmetic functions, + - * /, and a terminal representing a double-precision floating point ephemeral random constant in the range $[0, 1]$. Our fitness function is given by equation 8.

As in Section 4, market outcomes for each pricing rule were computed by simulating agents equipped with the Roth-Erev learning algorithm. We used the same parameters and the same numbers of buyers, 30, and sellers, 30, and 100 auction rounds, but with only 100 different supply and demand schedules, constructed by assigning agents different private values drawn randomly from a uniform distribution in the range $[30, 1000]$ each time an individual pricing rule was evaluated.

Figure 3 shows the actual pricing rule that was evolved after 100 generations (where $ASKPRICE$ is p_a and $BIDPRICE$ is p_b). This has been algebraically-simplified[4], but as can be seen it is still far from straightforward, something that is not surprising given the way that standard genetic programming approaches handle the evolution of the s-expressions that make up a program. Plotting the surface of the transaction price as a function of p_b and p_a, given in Figure 4, and comparing it with the surface for:

$$0.5p_a + 0.5p_b$$

given in Figure 5 shows that these two functions are approximately equal apart from a small variation when the ask price is very small or when the ask price is equal to the bid price. Thus the GP experiment has effectively evolved a pricing rule for a discriminatory-price k-CDA with $k = 0.5$—exactly the rule we used when establishing the fitness landscape.

These results suggest that the approach we are adopting is a reasonable one — we have managed to evolve a rule which in terms of the prices it sets, is close to a well established rule from the economics literature. The results also support the

[3] http://www.cs.umd.edu/projects/plus/ec/ecj/
[4] Using the MATLAB symbolic math toolbox

$$\frac{11913735914933551833672672699061}{20282409603651670423947251286016}\, p_a + \frac{81810593176186846190945933553906486857062204135}{182687704666362864775460604089535377456991567872} \left(-\frac{3437050858577137}{18014398509481984} \left(p_a - 2\, p_b \; \ldots\right.\right.$$

Figure 3: The first terms of the derived pricing rule .

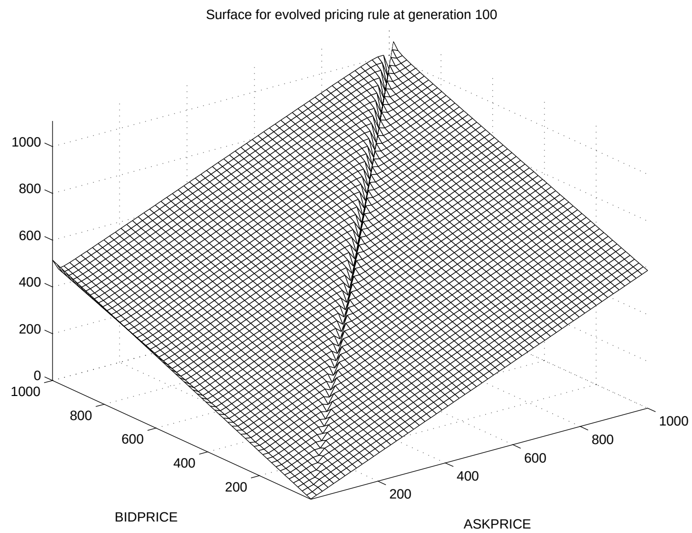

Figure 4: The transaction price set by the evolved auction rule.

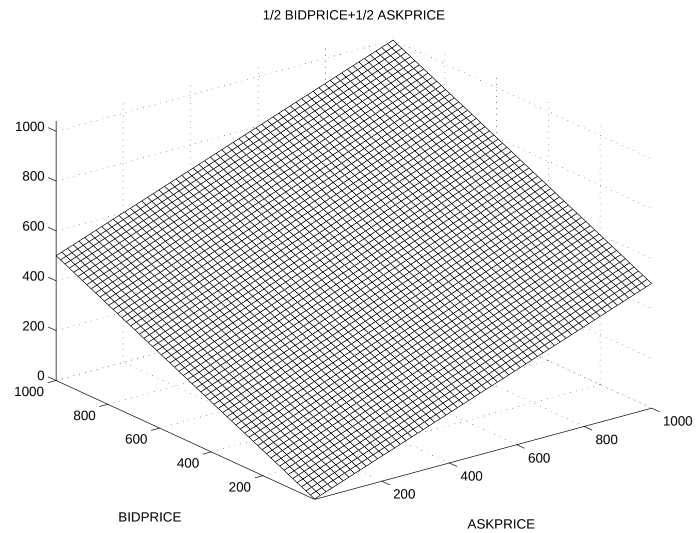

Figure 5: The transaction price set by the rule $0.5p_a + 0.5p_b$.

existing k-double auction rule since our GP search through the space of all functions of the bid and ask price has converged on a version of the k-double auction rule. This is in contrast to the results obtained by Cliff [1], which discovered a new form of auction between classical buy-side and sell-side auctions.

Although the fitness landscape for this benchmark problem is very simple, we see this as a means of validating our design technique before we move on to more complex scenarios. Future work will investigate the use of this technique for more complex market scenarios, and will include other aspects of the auction design in the search space: for example, matching rules, bid validation rules and so on. We have already begun to map alternative market scenarios, for example where we have a very small number of traders, using a brute-force search of k; figure 2 shows the fitness landscape for a 6-trader version of the design problem. Future work will analyse the evolved rules for a number of different market scenarios, for example where we have many more buyers than sellers and visa versa.

6 Discussion

The work described here is part of a larger research effort aimed at creating techniques and methodologies for *computer-aided auction design*. We have thus far identified two promising techniques that could play a part in such a technology: co-evolutionary mechanism design [14], and the optimisation technique described in this paper. These approaches are not mutually exclusive; we envisage that they will complement each other, and indeed complement standard analytic approaches to auction mechanism design.

For example, the optimisation approach might be used to find a pareto-front of promising mechanisms that perform well when agents play adaptive strategies. The auction designer might then pick a few of these designs from the pareto front that look as if they meet the criteria in hand, and then subject them to standard game-theoretic equilibria analysis, thus using the optimisation technique as a method of reducing the search-space for manual analysis. Once a mechanism has passed equilibria criteria, it might then be subjected to co-evolutionary experiments to probe it for "non-strategic" weaknesses in the protocol.[5]

In this scenario the "prey" population would be pre-populated with our candidate mechanism, and the "predator" population would be pre-populated with equilibrium bidding strategies; the predator population might then find non-strategic weaknesses in the auction population thus driving it it to more robust areas of the design space. This whole process of:

1. identify promising mechanisms through search;
2. pick and solve for equilibrium solutions; and

[5] By a non-strategic weakness we mean a weakness in the mechanism that occurs when game-theoretic assumptions are violated due to unforeseen real-world circumstances. For example, the game-theoretic analysis of a sealed-bid auction assumes that agents are not able to signal to each other, but the theory cannot account for features of the auction protocol that enable agents to use the auction protocol *itself* to send covert signals. It was just such a weakness in the German radio-spectrum auctions that seems to have allowed Manesman and T-Mobile to use the low-order digits of their bids to signal to each other and form a collusive price-fixing strategy[7]. Similar alleged occurrences were reported in the US spectrum auctions [3]

3. use co-evolutionary learning to identify non-strategic weaknesses,

might then be iterated through until no further weaknesses are discovered. At the end of the design process we hope to have auction mechanisms that: perform well against adaptive, possibly non-equilibrium, strategies; that perform well in equilibrium; and are robust against non-strategic predatory behaviour.

7 Summary

In this paper we have reported the results of two experiments in which we have examined the use of different pricing rules in a discriminatory price double auction. In the part of this work we used adaptive buyers and seller agents to evaluate the effect of changing the parameter k in the k-double auction pricing rule. The second part of the work then successfully used genetic programming to automatically acquire a transaction-pricing rule. The method we describe here could be used to automatically generate a discriminatory pricing rule for a continuous double auction to meet other specific socio-economic requirements.

References

[1] D. Cliff. Evolution of market mechanism through a continuous space of auction-types. Technical Report HPL-2001-326, HP Labs, 2001.

[2] P. Cramton, A. E. Khan, R. H. Porter, and R. D. Tabors. Pricing in the California power exchange electricity market: Should California switch from uniform pricing to pay-as-bid pricing? Technical report, California Power Exchange, 2001.

[3] P. Cramton and J. A. Schwartz. Collusive bidding: Lessons from the FCC Spectrum Auctions. *Journal of Regulatory Economics*, 17:229–252, 2000.

[4] I. Erev and A.E. Roth. On the need for low rationality, cognitive game theory: reinforcement learning in experimental games with unique, mixed strategy equilibria (mimeo).

[5] 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., 1993.

[6] J. K. Goeree and C. A. Holt. Ten little treasures of game theory and ten intuitive contradictions. *American Economic Review*, 91(5):1402–1422, 2001.

[7] P. Klemperer. How (not) to nun auctions: the European 3G telecom auctions. *European Economic Review*, 46(4–5):829–845, 2002.

[8] J. R. Koza. *Genetic programming: On the programming of computers by means of natural selection*. MIT Press, 1993.

[9] P. Krugman. Frank thoughts on the California crisis (mimeo), 2002.

[10] R. P. McAfee and J. McMillan. Analyzing the airwaves auction. *Journal of Economic Perspectives*, 10:159–176, 1996.

[11] J. McMillan. Selling spectrum rights. *Journal of Economic Perspectives*, 8:191–199, 1994.

[12] D. J. Montana. Strongly typed genetic programming. Technical Report #7866, Bolt Beranek and Newman, 10 Moulton Street, Cambridge, MA 02138, USA, July 1993.

[13] 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.

[14] S. Phelps, P. Mcburney, S. Parsons, and E. Sklar. Co-evolutionary mechanism design: A preliminary report. In J. Padget, D. C. Parkes, N. M. Sadeh, O. Shehory, and W. E. Walsh, editors, *Agent-Mediated Electronic Commerce IV*. Spinger Verlag, Berlin, Germany, 2002.

[15] S. Phelps, S. Parsons, P. McBurney, and E. Sklar. Co-evolution of auction mechanisms and trading strategies: Towards a novel approach to microeconomic design. In *Proceedings of the 2nd Workshop on Evolutionary Computation and Multi-Agent Systems*, New York, NY, 2002.

[16] A. E. Roth. The economist as engineer: game theory, experimentation, and computation as tools for design economics. *Econometrica*, 70(4):1341–1378, 2002.

[17] A. E. Roth and I. Erev. Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *American Economic Review*, 88(4):848–881, 1998.

[18] 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.

[19] P. R. Wurman, W. E. Walsh, and M.P. Wellman. Flexible double auctions for electronic commerce: theory and implementation. *Intern. J. of Decision Support Systems*, 24:17–27, 1998.