A Multi-Agent Model of the UK Market in Electricity Generation

Anthony J. Bagnall and George D. Smith, *Member, IEEE*

Abstract

The deregulation of electricity markets has continued apace around the globe. The best structure for deregulated markets is a subject of much debate, and the consequences of poor structural choices can be dramatic. Understanding the effect of structure on behaviour is essential, but the traditional economics approaches of field studies and experimental studies are particularly hard to conduct in relation to electricity markets. This paper describes an agent based computational economics approach for studying the effect of alternative structures and mechanisms on behaviour in electricity markets. Autonomous adaptive agents, using hierarchical learning classifier systems, learn through competition in a simulated model of the UK market in electricity generation. The complex agent structure was developed through a sequence of experimentation to test whether it was capable of meeting the following requirements: firstly that the agents are able to learn optimal strategies when competing against non-adaptive agents; secondly that the agents are able to learn strategies observable in the real world when competing against other adaptive agents; and thirdly that cooperation without explicit communication can evolve in certain market situations. The potential benefit of an evolutionary economics approach to market modelling is demonstrated by examining the effects of alternative payment mechanisms on the behaviour of agents.

Index Terms

adaptive agents, learning classifier systems, electricity market.

I. Introduction

The current trend in electricity industries around the world is towards increasing competition through deregulation and privatisation. For example, movement towards one large EU-wide market began in 1996 with EU Directive 96/92, which sets out common rules and targets concerning competition, unbundling and transparency. However, the particular nature of electricity generation and supply means that some form of regulation and market-mechanism-based control of a privatised market will always be necessary. In order to meet demand, to maintain the viability of the network and to ensure that generators are paid correctly for power they have generated, decisions need to be made about the nature of these mechanisms. Unfortunately, it is unclear what process one could use to achieve these goals and at the same time deliver some benefit to the consumer in the form of reduced electricity costs.

This paper describes research on a project sponsored by the National Grid Company (NGC) and conducted in the period 1996-2000. This paper extends the work published in [6], [5], [4], [7], and contains the following information not previously published. Firstly, the overall aims and objectives of the project are presented and, by encompassing all the subtasks presented in previous work, allows the opportunity to examine and analyse the results in the context of the market as a whole. Secondly, a more detailed, formal description of the market mechanism (Sections II and III) and the agent architecture (Section IV) are presented for the first time. Thirdly, in addition to a more complete description and analysis of the results published in [6], [5], [4], [7], new results and analysis of the characteristics of the market cycle are described in Section V. Finally, the conclusions presented in Section VI are described in the context of the project as a whole and, as such, are previously unpublished.

Recently, there has been a growth in the application of agents to simulate market scenarios (see, for example [26], [1], [17]). The aim of this project was to examine the feasibility of using autonomous, adaptive agent models to simulate generators in the UK electricity generation market in order to gain insights into the effects of certain market mechanisms on the bidding strategies.

This study is based on the pre-2001 UK market for electricity, details of which are presented in Section II. In 2001, the New Electricity Trade Arrangement (NETA) came into force, replacing the trading rules used in this study. The fundamental differences between the new system and the old are:

- The NETA directly involves electricity suppliers in addition to electricity generators.
- The NETA involves forward contracts between buyers and sellers of electricity (up to one year in advance of actual generation).
- Under the NETA, there is a daily balancing mechanism of bids and offers to deal with differences between contracted generation and demand and to adjust localized power flows to meet the constraints on the transmission network.

Details of the NETA can be found at the website of the Office of Gas and Electricity Markets [22].

An agent based simulation of the electricity market under the NETA was presented by Bunn and Oliveira [11]. This research demonstrates well the potential of agent based simulations. Other than the changes to the market rules and structure, the key differences between [11] and the work presented in this paper concern the complexity of the agent architecture and the nature of the information used to learn bidding strategies. The learning process of the agents used in [11] is based purely on historical market information, whereas the learning process of the agents used in this study is based not only on their own past profit performance but also on variables external to the behaviour of the agents, such as demand. Our agents are therefore presented with a more complex learning task, in that they attempt to learn and generalise over a large number of different environments. A more sophisticated learning algorithm must therefore be developed.

Bower and Bunn [10] also use an agent-based computational approach, utilising simple reinforcement learning algorithms to evolve supply strategies, to study the effects in an electricity market of moving from a uniform-price auction to a discriminatory-price auction. Nicolaisen et al. [19] adopt an agent-based approach to simulate a restructured electricity market in which prices are set by a discriminatory-price double auction, i.e. sellers and buyers both make price offers.

In the research presented here, however, our agents are simulating generators (sellers) in a pre-NETA England and Wales electricity market. Although no longer directly relevant to the current situation in England and Wales, our findings are nevertheless interesting from both a computational perspective, with our agents incorporating multiple learning mechanisms and case-based memories, and an economic perspective, due to the general observations of our agent behaviour and the fact that variations of System Marginal Price (SMP) (defined in Section II) have been widely adopted in other electricity markets.

A description of the model of the pre NETA market mechanisms is given in Section III. The agents represent the generating companies in the model. They use learning classifier systems in conjunction with other learning mechanisms to evolve strategies for bidding in the simulated market. A description of the agent architecture is presented in Section IV.

The potential for agent models was explored through a sequence of experimentation to meet the following objectives.

Demonstrate that the adaptive agents can learn optimal strategies in static environments.

The first set of experiments demonstrate that, using the architecture described in Section IV, agents are able to learn optimal strategies when competing against non-adaptive (static) agents following strategies modelled on those observed in the real world. Details of these experiments are published in [5] and [6] and are not reported here because of space constraints.

Show that the simulated market demonstrates recognisable trends and that agents can evolve behaviours observable in the real world.

The second set of experiments, reported in [4], is designed to determine to what extent the simulated market behaves in ways analogous to the real world market, and whether or not the agents learn bidding strategies that can be identified in the real world behaviour of the generating companies. It is important to be able to relate the agent behaviour and the trends in the simulated market back to the real world. The type of *evolutionary economics* approach used in this study is still relatively novel and controversial (see [21], [23] for examples of the principle). The first step in convincing those with knowledge of the market that there is any potential worth in an agent based model is the ability to illustrate that the agents can learn to behave in recognizable ways and that the market as a whole performs in a manner consistent with conventional theory. In Section V-A, we present an analysis of the patterns of behaviour of the agents and the market trends in the simulation.

Analyse, in the context of bidding strategies, how changes to the market mechanisms affect agent behaviour.

The ultimate benefit of an agent based economic simulation is the ability to investigate how alterations in market conditions affect the behaviour of agents which have the ability to learn to behave in ways analogous to generating companies. Eventually, this type of model could provide a tool to experiment with the effect of changes in market structure and to investigate regulator influence on patterns of generator behaviour. When the market was established, the justification for some of the mechanisms used was qualitative, and arguments about certain features continue to this day. There is no obvious best way of structuring an electricity market, but, despite the potential difficulties, deregulation is occurring around the world, most notably in America, where the electricity industry is currently being restructured to “give all American electricity consumers the right to choose among competitive providers of electricity” [13]. The effects of deregulation are not always as predicted. For example, in California, the first year of deregulation saw prices rise by 379 % and generating companies exploiting network constraints to gain excessive payments.

Understanding how the particular mechanisms employed in the market may affect generator behaviour is of great importance and autonomous, adaptive agents represent a potentially useful method of achieving this understanding.

In Section V-B, we illustrate the potential usefulness of this type of model as an advisory tool to decision makers in the electricity industry, by altering the structure of the market and observing the effect this has on agent behaviour. The market mechanism at the core of this particular analysis is the System Marginal Price (SMP) payment method.

Investigate whether the agents exhibit the potential for the evolution of cooperative behaviour.

From a more general research perspective, we are interested in investigating whether the agents learn to cooperate to increase mutual long term rewards at the expense of short term gain. The potential for cooperation exists in certain environments which share some of the characteristics of games like the Prisoners' Dilemma [2].

In the context of our model, cooperation is demonstrated by generating companies (agents) collaborating, in the absence of direct communication, in order to increase the price of meeting demand. This is undesirable as far as the consumer and regulator are concerned, and ultimately research should concentrate on mechanisms to minimize the potential for implicit collaboration. Before any case can be made for the introduction of market mechanisms to reduce the possibility of implicit collaboration, the system should be able to exhibit the occurrence of this behaviour under certain conditions.

We wish to discover firstly, whether the agents acting purely through short term reinforcement can learn to discover the long term benefits of cooperative bids, and secondly, under what conditions this cooperative behaviour is more likely to occur. In Section V-C we investigate several potential cooperative scenarios.

Finally, in Section VI, we summarize the results and discuss how the model could be extended.

II. Overview of the UK Market in Electricity Generation

This Section describes the key aspects of the UK market conditions and rules for operation prior to NETA. A more detailed description can be found in [7].

A generating company has one or more units that can generate electricity on to the National Grid. There are approximately 200 generating units regularly competing to generate power. Units are characterised by the fuel used to generate electricity. The most common types are Nuclear, Coal, Gas, Oil and Gas Turbine. The cost of generation is similar for units of the same type.

Each day a generating company must produce an *offer bid*, or just bid, for each unit it owns. A bid gives details of the conditions under which the generating company is willing to allow the unit to generate. A bid consists of up to 8 price parameters and 40 availability parameters. The price parameters describe a generation cost profile over the possible generation levels and give the cost of starting the unit (*start up cost*) and the cost of running the unit at zero generation (*no load cost*). The availability parameters describe the required generation profile and include:

- *minimum on time* - the minimum period a unit must run for;
- *minimum stable generation* - the lowest level at which the unit will generate;
- *maximum run up rate* and *maximum run down rate* - the rate at which the unit must increase (respectively decrease) generation when levels are below minimum stable generation; and
- *synchronizing generation*, the generation level at which the unit should synchronise with the grid.

The NGC collects the bids and uses them to form a schedule of generation for each unit. It does this in a two stage process.

Initially, the unconstrained schedule is formed. With the objective of minimizing cost, the scheduling software determines which units should be allowed to generate in order to meet forecast demand for the following day. The forecast demand curve consists of 48 half hourly estimates of the amount of electricity that will be required. This type of scheduling problem is commonly called the *unit commitment problem* [29]. The unconstrained schedule consists of a generation profile for every unit submitting a bid, and is constructed to satisfy the technical parameters submitted in the bid.

Subsequently, the constrained schedule is formed from the unconstrained schedule to ensure that constraints on the transmission network are not violated. Constraining a unit involves changing the generation profile from that specified in the unconstrained schedule. If a unit is constrained it may be constrained on (required to generate whether it was scheduled to or not) or constrained off (forced to not generate). Constraining is a key market feature and the mechanisms used to handle payment to constrained units have a large effect on generator behaviour. For example [9] shows how the structure of the Californian market may induce anti competitive behaviour.

The actual generation then follows that specified by the constrained schedule, with alterations made to account for differences between actual and forecast demand.

Once the actual generation profiles are known, the Settlement software calculates the payment due to each unit. Payment for power generated is based on two things: the unconstrained schedule and the *capacity premium*, an additional payment mechanism used in the UK market as an incentive to bid low in times of high demand. The unconstrained schedule is used to form the *System Marginal Price* (SMP). The SMP consists of 48 prices in pounds per MWh, and is set as the bid of the marginal generator for each period (i.e. the most expensive unit scheduled to generate).

The capacity premium is a function of *loss of load probability*, an estimate of the probability that availability cannot meet demand, and is added to the SMP to form the *Pool Purchase Price* (PPP) for each half hour slot. Unconstrained units are paid at PPP. The payment rules for units constrained either on or off are different. If a unit is constrained off, it is paid at *(SMP minus bid price)* for the power it was not allowed to generate despite being scheduled to do so. A unit that is constrained on is paid at bid price for the power it was constrained to produce.

The generation and transmission costs are then combined to form the *Pool Selling Price* (PSP), which is the amount paid by the electricity suppliers, who in turn pass the cost on to consumers. Hence, prior to NETA, the electricity suppliers had no direct influence on the amount they paid for generation and hence the amount paid by the consumer.

Bidding is obviously strongly influenced by external factors such as demand and constraints, but it is also affected by the type of generating unit. A summary of the observed real world bidding stategies by unit type are:

- *Nuclear units* are expensive to take off line and have little to gain by doing so. They tend to bid zero, or close to zero, in order to ensure generation.
- *Gas units* tend to bid low in order to get in the schedule and infrequently set SMP.

- *Coal units* tend to bid higher than gas stations and set SMP more often.
- *Oil/Gas turbine units* have higher generation costs but much lower start up costs. As a result, they tend to bid to be on during peaks and frequently set SMP.

III. The Simplified Market Model

Our approach to the task of modelling the generation market and generating companies has been to simplify the problem so that it is of a small enough size to allow both the analysis of behaviour and the potential for measuring performance against an optimal bidding strategy, yet still retain key features of the real world known to be important in generator bidding strategies.

A. Model of Generating Companies

The simplified model consists of n generating units, each controlled by an agent, i.e. each generating company has a single unit. A bid for a unit is a single quantity representing the Table A bid price. The Table A price is the average cost per MWh of generating during a day, allowing for start up costs.

The set of bids on any one day is denoted,

$$B =< b_1, \ldots, b_n > .$$

For experiments reported here, $n = 21$. Each unit is classified as one of four types: Nuclear; Coal; Gas or Oil/Gas Turbine. Unit type determines generation costs. Each agent has three cost parameters which are used to calculate profit from the payment and the generation profile. These are fixed costs, unit generation cost and start up cost. Fixed cost is a daily charge incurred independent of the generation level. Unit generation cost is the cost of generating a single megawatt for an hour (MWh), and start up cost is the cost of restarting the generator after it has been taken off line.

In our model, there are 5 nuclear units, which are set with low unit generation costs, high fixed cost and high start up cost. If the unit runs all day (i.e. no start up costs) at full capacity, the nuclear units need to receive an average payment of £3 per MWh (i.e. PPP averaged over the 48 time slots needs to be 3 or more) to make a profit. Gas units in our model have low generation costs. There are 6 Gas units which can make a profit at full generation if average payment is £6 per MWh. Coal units have higher generation costs than Gas units. There are 8 Coal units in the model which require payment at 8 per MWh for profitability. Oil/GT units have a high unit generation cost, low fixed cost and low start up costs.

Each agent has a generation capability of 3000 MW, except for the oil/GT units which have a capacity of 1000 MW. Hence the total capacity in the market is fixed at 57000 MW, an amount which always exceeds demand.

B. Model of Market Information and Mechanisms

Through consultation with the NGC and examination of historical bid data, the market variables deemed to have the most quantifiable influence on bidding behaviour were *Constraints*, *Demand* and *Capacity Premium*. The most obvious observed real world strategies depend on these factors. For example, an optimal strategy when a unit must be constrained to run is to bid as high as possible, since the constrained unit is paid at bid price. Data relating to these variables constitutes the environment information that is made available to the agents.

The two key market processes that need to be modelled are the scheduling algorithm and the settlement mechanism. The model captures the core techniques used without attempting to implement the more complex routines used to create the real schedules.

1) Market Information: The forecast demand for any day is a sequence denoted FD, the individual elements representing values for the 48 half-hour time slots,

$$FD = < FD_i \mid \quad 1 \leq i \leq 48 \quad > .$$

For experimentation we characterize the demand curve from real data, classifying the demand profile as typical of summer/winter demand and weekday/weekend demand.

The capacity premium is denoted *Cap*, and the particular values for the 48 half-hour time slots are written as

$$Cap = < Cap_i \mid \quad 1 \leq i \leq 48 \quad > .$$

Furthermore, by observing historic capacity premiums, we identify four distinct types of capacity premium curves.

The constraints are modelled in a way that attempts to imitate how the NGC actually handles limits on the network. When forming constraints, the NGC groups generating units together geographically, based on the restrictions on the transmission network. It then forms bounds on the generation amount within these groups in order to maintain network stability. We model this process in the following way: Suppose we have n generating units on the network, $U =< u_1, \ldots, u_n >$. On any day, these are partitioned into m non empty *constraint groups*

$$G =< g_1, \ldots, g_m | \; g_i \subset U, \; g_i \neq \{\}, \; g_i \cap g_j = \{\}, \; \forall i \neq j > .$$

Each constraint group can be in one of 3 states:

- Units in the group are unconstrained, i.e. there are no restrictions on the amount of power the units can generate. (State equals 0.)
- Units in the group are constrained on, i.e. there is a minimum total level of generation that is required from the group. (State equals 1.)
- Units in the group are constrained off, i.e. there is a maximum level of generation allowable from the group. (State equals 2.)

The state sequence specifies the state of each constraint group in G,

$$S =< s_1, \ldots, s_m \mid \; s_i = 0, \; 1 \; \text{ or } \; 2 > .$$

If a group is constrained on or off, there is a *constraint level* representing the *minimum* required total generation in MW of the group (if constrained on) or the *maximum* allowed total generation (if constrained off) for each time period in the commitment period. For clarity, we also store a constraint level if a group is unconstrained. The constraint levels for G are denoted

$$L = < l_1, \ldots, l_m > .$$

If a group g_j is in State 1 ($s_j = 1$) at level l_j, this means the total MW generation for each half hour time slot of all the units in group g_j must be at least l_j. If a group g_k is constrained off ($s_k = 2$) at level l_k, then the maximum total MW generation for each half hour for all the units in the group is l_k. In practice, we restrict each l_i to some small subset of $\Re$, using four levels for environments that are constrained on and four levels for environments that are constrained off. The constraint data, comprising groups, states and constraint levels, is denoted by the vector

$$Con = (G, S, L).$$

Thus the environment state is represented by (FD, Cap, Con) and the values these variables can take defines the environment variable space, E. The agent detector encapsulates this environment information using 10 bits, see Section III-B.5, so the number of possible states in the environment is 1024. A characteristic of the environment is that some states are much less likely than others, and these states represent the scenario where the agent can make the most profit. This mirrors the real world, where situations where units are constrained on in high demand are unusual but offer the potential for exploitation. In addition, some states will never occur (for example we cannot have maximum constraint payment in summer weekdays). As a result, there are 728 environment states with a probability of occurring greater than zero.

The environment state (FD, Cap, Con) and the bid data B are used to form the schedules. A *generation profile* or *genset* for unit i,

$$gp_i = < y_i^1, \ldots, y_i^{48} >,$$

specifies the level of generation (in MW) for each of the half hour time slots of commitment period. A *schedule* is a sequence of generation profiles for all units,

$$< gp_1, \ldots, gp_n >,$$

i.e. a schedule specifies the generation level of every unit for all 48 time slots.

2) Unconstrained Scheduling: The unconstrained scheduling unit uses FD and the bids, B, to form the unconstrained schedule US. The real US is formed to meet demand while attempting to minimize costs by the use of an heuristic rule-based algorithm. We use a simple *merit order loading* method to form the US. The merit order loading method involves simply ranking the bids by price then loading each half hour time slot in ascending order of price until demand is met.

3) Constrained Scheduling: The unconstrained schedule, US, and the constraint data, Con, are passed to the constrained scheduling module which then alters the generation profiles in order to satisfy the constraints. Note that it is assumed that constraints are consistent with demand. This means the constrained scheduler does not check to ensure that demand is still met by the constrained scheduler. The process of calculating the constrained schedule is described in Figure 1.

```
$CS = US$
For each constraint group $g_i$ in $G$
if $s_i = 1$ (constrained on)
    for each time slot j=1 to 48
        Calculate $X$ = sum of $US$ generation of units in $g_i$
        Sort units in $g_i$ by $bid^A$
        while $X < l_i$
            increase generation in $CS$ of cheapest unit
                not at availability
            increase $X$ accordingly
else if $s_i = 2$ (constrained off)
    for each time slot j=1 to 48
        Calculate $X$ = sum of $US$ generation of units in $g_i$
        Sort units in $g_i$ by $bid^A$
        while $X > l_i$
            reduce generation in $CS$ of most expensive unit
                still in schedule
            decrease $X$ accordingly
```

Fig. 1. Description of the constraining process

4) Settlement: The settlement unit takes the US, CS and Cap and returns PPP, SMP and the payments, H, due to each generating unit. The settlement unit first determines SMP and PPP. These are both sets of 48 scalars representing a price in £/MWh for generation in each half hour time slot. SMP for each time slot in the commitment period is the bid of the last unit loaded into the unconstrained schedule. PPP is then formed by adding Cap to SMP. All unconstrained generation is paid at PPP. If a unit has been constrained to generate above its unconstrained level, it is paid at its bid price for that generation. If a unit's constrained generation is less than its unconstrained (i.e. it has been constrained off) it is paid for the power it is not allowed to generate at (SMP-incremental), and does not receive the capacity premium.

5) Agent Detectors and the Environment Message: The *environment message* is a 10 bit string, and the environment variables are mapped on to this bitstring in the following way:

- *bits 1-6.* Constraints:

  Bits one and two represent the constrained on group (i.e. 00 - no group constrained on, 01 - group 1 constrained on, 10 - group 2 constrained on and 11 - group 3 constrained on), bits three and four represent the constrained off group and bits five and six represent the constraint level.

- *bits 7-8*. Demand: 00 - Summer weekend, 01 - Summer weekday, 10 - Winter weekend, 11 Winter weekday
- *bits 9-10*. Capacity Premium: 00 - none to 11 - high.

C. Simulated Daily Cycle

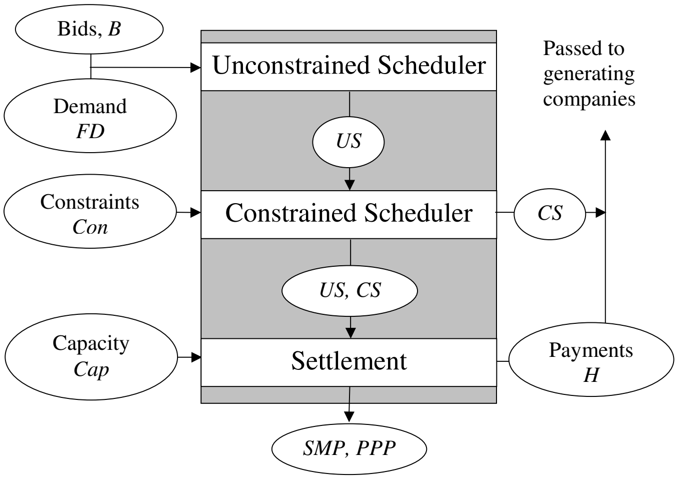

Fig. 2. Summary of the market processes in the simulated model

Thus, the market processes for any day, shown in Figure 2, can be summarized as follows:

1) **Inputs :** Bids (B), Demand (FD), Constraints (Con), and Capacity Premiums (Cap).
2) The **Unconstrained Scheduler** takes B and FD and produces an *Unconstrained Schedule* (US).
3) The **Constrained Scheduler** takes US and Con and produces the *Constrained Schedule* (CS).
4) The **Settlement Module** takes US, CS and Cap, calculates *SMP* and *PPP*, then outputs the Payments H, *SMP* and *PPP*.
5) **Outputs :** H and CS are passed back to each agent, which can then calculate its own profit. H, US, CS, *SMP* and *PPP* are passed to the system's statistics module which maintains data relating to past performance.

IV. The Agent Architecture

The agent architecture was developed through several evolutionary cycles by experimentation on single agent models [6], [5] and multiple agent models [4], [7]. In this Section we present the final model used in the multi-agent experiments, including refinements and details not described in previously published work.

When developing the agent architecture, certain initial conditions were imposed:

- *The ability to handle multiple objectives.* Our agents are primarily driven by the need to maximize profit on a daily basis, but in the real world, more complex objectives, such as maximizing market share and avoiding regulator punishment, also affect agent behaviour. Thus the architecture was developed with the potential for handling multiple objectives as quantified by alternative reward functions.

- *Limited memory resources.* The environment space for the model is relatively small, but it is important to recognise that more realistic scenarios will involve much larger environment and action spaces. The potential for scalability of the model is maintained by limiting the memory available to the agents, thus forcing them to attempt to generalise over the environments.

Each agent has to balance two related objectives. Objective 1 is to form a strategy that ensures that the agent does not lose money. Objective 2 is to find a rule set that maximizes profit over all environments. These objectives are quantified by two reward functions, R_1 and R_2 respectively, both of which are calculated from the profit. The profit, P, is simply the payment an agent receives from the settlement unit less the total cost of generation. R_1 is defined as

$$R_1 = \begin{cases} 100 & \text{if } P > 0, \\ 0 & \text{otherwise,} \end{cases}$$

and R_2 is

$$R_2 = \begin{cases} \frac{1000 \times P}{P_{max}} & \text{if } P > 0, \\ 0 & \text{otherwise.} \end{cases}$$

P_{max} is the profit the agent would make if it were to run all day to maximum capacity, and was paid at the maximum allowable price (31 in our model).

The task facing an agent is to form an estimate of the expected reward it will receive for each action in the current environment, then to choose an action based on these estimates.

We limit the space of possible bids an agent can make to one of 32 discrete values. The *Action Space* for agent i at any time t is $A_i = \{0, 1, \ldots, 31\}$. The agent action is mapped by a simple 1-1 mapping on to a unit's Table A bid price (in £/MWh). All the other bid parameters are fixed, and set to reflect the station type. Hence the effector copies the fixed parameters and the agent action into the agent bid.

The fundamental architecture we have adopted for each agent is a *Learning Classifier System*, LCS. A review of recent advances in the field of LCS can be found in [8] and [12]. Adaptations of the basic LCS structure have recently met with some success in areas such as robot control [18]. For our agents, the basic LCS structure used is based on Wilson's XCS [25] and we have adopted a complex agent structure similar to that used by Dorigo in [15], in that the agent has a high level action-decision controller and learning coordination mechanism. This structure sends signals to and takes input from two learning classifier systems (LCSs) and, depending on the current agent objective and utilizing some top level long term memory structures which we call *case lists*, uses this information to decide on a final action.

The two classifier systems, referred to as CS_A and CS_B, are used to help the agent meet the two objectives we have set it, firstly, to develop a rule set that will insure the agent does not make a loss, and secondly, to discover a rule set to maximize profit. CS_A starts with a set of randomly generated rules while CS_B starts with no rules at all. Each rule has prediction, error and fitness parameters. The prediction is an estimate of the expected reward for following a rule's action in matching environments. The error is an estimate of the variation of reward around the expected value. The fitness estimates the relative quality of the rule in relation to the current rule set, where the lower the error the fitter the rule.

Details of the LCS structure are discussed in Section IV-A.

A. The Learning Classifier Systems

The two LCSs operate by taking the current environment as input then forming *match sets*, respectively M_1 and M_2, of rules consistent with the current state. They then each form *prediction arrays*, respectively P_1 and P_2, from the parameters of the rules in each match set. The prediction array provides the system's estimate, $P(a)$, of the expected reward for each action, $a \in A$. The prediction arrays are passed to the controller (see Section IV-C).

Each LCS has three components: a performance system to produce a prediction array for any given input and rule set; a reinforcment system to alter the parameters of the current rule set based on the environmental feedback; and a rule discovery component to alter the current rule set to find new, potentially superior, rules.

The two LCSs that the agent employs have the same performance system and reinforcement mechanisms, but differ in the rule discovery methods used. CS_A uses reward function R_1 and is required to produce a concise rule set that is a good description of $E \times A \rightarrow R_1$, where E is the space of possible environments and A is the action space. CS_A is very similar to Wilson's XCS [25], but has some minor modifications described below. It was found to be difficult to design a LCS to produce a comprehensive description of $E \times A \rightarrow R_2$ when using a relatively small rule set. Instead, we designed the second classifier system to concentrate on the peak areas of R_2, i.e. on areas of the environment space where large profits can be made. These areas are mostly made up of areas where the agent is constrained on or off (messages matching 01******** and **01****** respectively) and where the capacity premium is high (********11) and these environments are generally much rarer than other environments. CS_A starts with 200 randomly generated rules while CS_B starts with no rules, is populated with rules from the *Good Case List* (GCL) when the agent switches to an aggressive state and can have a maximum of 400 rules (see Section IV-B for a description of the GCL).

Performance Component

A value in the prediction arrays P_1 and P_2 in position a is an error weighted average of the prediction values of classifiers in the corresponding match set M_i with action a. Each value $P_i(a)$ is the respective classifier's estimate of the expected reward for following action a in the current environment. If no rules in the match set advocate an action, it is assigned a prediction of 0.

Reinforcement Component

The action set is the set of rules advocating the chosen action and is formed at the beginning of the next time step, before the reinforcement stage.

Each rule has a prediction, error and fitness. The prediction and fitness are adjusted using the Widrow-Hoff delta rule. If the prediction at time t for rule i is denoted $p_i(t)$ then the Widrow-Hoff delta rule states that the prediction at time $t+1$ is

$$p_i(t+1) = (1-\beta)\cdot p_i(t) + \beta \cdot R,$$

where β is the *learning rate* and R is the external reward from the previous time step. During the fitness calculation a further two classifier parameters are calculated. The *accuracy*, κ_i, is defined as $\kappa_i = \exp[\ln(\alpha)\cdot(\epsilon_i - \epsilon_0)]$, where α and ϵ_0 are constants and ϵ_i is the error parameter of rule i. The *relative accuracy* of each rule in the action set, i.e. $c_i \in A(t)$, is κ'_i, where $\kappa'_i = (\sum_{c_j \in A(t)} \kappa_j)^{-1}\kappa_i$. The fitness $f_i(t+1)$ of rule i at time $t+1$ is then given by

$$f_i(t+1) = (1-\beta)\cdot f_i(t) + \beta \cdot \kappa'_i.$$

Following [25], the MAM procedure is also employed for the prediction. The MAM procedure involves using the simple average of the rewards until the rule has been a member of an action set at least $1/\beta$ times.

For the error parameter, we use the standard deviation of the rewards that a particular rule has received, divided by an estimate of the maximum reward deviation, which is 50 for CS_A and 500 for CS_B. One can view the error parameter as a longer term gauge of performance. In time, and depending on the value of β used, whereas previous poor rewards will have very little impact on current prediction values, some record of past mistakes will remain in the error parameter.

Discovery Component of CS_A

Both CS_A and CS_B, first described in [5], use a genetic algorithm (GA) as the rule discovery component, i.e. to generate new rules from old. The GA used by CS_A differs from that used by XCS only in the fact that it produces a single offspring rather than two (for implementation reasons), and in the manner of crossing the parents to produce a new action. Since the actions are integers, using a bitwise cross can often be very disruptive. Instead, an offspring either inherits one of its parent's actions or a new action is chosen by sampling a probability distribution centred around the average of the parents' actions. Mutation on the child's action can also occur and consists of either increasing or decreasing the action by one.

If CS_A receives a cover signal from the controller it forms a new rule with an action not currently present and not on the list of banned actions which may form part of the cover signal (if the current environment is on the *Bad Case List* - BCL). To do this we first examine the rule set to see if there is a very good rule that is very close to matching (in that it has only one non-wildcard bit different to the environmental message). A good rule is one that has a prediction value above the current 100 day percentage of correct actions and an error less than 1% of the maximum error. If such rules exist one is chosen probabilistically, with weighting in favour of more specific rules (fewer wildcards), and a new rule is created with the non-matching bit set either to match the environment or set to a wildcard. If no rules fit the matching criteria, a matching condition is generated randomly and a random candidate action is chosen. The rule thus created is then merged into the rule set using the normal merge mechanism.

Discovery Component of CS_B

CS_B, described in detail for the first time in this paper, is not meant to maintain full coverage of E, since it has limited resources. Instead it is supposed to maintain a good coverage of the action space for certain areas of the environment space. To achieve this we use a GA that acts on the whole rule set. The GA is triggered periodically and attempts to create one rule for each of the 32 possible actions. The GA only considers rules that have passed the MAM threshold (i.e. have been active at least $1/\beta$ times). For each action, two parents are then selected via roulette, with fitness set to the prediction values. If different rules are selected, then the condition of the child is created normally via single point crossover. If the same rule is selected twice a new action is chosen for the offspring (either by adding one to the action, subtracting one from the action or randomly choosing an action) and the child's condition is copied from the parent. Mutation proceeds as normal.

This GA method is designed to find rules for each action with conditions that offer maximum predicted reward while allowing some copying of conditions from one action to another. It also helps to maintain some diversity by making the creation of duplicates less likely. A rule created in this way replaces a rule chosen panmictically by probabilistically sampling a distribution derived relative to a value that is a function of a rule's prediction, error and age (to slightly reduce the probability of deleting younger rules).

If R is the current reward and P is the current value of the prediction parameter for a rule then the first error measurement, ϵ^1, is calculated using the standard update rule,

$$\epsilon^1 = (1-\beta) \times \epsilon^1 + \beta(|R-P|),$$

where β is the learning rate. ϵ^1 is an estimate of the mean absolute deviation, with greater weighting placed on more recently observed deviations.

The second error measurement, ϵ^2, is the standard deviation of the rewards received, which can be calculated by keeping a running total of the sum of the squared reward (denoted r_{S^2}), and the sum of reward (r_S), and is defined as follows:

$$\epsilon^2 = \surd\left(\frac{\left(r_{S^2} - \frac{r_S}{w}\right)}{(w-1)}\right), \qquad w > 1,$$

where w is the number of wins of the rule (i.e. the number of times it has received a reward). We use the two error measurements to attempt to discriminate between situations where variation in reward is caused by poor generalization and situations where the variation in one agent's reward is the result of variability in the other agents' bidding strategies. Since we are not allowing the agent to keep a record of past rewards received and other agent bids, we cannot hope to make a foolproof distinction between these possible causes of error. Instead, we attempt to introduce a general mechanism which reduces the probability of automatically discriminating against environments with high variability of reward.

Let $\epsilon^1_m, \epsilon^2_m$ be the maximum errors in the population, P_m be the maximum prediction parameter in the population and age_m be the age of the oldest rule in the rule set. The deletion value, d_{c_i}, of rule c_i is then

$$d_{c_i} = (P_m - P_{c_i}) + 20 \cdot \frac{\epsilon^1_{c_i}}{\epsilon^1_m} + 20 \cdot \frac{\epsilon^2_{c_i}}{\epsilon^2_m} - 5 \cdot (1 - \frac{age_{c_i}}{age_{max}}) - |\epsilon^1_{c_i} - \epsilon^2_{c_i}|.$$

d_{c_i} represents an heuristic quantification of the quality of a rule based on the prediction, errors and age of a rule (low values equating to better rules). The deletion values are then normalized to form a probability of deletion for each rule. We include a term for the difference, $|\epsilon^1 - \epsilon^2|$, of the two errors to decrease the probability of deleting rules where there may be evidence that the variation is caused by changes in opponent's strategies. The age term is included to decrease the probability of young rules being deleted.

B. The Case Lists

The case lists are a long term memory facility used by the controller to focus its resources on certain environments that seem, from past experience, to have an important role in meeting the objectives. There are two case lists, the bad case list (BCL) and the good case list (GCL), which are aids for meeting objective 1 and objective 2 respectively. Each case list consists of up to 20 cases. A case consists of a message string to identify the particular environment the case relates to, a prediction array and an array to count the occurrences of each bid. When a particular case occurs, the prediction for the action that the agent finally chose is updated using the Widrow-Hoff delta rule. Cases are added to the BCL by the controller when a frequently occurring environment triggers an action that yields a loss, and there are no actions in the prediction array of CS_A with a value at least as high as the selected action. Cases are added to the GCL when a particularly large profit is achieved. Once the list is full, the controller has the capability to replace one case with another.

C. The Controller

In [4], we outlined the function of the controller but did not provide a full description of the internal operations. The operations the controller performs are:

Op-A *Formulate an estimate of the expected reward for each action in relation to its competing objectives.*

It does this by combining the prediction array from the classifier, and the prediction array of the closest environment on the appropriate case list (by Hamming distance), with the combination being weighted so that the further away the case, the less effect it will have on the prediction array.

Op-B *Decide on which objective it is primarily interested in meeting and hence decide on an action.*

The controller bases its choice of the current primary objective on long term performance and the quality of the prediction arrays for the current environment. The need for exploration and exploitation is balanced by using a Boltzmann weighting to form a probability distribution over the action space. The probability of selecting action a is given by

$$P(a) = \frac{e^{\frac{f(a)}{t}}}{\sum_{a' \in A} e^{\frac{f(a')}{\tau}}},$$

where τ is the temperature, f is the normalized final prediction array and A is the action space. Temperature is used to balance between the need to exploit existing information, when the agent is confident in the accuracy of the prediction values, and to explore to get new information, when the agent has not experienced the current environment many times.

The temperature is determined dynamically by the agent by assessing how well it is meeting its current primary objective and how experienced the rules in the current match set are. The use of a temperature scheme to control the balance between exploration and exploitation has been frequently suggested, e.g. [25]. Our previously unpublished method of calculating the temperature is based on the principle that the temperature should decrease as the agent gains more information concerning the environment, unless the current match set is poor or there is evidence of increased variability in the reward function. The temperature range is from $\tau = 0$ (greedy exploitative strategy) to $\tau = 0.1$ (exploratory strategy).

If the current annual average reward is within 95% of the best annual reward received within the last five years, the agent exploits ($\tau = 0$). If, on the other hand, it is doing worse than this, it calculates the temperature based on the following variables:

Estimate of exposure

The agent first estimates the number of times the environment has been encountered with the current rule set. Let x be an estimate of the number of exposures to the current environment, calculated using the following formula

$$x = \sum_{c \in M} \frac{m_c}{2^{y_c}},$$

where y_c is the number of wildcards in rule c and m_c is the number of times c has matched an environment. If the environment is on a case list, it does not use this formula. Rather, it uses the explicit record of the number of exposures recorded with the case list. As x increases, the temperature will decrease, unless the error is also increasing.

Match set error statistics

The agent examines the rule set to find the lowest error of rules in the match set of rules passing the MAM threshold, denoted m_{ϵ^1} and m_{ϵ^2}. If no rules have passed the MAM threshold, the temperature is set to 0.1. The agent also finds the maximum difference between the two error measurements,

$$m_d = \max_{c \in M} |\epsilon_c^1 - \epsilon_c^2|.$$

If either m_{ϵ^1} or m_{ϵ^2} is high, we would like to explore more, the exception being when m_d is also high, in which case we want to decrease the exploration (to encourage the agent not to abandon cooperative solutions). The statistics $x, m_{\epsilon^1}, m_{\epsilon^2}$ and m_d form the basis of the temperature calculation. Firstly, we use the summation function

$$S = \frac{m_{\epsilon^1} + m_{\epsilon^2}}{x + m_d}$$

to produce a summary scalar of the variables of interest, then we use the squashing function

$$\tau = 0.1 \cdot (1 - e^{-S})$$

to map this summary function on to the range (0,0.1).

Op-C *Oversee alterations to strategy by exercising some control over the rule discovery processes of the learning modules.*

If the controller deems the prediction from CS_A to be unacceptable it can send a rule creation cover signal to both classifiers and receive an altered prediction array. The controller compares the best predicted reward with the percentage of environments on which the agent has made a profit over the previous 365 days, and deems the whole prediction array unacceptable if the best reward is less than 50% of the annual percentage.

The controller also maintains the case lists, i.e. adding cases to the lists and removing cases from the list. When a case is removed, rules may be added to the corresponding classifier to ensure that the information from the outgoing case is not lost.

V. Results

This Section describes the results of experimentation to meet the objectives identified in the introduction.

A. Learning Real World Behaviour

The first question to address is: can the agents learn bidding strategies that can be identified in real world behaviour? Specifically, interest lies in whether the agents approach the unconstrained strategies determined by station type, as described in Section II, and how they perform in constrained environments. Specifically, do they learn to bid higher when constrained on and lower when constrained off, and do they adapt to differing constraint levels?

The first experiment consists of a run of 100000 days. These results have been previously published in [4], although in less detail than here. Figure 3 illustrates the overall strategies adopted by the four station types during unconstrained days (environment matches 0000******). Note that Oil represents both Oil and GT units.

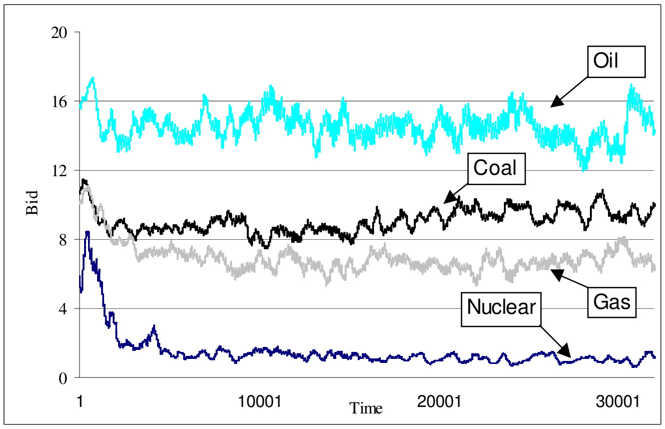

Fig. 3. 365 day moving average time series of bids by unit type for unconstrained environments. A data point is the average bid for units of that type for the day in question.

The data series were formed by taking the average bid by station type for each day then forming a 365 day moving average time series. This graph clearly illustrates that nuclear units are bidding at or close to zero, gas and coal units are bidding close to the level required for profitability and that oil/GT units are bidding high in order to capture peak generation, which is broadly equivalent to the behaviour observed in the real world. Further analysis shows that the nuclear units are fairly unresponsive to demand whereas the coal and gas units tend to increase their bids in times of high demand unless the capacity premium (the incentive to bid low) is at maximum, in which case they revert to lower bids. This behaviour is also broadly consistent with real world strategies.

In the initial 10000 unconstrained days, the coal and gas agents learn to bid, on average, at just above the level needed for profit. They then compete in order to remain fully in the schedule. First the average gas bid drops as the gas agents learn to rely on the coal units to set non-peak SMP. The coal units then suffer from not generating at these times and hence become more defensive, dropping their bids to get into the schedule. However, the result of this is a lower SMP, hence everyone suffers from reduced payments. The gas units respond by raising their bids. Gradually, the coal units appear to discover more stable strategies at around bids of £9 and £10, with occasional dips being met with a compensatory rise by the gas units (and vice versa).

We are also interested in an overview of bidding behaviour evolving in constrained environments. When a group is constrained on, the generation required can be at 4 levels, and broadly speaking it would be expected that the agents bid higher when the constraints are high (when more units are constrained on). The only exception to this is, when the capacity premium is maximum, it may be worth bidding low in order to get the bonus payment. Figure 4 illustrates the overall strategies adopted by showing the difference between SMP and the average bid of the constrained units.

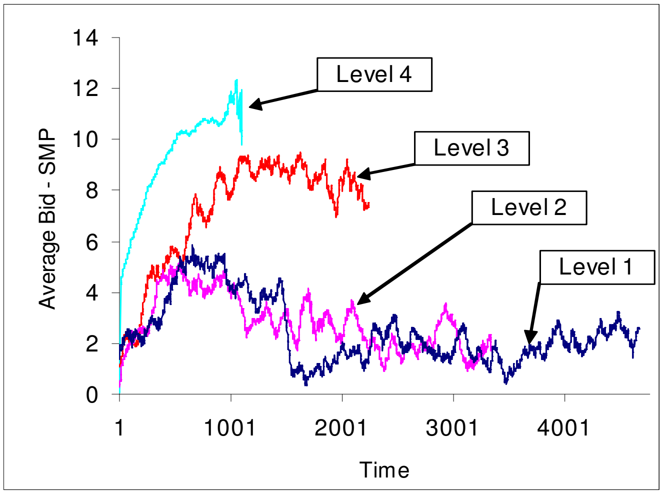

Fig. 4. For each constraint level the difference between the average bid of the constrained units and the average SMP for that day was calculated. The data shown is the 100 day moving average of this difference. Constraint levels occur with different probabilities, level 4 being the rarest and level 1 the most common.

Figure 4 indicates that the agents are learning to bid higher when they are maximally constrained on (level 4), with a steady increase in the difference between SMP and average bid. When constrained at level 3, the agents progressively learn to bid higher, but tend not to bid higher than 8 or 9 pounds above SMP. With level 1 and 2 constraints the agents seem to initially start cooperating to receive higher rewards, but, after exposure to approximately 1000 environments, progress is halted and bidding becomes closer to SMP. The difficulties of reaching a cooperative solution are discussed in Section V-C.

The overall strategies are as expected. This can be confirmed by examining part of the final rule set of one of the agents. Table I gives a representative sample of rules matching 01******** for unit 17, a coal unit. Each group consists of a balance of station types. Behaviour in constrained environments is generally independent of station type, since the optimal strategy is usually to maximize constraint payments irrespective of generation costs. It clearly illustrates that the agent has learnt the value of bidding at maximum (31) when the constraint level is maximum (environments matching 01**11****). There is a good spread of rules covering both level 3 and 4 constraint levels (i.e. matching 01**1****), illustrating the formation of a default hierarchy for these environments. There are fewer rules covering levels 1 and 2 as resources have been concentrated more on the higher constraint levels.

TABLE I

RULES FOR AGENT 17 MATCHING 01****** AT THE END OF A RUN OF 100000 DAYS (FROM LCS CS_B)

Condition	Bid	Pred	Condition	Bid	Pred
01*11101**	31	374.62	01*111**1*	22	309.43
01**11**1*	31	396.66	01**1001**	22	231.76
011*11**0*	31	384.20	01001***0*	22	288.51
01*011**0*	29	384.32	01001*1***	22	218.39
01*011****	29	384.51	01001**1**	22	239.69
01*01101**	30	390.31	01001****0	22	210.07
01001***0*	26	221.1	01*01001**	22	234.24
01001**1**	26	205.57	01001**1*0	22	238.68
01001***10	26	175.78	01*01****0	22	218.22
01*11101**	25	340.62	01**1*1*0*	22	216.31
01**10**0*	25	210.48	01001***0*	21	272.75
01**10**0*	25	207.88	01001***0*	20	230.61
01**10*10*	24	204.69	01111***1*	17	226.08
01001**1*0	23	202.02	*10*10**11	6	203.41
01**1000*1	22	283.85	*1***0**11	7	165.19
01001*1*0*	22	296.81	*1***0**11	11	210.63

B. Alternative Market Structures

In order to show the potential for this type of model, the next experiment is designed to test the effect on agent behaviour of changing market conditions. Some of these results have also been presented at conferences [4], [7] (in particular, Figures 5 and 6), although with less discussion than provided here. The boom/bust cycle shown in Figure 7 is previously unpublished.

The particular aspect of the market investigated is the payment calculation. SMP is the price of the last unit loaded onto the unconstrained schedule, and this price is paid to all units in the schedule. We are interested in seeing the effect on the system of agents being paid at their bid price rather than SMP. The main question arising is: does the removal of the safety net option of bidding low to accept SMP lead to agents learning cooperative behaviour patterns which result in higher average bids? A further run of 100000 days was conducted with the new payment calculation method for unconstrained generation. Figure 5 shows the bids by station type. It is apparent that, on average, each type of agent is bidding higher than in the previous experiment (illustrated in Figure 3). This experiment is interesting because it lends anecdotal evidence to support the decision to use SMP to calculate payments. In terms of the model, the ability to avoid complex game playing with the SMP approach means that the agents have less incentive to learn cooperative strategies.

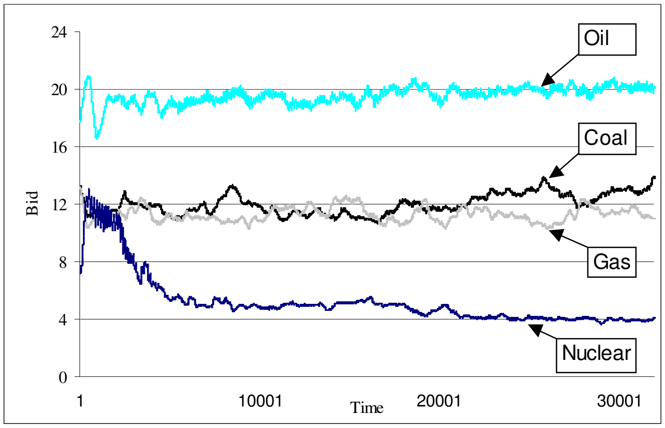

Fig. 5. 365 day moving average time series of bids by unit type for unconstrained environments when agents are paid at their bid price rather than SMP.

Whilst Figure 5 shows that altering the payment system so that agents are actually paid at their bid price rather than SMP makes the agents bid higher on average, the fact that the agents bid higher does not necessarily mean that the generation is more expensive. In order to examine the effect of the settlement method on the cost of meeting demand, we recorded the total payments made to the agents on each day. Figure 6 shows the 365 day moving average total payment for settlement at SMP and payment at bid price (no SMP) for the first 30000 unconstrained environments of an experiment over 100000 days.

Firstly, under both systems, the total cost of meeting demand is decreasing as the market becomes more efficient. Initially, as the agents' strategies are close to random, large payments will be incurred. The cost of meeting demand is much higher initially under the SMP system, since just one high bid in merit will result in large payments to all generators. After 20000 unconstrained days the payments are in the range 4 to 6 million, and further experimentation showed that payments tended to remain in this range after this.

Secondly, under both payment methods, the market exhibits the kind of volatility one would expect in a real world market of this kind. This illustrates that the agents are actively attempting to improve their performance rather than converging to stable strategies.

Figure 7 shows that, under the SMP system, the pattern of payments after 20000 unconstrained days seems to

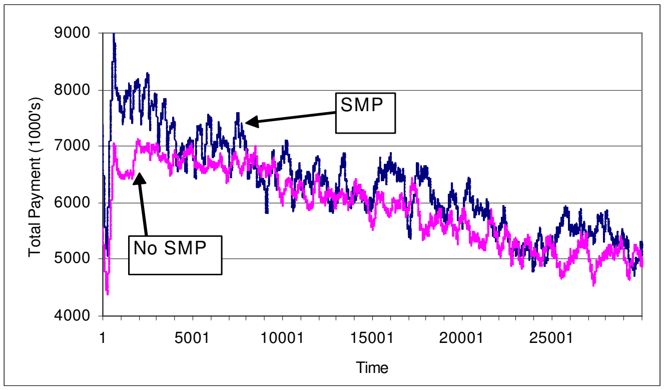

Fig. 6. 365 day moving average of total daily payments made to generators over 30000 unconstrained environments

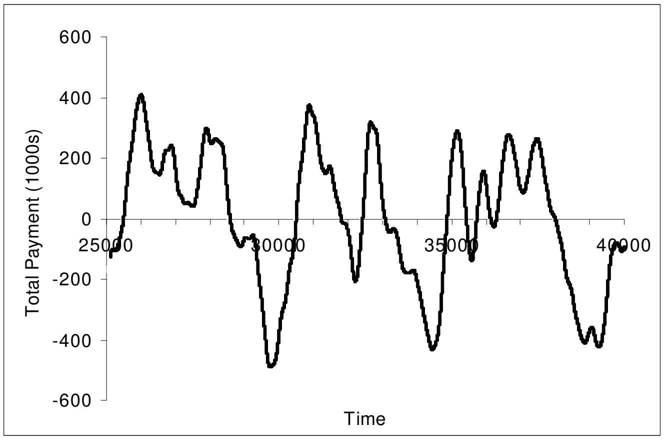

Fig. 7. Smoothed total daily payments for period 20000-40000 days. Figures have been normalised around the average for that period

exhibit the characteristics of a boom/bust cycle, in that rises in payments are sustained for a short period before the price collapses as competition drives the agents to optimize their own performance. Figure 7 shows three such market cycles with a period of approximately 5000 days.

Finally, Figure 6 suggests that the SMP settlement method is actually more expensive overall than payment at bid price, although the payments seem to be converging to approximately the same level (further experimentation indicated that the SMP method remained slightly more expensive). A closer examination of the data reveals the reason for this. Figure 8 and Figure 9 show the average reward received by the nuclear units and coal units respectively.

The actual reward levels are dependent on the system parameters, but it is clear that, under No SMP settlement, the nuclear units are performing worse and the coal units are performing better. With bid payment, the nuclear units have to actively find profitable bids, and this results in them being 'shifted' (brought in and out of schedule) more

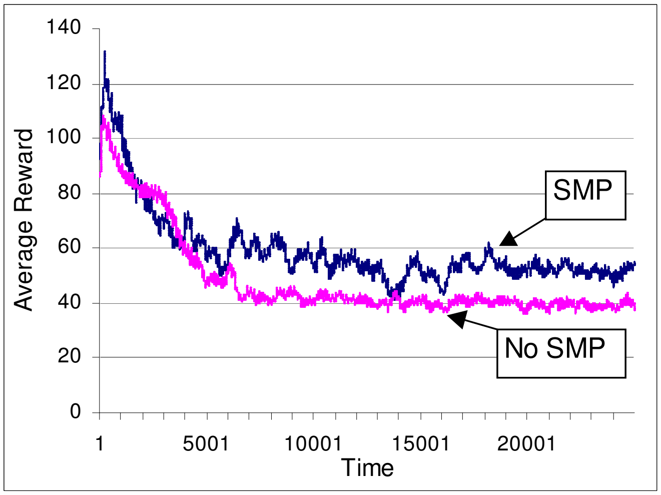

Fig. 8. 365 day moving average reward received by the nuclear units in unconstrained environments.

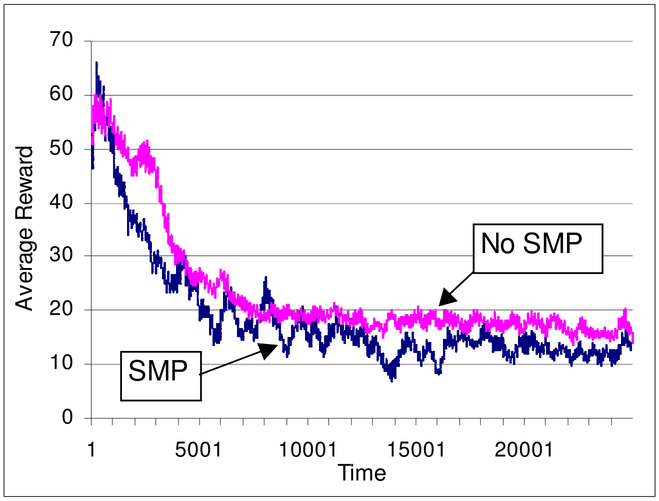

Fig. 9. 365 day moving average reward received by the coal units in unconstrained environments.

often. Nuclear units are expensive to shut down and restart, so the reward to nuclear units is reduced. They are receiving higher payments but incurring relatively higher costs also. It is not particularly desirable to have nuclear units coming in and out of schedule regularly and this was one of the reasons for using the SMP settlement method in the real world market. Our simulation seems to provide some support for this reasoning.

Figure 9 shows that the coal units are receiving higher payments under the No SMP system. The fact that they can no longer simply accept SMP by bidding low and running all day means they are more actively attempting to find profitable bids, resulting in the coal units being marginal more often.

There seems to be more variation in the reward under the SMP system, but this is probably just reflecting the fact that payments under an SMP system are more dependent on other agent behaviour.

C. The Evolution of Cooperation

In Section I, we identified the following question as worthy of investigation: can the agents learn to cooperate to exploit certain market structures and thus increase the overall cost of generating electricity?

Ultimately, we would like to use this kind of model to study under what conditions cooperation is likely to emerge. This would enable the market designers and regulators to examine the effect of introducing alternative mechanisms on the behaviour of simulated generating companies. However, cooperation in such a complex environment is both hard to define and difficult to achieve. The observation made in Section V-A, that agents are behaving in predictable ways in unconstrained environments, essentially means that the agents are not cooperating in order to maximize their mutual payoffs. Simplistically, if every agent bid the maximum bid every day, SMP would be maximum, they would all be partially scheduled and hence make larger profits. Such complex cooperation is unlikely to be achieved without some coordination, since there will always be an incentive for a single agent to defect. Yao [27] and later Axelrod [3] observed that mutual cooperation was hard for multiple agents to achieve in the n player Prisoners' Dilemma. Axelrod then looked at how the agents could learn to adopt social norms (i.e. all cooperate) through the ability to exact revenge for defection. However, in our model such direct communication between the agents would constitute an illegal anti-competitive act, hence it is inappropriate to include this type of mechanism in our model. It is unrealistic, therefore, to expect the agents to achieve this kind of cooperative equilibrium. However, we can investigate whether alternative forms of cooperation can emerge.

More complex forms of cooperation can also increase profits: if a certain proportion of agents bid high, the others get an immediate payoff advantage, thus if the agents take turns in making these large bids, they can all increase their long term profit. Scenarios that include multiple levels of cooperation and "reputation" (a form of indirect reciprocity as defined in [20] and described in [28], [14]) could usefully be adapted to model electricity markets.

C1: 7 Player Games

We are interested in whether the agents have the capacity to learn to cooperate, and whether there is any evidence of this cooperation emerging. To illustrate that the agents can learn the benefit of cooperative strategies, we examine the behaviour of agents in a subset of environments, where behaviour is easier to analyze and cooperation has a larger benefit. We concentrate on environments that are constrained on. These environments are, in effect, 7 player games rather than 21 player games, since the bids of the agents in a different constraint group do not have much affect on the reward function of those in the group that is constrained on. Figure 4 shows that, on the macro level, the agents are learning to cooperate when constrained on, in that they learn to bid higher, but ideally we would like the agents to be able to learn a form of cooperation analogous to the Prisoners' Dilemma, in that they forgo the opportunity for an immediate higher reward for the potential of longer term higher average reward (i.e. they learn the benefit of finding Pareto optimal solutions which are not dominant or Nash equilibrium solutions for the one off game).

Consider environments where group 1 is constrained on at level 3 (environments matching 01**10****). In these environments, units in group 1 are constrained on in order of price until 16500 MW capacity is reached for the group. The total capacity of the group is 19000 MW, thus each agent wants to bid as high as it can, but not to bid higher than all the other agents since the highest bidding agent will only generate 500 MW. However, if two agents make the same bid they are both partially constrained on and thus share rewards. These environments have similarities with the Prisoners' Dilemma, where cooperation corresponds to two or more agents in the group making the same large bid. So if two agents make the maximum bid, then all agents in the group will benefit, but those bidding just lower than those making the maximum bid will receive greater payment. Thus an ideal cooperative solution would be one where each agent takes turns to be one of the agents bidding maximum.

Analysis of the bid data used to create Figure 4 shows that of the last 1000 environments matching 01**10****, 221 met the cooperation criteria of the highest bid in the group being the same, although generally at a bid less than the optimal cooperative bid of 31 (20-25 is most common). Although this seems discouraging, further examination reveals that in a further 351 environments from the last 1000, the cooperation criteria would have been met (i.e. several agents made the same high bid) except for the bid of a single agent, which attempted an even higher bid in the hope of greater profit (a hope that is not realized). It seems the agents are attempting to reach a higher equilibrium cooperative bid by alternatively attempting higher bids, but the dynamics of rule creation, the explore/exploit balance and the limited size of the rule set stop the agents learning to fully exploit the cooperative bids of higher potential.

Higher bids have an associated higher risk in non constrained environments which a rule may mistakenly also cover. This means that the rules suggesting high actions will, on average, have higher level of error and hence have a higher deletion probability. However, this very restriction may well aid the emergence of the cooperative equilibrium at a lower level. The fact that the agent cannot store a complete environment/action space representation means the agent will generally only pick from a subset of the possible actions. Table I illustrates that there is a preponderance of rules with action 22, a common cooperative bid. This in itself does not make the action more likely to be selected, but the high associated fitness and prediction mean the action will maintain a strong presence thanks to the operation of the genetic algorithm.

C2: 2 Player Games

In the previous subsection we observed some behaviour which could be considered cooperative, but the agents were unable to maintain cooperative solutions for a variety of reasons. In this subsection we alter the constraint groups and perform further experiments. This serves to meet two of our stated objectives, firstly to examine the affect of altering market structure and secondly to study the emergence of cooperation in the multi-agent system. The question we are asking in terms of the model is; does having more tightly coupled constraint groups increase the likelihood of anti-competitive collaboration? However, this question is somewhat premature, since we have not properly illustrated that the agents are able to reach cooperative solutions in any environments. Because of this, we make the grouping as simple as possible.

In previous experiments, units were constrained in fixed groups of seven units. We change this set up so that one of the constraint groups consists of only two coal units. This makes it easier to examine the nature of the game for certain environments. In addition, it is assumed that the agents would find it easier to learn how to cooperate with just one other agent to compete against. These two agents can be constrained on at one of 4 constraint levels, 1000 MW, 3000 MW, 4000 MW or 6000 MW. This means that if the group is constrained on, the two units must generate at least this much power throughout the day between them.

Table II shows part of the reward function for one of the games of particular interest. Note that the rewards are calculated assuming the other adaptive agents (not in this constraint group) bid at production cost, and so the actual rewards received when the agents are in merit (i.e. when they bid low) will vary from those given in Table II. In this game the two units need to generate at a minimum of 4000 MW (i.e. the constraint level is level 3). If both agents make the same bid, they are both constrained at 2000 MW, but if one bids lower than the other, the agent bidding lower generates at 3000 MW and the other can only generate 1000 MW. Thus, if both agents bid 31, they both receive a reward of 333. However, if one agent bids 31, the other agent can gain an immediate advantage by bidding 30 and receiving a reward of 429. Making such a bid will discourage the agent who bid 31 from making future bids of 31, since it will receive less reward. This game has similar characteristics to the Prisoners' Dilemma, and ideally we would like the agents to learn to find a mutually beneficial cooperative solution such as (31,31).

TABLE II

Reward matrix for environment 0100100000 for the two agents in constraint group 1 when all other agents bid at production cost. The top figure is the reward for Agent 1, the bottom for Agent 2

	Agent 2							
Agent 1	0		16		25		30	31
0	47 47	...	47 103	...	47 174	...	47 214	47 222
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
16	103 47	...	154 154	...	206 174	...	206 214	206 222
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
25	174 47	...	174 206	...	262 262	...	349 214	349 222
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
30	214 47	...	214 206	...	214 349	...	321 321	429 222
31	222 47	...	222 206	...	222 349	...	222 429	333 333

The fundamental difference between the game shown in Table II and the Prisoners' Dilemma is that the actual act of cooperation or defection is not constant. A cooperative bid is a bid that is the same as the other agent's bid, whereas a defection is bidding below the other agent's bid. Thus a bid of 30 yields a mutual reward if the other agent bids 30, but is a defection if the other bids 31. The only bid that is cooperative independent of the other agent's bid is 31. Because of this, and because of the agents' need to continue exploring the bid space, it was thought unlikely the agents would be able to find and maintain a cooperative equilibrium. The dynamic reward function and the relatively large action space make it hard to apply the learning models used to find Nash equilibria in similar, static games [24].

Experimentation showed that, as with the previous constraint groups, the agents learnt to bid higher than SMP to gain the constraint payments. However, for level 3 constraints, the agents were unable to maintain a cooperative solution, despite the agents attempting to do so more than might be expected. Table III shows bids for the last 50 occurrences of environment 0100100000. In the last 50 times this environment was observed, each of the two agents made the maximum bid of 31, 9 and 7 times respectively, but these bids only coincided twice.

TABLE III

BIDS FOR LAST 50 OCCURRENCES OF ENVIRONMENT 0100100000 FOR THE TWO AGENTS IN CONSTRAINT GROUP 1

27 21	30 21	26 31	30 30	30 27
23 26	23 31	26 26	23 22	31 21
26 22	23 26	23 26	26 30	27 21
24 22	31 26	30 31	23 31	29 27
24 27	31 23	23 27	30 22	31 27
23 24	31 31	28 17	30 27	20 22
31 22	30 21	23 23	21 21	26 21
23 24	23 27	26 17	30 31	27 27
23 26	23 21	23 23	31 31	30 21
23 21	31 19	23 26	31 27	23 27

The fact that the agents are bidding 31 relatively frequently, despite it not being particularly profitable, is probably a result of the agents generalizing information from constraint level 4 environments (where 31 is the dominant equilibrium Pareto optimal action for both agents) to constraint level 3 environments. There are several possible reasons why the occasional mutual bids of 31 are not maintained. Firstly, the constrained on environments occur relatively infrequently, therefore the information gained about the benefits of cooperating may not be maintained by the agent for a sufficient length of time. Secondly, the fact that the agents maintain some probability of exploring any environment means the agent may change bid even if 31 has the highest estimated reward, and thirdly, the relatively large number of possible actions available to the agents makes it much harder to find and maintain a single bid pattern. It could also be the case that the incentive to cooperate is not large enough. Other environments provide greater rewards, and the agents concentrate their limited memory resources on the most profitable environments. Experimentation showed that, with 32 possible actions, cooperation rarely, if ever, emerged.

C3: 2 Player Games - restricted bid set

In order to give the agents a better chance of learning to cooperate in constrained environments, we ran further experiments where the two agents were limited to bids of 0-15, 25 or 31. In constraint level 3 environments, a bid of 0 to 15 is essentially incorrect, as on any normal range of actions by the other adaptive agents it will lead to a reward that is strictly dominated by the actions 25 and 31 (see Table II). A bid of 25 is classified as defection and a bid of 31 as cooperation.

In experimentation, the agents quickly learnt not to bid in the range 0-15 in all constraint level 3 environments. After exposure to 500 constrained on level 3 environments, over 90% of all bids by both agents were either 25 or 31. The speed of this convergence to a subset of bids illustrates the advantage of being able to group games together and use information from one game to help learn how to play another, since it is generally incorrect to bid 0-15 in both constraint level 2 and level 4 environments also.

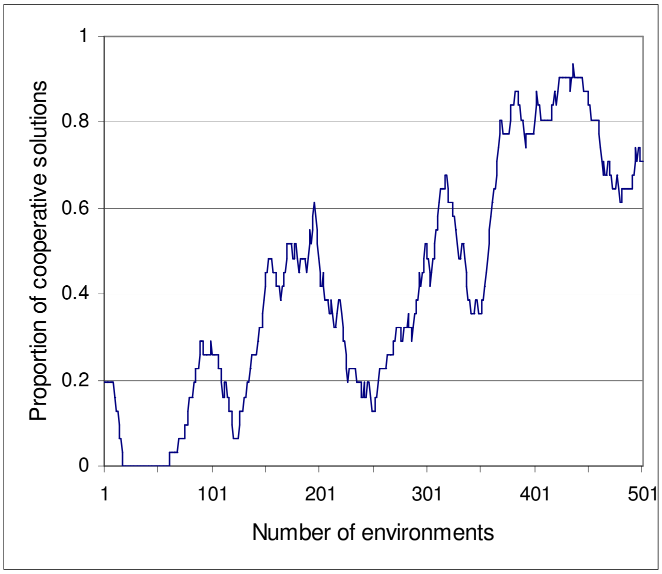

Fig. 10. Proportion of cooperative solutions over 30 instances for the first 500 occurrences of a particular constraint level 3 environment.

Figure 10 describes the progress towards cooperation for the first 500 instances of a particular game at constraint level 3. The data consists of the proportion of games over a 30 game period where both agents made the cooperative bid of 31. The competition in this game is characterized by a series of increases in cooperation followed by a rapid series of defections as one of the agents attempts to exploit the other's tendency to cooperate. Each new peak is at a higher level, which suggests the agents are learning to cooperate more after each setback. The sudden drop in cooperative acts is probably the result of the rule discovery mechanism introducing lower actions found to be successful in other environments. Many other constraint level 3 environments exhibited similar progress towards cooperation, although in some the agents converged to a strategy of mutual defection.

VI. Conclusions

In the Introduction, we identified our goals for this sequence of experimentation. We now summarize how well these goals have been met:

Can the agents learn to behave in ways observable in the real world?

In Section V-A, we compared the aggregate behaviour of the agents with that expected from the real world model. In order to make a convincing case for the long term potential of an evolutionary approach to economic modelling, it is important to be able to define a system where the agents behave in ways that are comprehensible in terms of the real world scenario. The model contains aspects of the actual market known to be important in generator bidding, and agents acting in the simulated environment have been shown to evolve strategies analogous to real world bidding strategies. With relatively limited resources, the agents learn to group similar environments together through the use of classifier system rules while maintaining a selective focusing on certain influential environments through the case lists. The hierarchical structure allows the balancing of objectives and by forming a default defensive rule set with CS_A the agents can further concentrate their memory on grouping and modelling the more profitable environments.

How does altering market structure alter agent behaviour and market conditions?

The possible future application of this type of model as an experimental advisory tool was illustrated in Section V-B by showing the effect of altering the market structure and observing behaviour. It was shown that the agents bid higher under a pay at bid price system as opposed to an SMP settlement system. However, the SMP system results in slightly higher total cost of generation. Obviously the application is far from being of real practical use. However, the agent architecture was designed in such a way as to allow for the potential scaling up of the environment size and for the inclusion of more realistic agent objectives. The model could be made more realistic in many ways. These include allowing a dynamic alteration of generation capacity by allowing agents to stop trading and new agents to enter the market, giving the agents alternative objectives such as maintaining market share, letting one agent control more than one generating unit and more accurately modelling the demand curve. All these features would allow the potential for more complex and subtle strategies to emerge.

Can the agents learn to cooperate?

In Section V-C, certain aspects of cooperation have been seen to emerge, although the large number of available actions, the exploration/exploitation policy and the potential incorrect generalization over environments make it difficult to maintain long term mutually beneficial strategies. It would be of interest to examine in more detail the potential for the evolution of cooperation and the effect of altering the distribution of game types on this potential cooperation. In terms of iterated games with unknown reward functions, the model introduces a realistic feature not often considered when looking at evolutionary agents interacting: namely, the fact that interactions in a game environment are rarely under identical circumstances, i.e. with identical reward functions. Agents may play games against each other where known factors, variables other than the other agents' bids, affect the reward matrix. The use of classifier rules and the hybridization with case based reasoning give the agents the potential to transfer knowledge from one game to a potentially similar, but less familiar, game.

The multi-agent model illustrates the potential benefits of an evolutionary approach to modelling economic situations where certain quantifiable factors independent of the players in the market affect the reward. The necessary coupling of scheduling and settlement and the requirement of meeting demand means electricity generation markets will always involve some degree of external control. Understanding how generator companies may behave under differing methods of external control and alternative market systems is of crucial interest. An autonomous adaptive agent approach offers an additional means of gaining this understanding and reinforcing or questioning opinions about behaviour derived from more traditional economic modelling approaches. The nature of the balancing mechanism is crucial to the efficiency of the market. Role playing simulations were commissioned to investigate, amongst other things, the effect on player's strategy of pay at bid price and SMP price setting mechanisms and the potential for the emergence of high priced equilibria [16]. These simulations consisted of 14 teams of participants consisting of people working in the electricity industry and students. This work re-emphasises the potential for an agent based approach to understanding these mechanisms identified in many papers such as [11], and demonstrates that a more complex learning mechanism can result in the evolution of realistic behaviour.

Although the market simulated in these experiments has now been replaced with the NETA, the environment variables are still relevant to electricity generation. The fact that the agents learn to bid independent of the trading rules (i.e. they are not attempting to learn loopholes in the system), means the agent architecture could simply be applied to a NETA simulation. Combining complex agent structure with a more accurate market simulation may provide greater insights into the effects of changes in market structure on the behaviour of competitors in the electricity industry.

Acknowledgment

The authors would like to thank the National Grid Company and the EPSRC for supporting this work.

References

[1] P. Anthony and N. R. Jennings. Developing a bidding agent for multiple heterogeneous auctions. *ACM Transactions on Internet Technology*, 3(3):185–217, 2003.

[2] R. Axelrod. The evolution of strategies in the iterated prisoner's dilemma. In L. D. Davis, editor, *Genetic Algorithms and Simulated Annealing*, chapter 3, pages 32–41. Morgan Kaufman, 1987.

[3] R. Axelrod. *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princetown University Press, 1997.

[4] A. J. Bagnall. A multi-adaptive agent model of generator bidding in the UK market in electricity. In D. Whitley, D. Goldberg, E. Cantú-Paz, L. Spector, I. Parmee, and Hans-Georg Beyer, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 605–612. Morgan Kaufmann: San Francisco, CA, 2000.

[5] A. J. Bagnall and G. D. Smith. An adaptive agent model for generator company bidding in the UK power pool. In *Lecture Notes in Computer Science 1829, Artificial Evolution*, pages 191–203. Springer-Verlag, 1999.

[6] A. J. Bagnall and G. D. Smith. Using an adaptive agent to bid in a simplified model of the UK market in electricity. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, page 774. Morgan Kaufmann: San Francisco, CA, 1999.

[7] A. J. Bagnall and G. D. Smith. Game playing with autonomous adaptive agents in a simplified economic model of the UK market in electricity generation. In *IEEE-PES / CSEE International Conference on Power System Technology POWERCON 2000*, pages 891–896, 2000.

[8] T. Băck, U. Hammel, and H-P. Schwefel. Evolutionary computation: Comments on the history and current state. *IEEE Transactions in Evolutionary Computation*, 1(1):3–17, 1997.

[9] S. Borenstein, J. Bushnell, and S. Stoft. The competitive effects of transmission capacity in a deregulated electricity industry. *Rand Journal of Economics*, 31(2):294–325, 2000.

[10] J. Bower and D. W. Bunn. Experimental analysis of the efficiency of uniform-price versus discriminatory auctions in the england and wales electricity market. *Journal of Economic Dynamics and Control*, 25(3):561–592, 2001.

[11] D. W. Bunn and F. S. Oliveira. Agent-based simulation - an application to the new electricity trading arrangements of England and Wales. *IEEE Transactions in Evolutionary Computation*, 5(5):492–503, 2001.

[12] M. Butz, T. Kovacs, P. L. Lanzi, and S. W. Wilson. Toward a theory of generalization and learning in XCS. *IEEE Transactions in Evolutionary Computation*, 8(1):28–46, 2004.

[13] Northwest Power Planning Council. Summary of national electricity restructuring legislation in the 104th and 105th congresses. http://www.nwppc.org/natleg.htm, 1997.

[14] P. J. Darwen and X. Yao. Co-evolution in iterated prisoner's dilemma with intermediate levels of cooperation: application to missile defense. *International Journal of Computational Intelligence and Applications*, 2(1):83–107, 2002.

[15] M. Dorigo and U. Schnepf. Genetics-based machine learning and behaviour based robotics: a new synthesis. *IEEE Transactions on Systems, Man and Cybernetics*, 23(1):141–154, 1993.

[16] London Economics. Role playing simulations of the new electricity trading arrangements. a report to the RETA programme. http://www.ofgas.gov.uk/elarch/index.htm, 1999.

[17] M. He, N. R. Jennings, and H.-F. Leung. On agent-mediated electronic commerce. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):985–1003, 2003.

[18] J. Hurst and L. Bull. A neural learning classifier system with self-adaptive constructivism for mobile robot control. *Artificial Life - In Press*, 2005.

[19] J. Nicolaisen, V. Petrov, and L. Tesfatsion. Market power and efficiency in a computational electricity market with discriminatory double-auction pricing. *IEEE Transactions in Evolutionary Computation*, 5(5):504–523, 2001.

[20] M. A. Nowak and K. Sigmund. Evolution of indirect reciprocity by image scoring. *Nature*, 393:573–577, 1998.

[21] R. G. Palmer, W. Brian Arthur, J. H. Holland, B. LeBaron, and P. Tayler. Artificial economic life: A simple model of a stockmarket. *Physica D*, 75:264–274, 1994.

[22] Office Of Electricity Regulation. New Electricity Trading Arrangements . http://www.ofgem.gov.uk/metering/industframe.htm.

[23] S. Schulenberg and P. Ross. An adaptive agent based economic model. In P. L. Lanzi, W. Stolzmann, and S. W. Wilson, editors, *Learning Classifier Systems: An Introduction to Contemporary Research*, volume 1813 of *LNAI*, pages 263–282, Berlin, 2000. Springer-Verlag.

[24] Y. S. Son and R. Baldick. Hybrid coevolutionary programming for nash equilibrium search in games with local optima. *IEEE Transactions in Evolutionary Computation*, 8(4):305–315, 2004.

[25] S. W. Wilson. Classifier fitness based on accuracy. *Evolutionary Computation*, 3(2):149–175, 1995.

[26] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

[27] X. Yao and P. J. Darwen. An experimental study of N-person iterated prisoner's dilemma games. *Informatica*, 18(4):435–450, 1994.

[28] X. Yao and P. J. Darwen. How important is your reputation in a multi-agent environment? In *Proc. of the 1999 IEEE Conference on Systems, Man, and Cybernetics*, volume 2, pages 575–580, Oct 1999.

[29] F. Zhuang and F. D. Galiana. Towards a more rigorous and practical unit commitment by lagrangian relaxation. *IEEE Transactions on Power Systems*, 3(2):763–773, 1988.