

Marketecture: A Simulation-Based Framework for Studying Experimental Deregulated Power Markets

Karla Atkins* Chris Barrett* Christopher M. Homan† Achla Marathe‡
Madhav Marathe* Shripad Thite§

Abstract

In this paper, we present MARKETECTURE, an agent-based, microeconomic, scalable model for studying deregulated power markets. Features that distinguish it from previously studied models include: the ability to generate individualistic, demographics based, elastic demand profiles; a highly configurable system that supports different matching algorithms for buyers and sellers, different market clearing mechanisms; ability to aggregate individuals to different classes; an electrical grid to physically clear the economic contracts etc. This paper describes the model and its various features in detail. A case study is done for the city of Portland, Oregon, to evaluate the performance and efficiency of the market under different market clearing algorithms and sellers' strategies. We analyze the structural properties of the market under different scenarios to validate our model. Our results show that if Vickrey auction clearing mechanism can induce the sellers to reveal their true production costs and bid at competitive level, the market performance can be almost pareto-efficient. The weighted average clearing method in the poolco market results in the lowest market clearing price (MCP). However, the market clearing quantity (MCQ) is also low which results in deadweight loss to the society. Our findings also show that the different orders of market execution (bilateral and poolco) can significantly affect the performance of the markets.

1 Introduction

California's recent, failed attempt to deregulate its electrical power market casts a shadow of doubt on the deregulation plans of other states, such as Nevada, Arkansas and New Mexico. In an effort to gain a better understanding of what went wrong in California, and how in the future one can build efficient, reliable markets, a number of researchers [2, 3, 8, 26, 30, 34, 33] have studied the deregulated aspects of the US electricity market. These studies have greatly enhanced our understanding of the issues involved in designing and administering deregulated electricity markets. However, most of the studies in the literature are restricted to static situations, markets with too few players, small networks etc. Additionally, many of the studies make assumptions such as perfect rationality, symmetric knowledge between players, global decision making, etc., in order for the studies to be feasible. The advent of computer-based modeling allows us to relax many of these constraints while maintaining the feasibility of our calculations. A number of experimental studies investigate various forms of trading that are or can be used by electrical markets [7, 8,

*CCS-5, P.O. Box 1663, MS M997, Los Alamos National Laboratory, Los Alamos NM 87545. email: karla,barrett,marathe@lanl.gov.

†Dept. of Computer Science, Rochester Institute of Technology. Work done while visiting Los Alamos National Laboratory. email: cmh@cs.rit.edu.

‡CCS-3, P.O. Box 1663, MS B265, Los Alamos National Laboratory, Los Alamos NM 87545. email: achla@lanl.gov.

§Dept. of Computer Science, Univ. of Illinois, Urbana Champaign. Work done while visiting Los Alamos National Laboratory. email: thite@uiuc.edu.

10, 13, 14, 18, 19, 22, 23, 35]. In particular, these studies investigate the design of, efficiency in, and market power of a restructured electricity market under different forms of market clearing and competition.

The experimental work in electricity market restructuring (economics, in general) can be partitioned into two broad classes. One is the human-based laboratory experiments, in which experiments are performed in a controlled environment and the cash-motivated human subjects are used as agents [23].¹ Human-based laboratories provide a natural environment for individual preferences and their reaction to institutional and regulatory changes to be studied. However, there are several limitations on the kinds of experiments that can be performed using human subjects. For example, inability to scale to a realistic number of players, set up complicated ‘rules of the game’, play out collusive and learning behavior, simulate the physical clearing of contracts etc.

The other class is the computer-based experiments which use agent based computational models [22, 34, 27, 5, 6]. These models can address a number of complex and interrelated issues that cannot be resolved using conventional economic models which are capable of providing solutions only in analytically tractable ways. These models can be scaled easily to realistic levels. In this paper, we present MARKETECTURE, an agent-based, individualistic, highly scalable, computational model which provides a general framework to study different markets, trading institutions and agents’ strategies. Our prototype focuses on the electricity market and has several unique features that distinguish it from previously-studied models. (i) Based on information from the United States Census and from a microscopic mobility simulator, the MARKETECTURE simulator generates for each individual (in the population considered) a time dependent, spatio-temporal, non-linear, elastic demand profile. (ii) Consumers and power generators are endowed with very realistic features, such as limited knowledge about other market entities, the ability to form into groups and place bids and asks as aggregate groups, bounded rationality, etc. (iii) Parameterized and configurable markets that allow different clearing mechanisms, matching algorithms, trading strategies. (iv) The electrical power grid, whose naturally limited capacity to deliver power from any generator to any consumer affects the clearing price of power, is part of the model. The details of each of these features are described in later sections of this paper.

The rest of this paper is organized as follows. In Section 2, we discuss the overall schematic design of our large-scale, microeconomic simulator. The basic design consists of four modules. The modules are the mobility based model for creating individualized power consumption patterns, the supply, the market, and the control module that coordinates the execution of a simulation. Section 3 discusses the algorithmic aspects of our simulator. Section 4 describes a case study for the city of Portland, Oregon. The study yields insights into the structural properties of the market. Section 5 describes some of the computational issues involved in implementing the simulator. Finally section 6 concludes the paper.

2 Description of the MARKETECTURE Simulator

Figure 1 depicts schematically the MARKETECTURE simulator. It is made up of four main modules: control, supply, demand, and market. The supply, demand, and market modules contain entities, e.g., buyers, sellers, markets, etc., whose actions and interactions drive the model. Each entity has a set of attributes that determine its individual character. We now describe in greater detail each of these components.

2.1 Demand Module

The MARKETECTURE simulator is designed to model the running of multiple market sessions over a long period of time. Since a person’s demand for power varies over time (e.g., people tend to consume less power when they sleep than when they are awake) it is important that this variation is modeled in as realistic a

¹For human based laboratory studies in other areas, see [15, 24, 28].

manner as possible. The demand module of the MARKETECTURE simulator is responsible for this modeling task. This module has three primary classes, the UPMOSTINTERFACE, which gathers demand data from a mobility simulator, called UPMoST [29]; the INDIVIDUALS who move around the simulator and demand power; and the LOCATIONS that INDIVIDUALS visit.

2.1.1 UPMOSTINTERFACE

The UPMOSTINTERFACE interacts with an urban population mobility simulator developed at Los Alamos National Laboratories called UPMoST [29]. The UPMOST simulator produces, for each INDIVIDUAL in the simulator, activity and mobility information. The UPMOST simulator, in turn, receives as input actual demographic information on the population being simulated as well as information on the physical infrastructure of the location being simulated, such as the geographic locations of buildings and roads. In short, UPMOST provides us with

- the power demand profiles of synthetic INDIVIDUALS² whose behavior is based on real demographic data, and
- the physical location of each synthetic INDIVIDUAL which allows us to determine the amount of power demand at each physical location over the course of a day.

We are thus able to compute the power demand at each LOCATION in the simulator at any given time as a function of the type of LOCATION, the number of INDIVIDUALS at that LOCATION, and the type of activities performed by those INDIVIDUALS at that LOCATION at that time. We use hourly intervals to create the demand profiles for each INDIVIDUAL and LOCATION. The UPMOSTINTERFACE has no attributes. There is only one instance of the UPMOSTINTERFACE.

2.1.2 LOCATION

A LOCATION represents the places where an INDIVIDUAL is at any point in time, such as: a house, office, shopping mall etc. In general, a LOCATION has no attributes. However, when the simulator is used to model an actual city like Chicago or Portland, we associate with each LOCATION its UTM [31, 9] coordinates.

2.1.3 INDIVIDUAL

An INDIVIDUAL i represents a person in the simulator. It has three attributes:

- $c_{activity}$, the activity that i is engaged in at a particular hour. The simulator recognizes a finite set of activities, such as: shop, visit, school, social recreation, work, home etc. Each activity has certain power requirements, represented by the attribute.
- c_{income} , the income of i .
- l , the current LOCATION of i .

The $c_{activity}$ and c_{income} attributes of each INDIVIDUAL are used to determine the amount of power demand in the system at any point in time. The attribute l determines the geographical distribution of the demand at any point in time.

²Synthetic individuals are people from a synthetic population. Our synthetic population is an imitation of the real population. The synthetic population preserves the key features of the real population and has the same statistical properties (such as correlation structure, joint distributions of the demographic variables, spatial distribution of households etc.) as the real population.

2.2 Market Module

The market module can simulate a variety of power markets, each having a distinct clearing mechanism. Currently, the MARKETECTURE simulator models two markets: a POOLCOMARKET and a BILATERAL-MARKET. In a bilateral market, a buyer and seller independently set up the physical and financial terms of the trade and are responsible for dispatching and receiving the physical deliveries of the commodity. In addition, the participants of the bilateral contract are also responsible for informing the independent system operator (ISO) of the size, source and sink locations of the contract. In a poolco market model, the buyers and sellers submit their bids and asks to the power exchange. The power exchange then dispatches ASK in economic order until all the bids have been satisfied.

There are two classes of trading agents, BUYERS, who on behalf of CONSUMERS go to the markets and buy power, and SELLERS, who on behalf of GENERATORS sell power. Each market has one attribute: a price cap p_{cap} , which is the maximum price per unit of power for which power may be traded on that market. There is only a single instance for each of the POOLCOMARKET and the BILATERALMARKET.

2.2.1 BUYER

A BUYER represents a collection of CONSUMERS (see below) and makes all economic decisions (with respect to the power markets) on its constituent CONSUMERS' behalf. For example, a BUYER could be the head of the household, owner of McDonald's, Wal-Mart etc. Each BUYER has one attribute: a list of the CONSUMERS $C = \{c_1, \dots, c_n\}$, that the buyer represents.

2.2.2 CONSUMER

CONSUMERS are not, in our model, people, nor are they agents that directly interact with the market. CONSUMERS are essentially the locations which are owned and represented by BUYERS. Each CONSUMER c has six attributes:

- L , a list of LOCATIONS that the Consumer represents.
- b , the BUYER who represents the CONSUMER in the POWERMARKET.
- u , the BUS from which c draws power.
- $f_{demand}(\cdot)$, a demand function of the form $f_{demand}(p) = \alpha + \beta/p$ that maps quantities to prices.
- α , the fixed amount of power that c always needs.
- q_{target} , the amount of power that c would like to buy.

The value of c 's demand function $c.f_{demand}(\cdot)$ at any point in time is based on the number of INDIVIDUALS at each LOCATION $l \in c.L$ and the quality of the location.

2.2.3 SELLER

A SELLER represents a collection of GENERATORS and makes all economic decisions (with respect to the power markets) on its constituent GENERATORS' behalf. Each SELLER s has the following attributes:

- $G = \{g_1, \dots, g_n\}$, a list of GENERATORS that the SELLER represents.
- $f_{ed}(\cdot)$, an invertible estimated market demand function of the form $f_{ed}(p) = a - bp$ that maps quantities to prices

- $s_{ask} \in \{oligopolist, competitor, competitiveOligopolist\}$, an asking strategy.

We assume that, for any SELLER $s' \neq s$, s does not know $s'.s_{ask}$, where s_{ask} is the asking strategy of s' .

2.2.4 ASK (respectively, BID)

An ASK (respectively, a BID) represents the amount and quantity of power that a SELLER (respectively, BUYER) wishes to sell (respectively, buy). Each ASK (respectively, BID) has three attributes:

- g (respectively, c), the GENERATOR (respectively, CONSUMER) on whose behalf it is placed.
- q , the quantity of power.
- p , the price per unit of power.

2.2.5 POOLCOMARKET

In each bidding round in a POOLCOMARKET, all SELLERS and BUYERS submit to the market their respective bids and asks. On the supply side, each SELLER s submits for each GENERATOR $g \in s.G$ exactly one ASK. On the demand side, each BUYER b submits a demand function which is a horizontal summation of all its constituent CONSUMERS' demand functions. It is the job of the POOLCOMARKET to satisfy the needs of the BUYERS in as cheaply and fairly a way as possible. This is done by dispatching ASKS in economic order, i.e., the cheapest ASK is dispatched first, followed by the next cheapest one, and so on, until all the BIDS are served. To ensure fairness, at the end of each bidding round the unit price of all the power dispatched is the same, regardless of the unit prices asked by each SELLER. We call this unit price the *market clearing price* (MCP). Three distinct *clearing policies* for setting the MCP are considered: *normal*, *Vickrey auction* and *weighted average* clearing. These clearing policies have either been implemented by ISOs like PJM and California in the past or considered by theoreticians and experimentalists [1, 20, 21, 25].

In a *normal* clearing, the MCP is the price of the *marginal* ASK, i.e., the ASK having the greatest unit price of all the ASKS dispatched by the Poolco market. When this clearing mechanism is used, SELLERS have an incentive to raise the unit price of their ASK beyond the competitive level. If the sellers are successful in inflating the price of the marginal ASK, every seller profits from it.³ However, this incentive is counteracted by the possibility that, in raising the unit price asked, a SELLER can be undersold by another SELLER. On the positive side, *normal* clearing induces producers to reduce their cost of production. The uniform clearing price allows higher profit margins to cheap producers.

The Vickrey auction policy was designed to induce truthful revelation of production costs and efficient dispatching [32]. In a Vickrey auction, the MCP is the price of the cheapest ASK not dispatched. Vickrey suggested that if the price received by the SELLER is independent of his ASK, i.e. if the sellers' ASK and his pay-off can be decoupled, all sellers would have an incentive to bid at their marginal cost. In such a case, a SELLER will be able to influence his payoff only to the extent that it affects the probability of his GENERATORS being called into operation. To maximize that probability, it would ask the marginal cost and yet be guaranteed to have a positive return if it is called into operation. See [13] for more on Poolco and Vickrey clearing mechanisms.

Under the weighted average clearing, MCP is determined by the weighted average price of all the ASKS dispatched by the Poolco market where the weights are simply the quantities offered by each of the selected SELLERS. If SELLERS are known to exercise market power, this clearing mechanism can keep the market clearing price in check.

³This behavior can also lead to economic inefficiency because the less efficient SELLERS may submit lower asks than the more efficient SELLERS resulting in the dispatch of less efficient ones.

2.2.6 BILATERALMARKET

In a BILATERALMARKET, BUYERS and SELLERS pair off and independently negotiate the physical and economic terms of the contracts. Each bilateral contract has to be submitted to the ISO for physical clearing. The ISO determines if the power grid has enough transmission capacity to support the physical terms of the contract. If (and only if) it does, the contract clears.

2.3 Demand Module

The demand module consists of the physical POWERGRID, the independent systems operator (ISO), and GENERATORS. The POWERGRID is made up of BUSES and LINES.

2.3.1 POWERGRID

The POWERGRID is the graph-like physical infrastructure that delivers power from one location to another. It has two parameters:

- $U = \{u_1, \dots, u_n\}$, a set of BUSES (defined below).
- $L = \{l_1, \dots, l_n\}$, a set of LINES (defined below).

There is only one instance of the POWERGRID.

2.3.2 BUS

A BUS is a point at which power may be injected or removed from the POWERGRID. A BUS may also route power to other BUSES along different LINES (see below). Each BUS b has two attributes:

- $L = \{l_1, \dots, l_n\}$, the LINES that are connected to b .
- q_{inj} , the quantity of power currently injected into b .
- θ , the phase angle.

2.3.3 LINE

A LINE transports power from one BUS to another. Each LINE l has three attributes:

- q_{max} , the maximum amount of power that at any moment l may carry.
- q_{flow} , the amount of power currently flowing through l .
- x , the reactance of the line.
- $U = \{u_1, u_2\}$, the pair of BUSES that l joins.

2.3.4 ISO

The ISO administers the POWERGRID. From the perspective of the simulator, ISO is the interface between the markets and the POWERGRID. When, during a market session, negotiations for buying and selling power reach a point where the market needs to validate that the POWERGRID can physically deliver the power being negotiated, the ISO is called in. The ISO then determines the feasibility of delivering the power being negotiated by simulating the POWERGRID with the negotiated power injected into the POWERGRID. The ISO has no attributes. There is only one instance of the ISO.

2.3.5 GENERATOR

GENERATORS produce electricity. Each GENERATOR g has seven attributes:

- s , the SELLER (defined below) that represents the GENERATOR in the markets.
- u , the BUS that connects g to the power grid.
- $f_{cost}(\cdot)$, an average total cost function of the form $f_{cost}(q) = aq + b$ that maps quantity to price.
- q_{ask} , the quantity of power that the g 's SELLER (defined below) plans to sell on the markets.
- m_{share} , the fraction of the total quantity of power sold on the markets that g sells.

2.4 Control Module

The MARKETECTURE simulator requires input in order to initialize each of the attributes of each entity in the simulation. As mentioned above, CONSUMERS are initialized by data from the UPMoST simulator. All other entities are initialized by input files. Thus, the simulator is highly configurable at run time. The control module has no entities, but rather controls the behavior of all the other entities in the simulator.

3 Algorithmic Description of the MARKETECTURE Simulator

We now describe how the simulator works. A key feature of our design is that the simulator is highly configurable at run time. It executes a sequence chosen from set of possible actions. The actual order in which the actions are executed is determined by the user, who, before the simulation takes place, writes a control script. Actions may be repeated multiple times. We now describe each of these actions in detail and provide their run-time complexity.

For most of the discussion in this section we adopt the following notation: for an entity e having attribute a , we will use the “.” operator to denote the value of a . For instance, given two CONSUMERS c_1 and c_2 , $c_2.u$ refers to c_2 's BUS and $c_1.u$ refers to c_1 's BUS.

A MARKETECTURE simulation begins in the control module, which initializes all the entities in the simulator and then actually runs the simulation.

3.1 Initialization

During initialization, the control module of the simulator opens user-defined files, which determine all the attributes of the the POWERGRID, the number of CONSUMERS, SUPPLIERS, GENERATORS, and SELLERS, and initializes all of the attributes of these classes, except for the $f_{demand}(\cdot)$ and q_{target} of the CONSUMERS, which are determined by the RESETCONSUMER algorithm (see below). The control module sets the global variable TIME to zero and calls the RESETCONSUMERS algorithm. Finally, it opens the control script and processes it.

For the POWERGRID and for each CONSUMER, BUYER, GENERATOR, and SELLER e , we call the value assigned during initialization to the each of the attributes a of e the *initial value* of $e.a$.

The running time of initialization is $\Theta(C + G + n^3)$, where C is the number of CONSUMERS, G is the number of GENERATORS, and n is the number of BUSSES in the simulation.

3.2 Control Script Processing

The heart of a MARKETECTURE simulation is the processing of the control script. The control script is simply a text file having one line for each action the writer of the script wants the simulator to do. Here is an example control script:

```
pricecap 100
bind random .25
runBilateralMarket
runPoolcoMarket Normal
reset
runPoolcomarket Vickrey
bind random .1
runBilateralMarket
runPoolcoMarket Wtdavg
bind random .2
runBilateralMarket
reset
...
```

As the above example shows, each line in the control script names one of four primary events: `reset`, `bind`, `runBilateralMarket`, and `runPoolCoMarket`. The events `runBilateralMarket` and `runPoolCoMarket` run a session of the market each event named after. The event `bind` determines which BUYERS interact with which SELLERS in the BILATERALMARKET. The event `RESET` increments the simulation to the next bidding period and resets the entity attributes that have been changed during market sessions. In the remainder of this section, we will describe each of these actions algorithmically. The following simple algorithm describes the process of reading and executing the control script.

Algorithm PROCESSCONTROLSRIPT

Foreach line l in the control script,

Call the algorithm corresponding to the event named in l .

The running of the control script does not have a fixed run-time complexity because its running time depends heavily on the content of a particular script.

3.2.1 Reset Event

As market sessions run, the demand lessens, supply diminishes, and the POWERGRID gets loaded with power. These dynamics are captured by the attributes of each of the entities in the simulator. After the power sold on the market is spent, a new period of time for which power must be bought and sold opens and the attributes of the entities need to be reset. This is the purpose of a reset event. Note that, as in the example script above, it is possible to run multiple sessions of the same market between reset calls. In this case, subsequent sessions run before the next reset event reflect the diminished demand and supply and increased load on the POWERGRID caused by previous sessions.

Algorithm RESET

Let $\text{TIME} = \text{TIME} + 1$;

Foreach BUS $u \in U$ in the POWERGRID,

Let $u.q_{inj} = 0$;

Call RESETCONSUMERS (defined below);

Call RESETGENERATORS (defined below).

The total running time of this algorithm is $\Theta(n + G + C + I)$, where n is the number of BUSES, G is the number of GENERATORS, C is the number of CONSUMERS, and I is the number of INDIVIDUALS. The RESETCONSUMERS algorithm is called both by the RESET algorithm and during the initialization of the simulator. Its main purpose is to determine the demand CONSUMERS have by contacting the UPMOST simulator. Note that the UPMOST simulator is an entirely different system from MARKETECTURE; it is not even necessary for it to run concurrently with a MARKETECTURE simulation, as long as the UPMOST simulator has previously output the data necessary for the MARKETECTURE simulator to run. Note also that, because the infrastructure for getting the demand data is encapsulated in the UPMOSTINTERFACE entity, the MARKETECTURE simulator could easily be adapted (by replacing the UPMOSTINTERFACE entity) to get its demand data from some other source.

Algorithm RESETCONSUMERS

Foreach INDIVIDUAL i ,

Retrieve from the UPMOSTINTERFACE the attributes of i for the current value of TIME.

Foreach CONSUMER c ,

Let I be the set of INDIVIDUALS associated with c ;

Let $\alpha = \sum_{i \in I} i.c_{activity}$, that is, α is the sum of the activity coefficients of each INDIVIDUAL in I ;⁴

Let $\beta = \gamma \sum_{i \in I} i.c_{income}$, that is, β is the sum of all the income of each INDIVIDUAL in I multiplied by a small constant *gamma*;

Let $c.f_{demand}(p) = \alpha + \beta/p$, where p is the price;

Let $c.q_{target} = f(\alpha, \beta)$

The running time of this algorithm is $\Theta(I + C)$, where I is the number of INDIVIDUALS and C is the number of CONSUMERS.

The RESETGENERATORS algorithm is called both by the RESET algorithm and during the initialization of the simulator. The purpose of RESETGENERATOR is to reset each GENERATOR g 's cost function $g.f_{cost}(\cdot)$ and to determine how much power SELLER $g.s$ will offer in the markets (i.e., the value of $g.q_{ask}$).

⁴The power demand at a location is the sum of the power demand that is independent of occupancy and the sum of the demand functions for all individuals at that location. Therefore,

$$\alpha = \left(\sum_k \text{type}(k) \cdot \theta(k) \right) + \left(\sum_{i \in I} \theta(c_{activity}(i)) \right)$$

$\text{type}(k)$ denotes the location that is used for activity type k , $\theta(k)$ is the quantity of power that is required at this location regardless of the number of individuals present. The second term shows the demand that is a function of the occupancy.

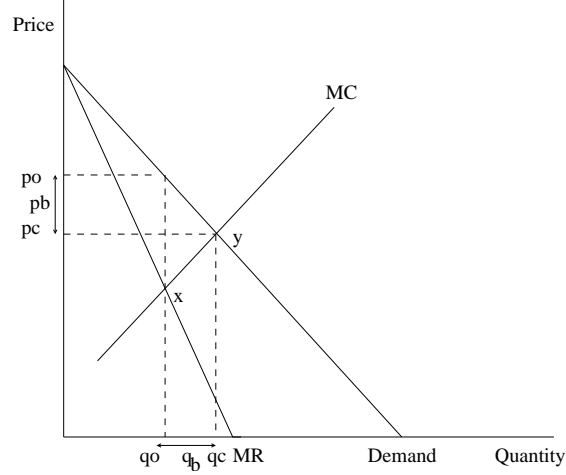


Figure 2: Display of oligopoly (x) and competitive equilibrium (y): MC represents the marginal cost, MR the marginal revenue, p_c and q_c the competitive price and quantity, p_o and q_o the oligopoly price and quantity respectively.

The asking behavior of each SELLER s is based on the strategy SELLER s wants to adopt. SELLER can select one of the three strategies i.e., *competitor*, *oligopolist* and *competitive-oligopolist*. Figure 2 shows how $g.q_{ask}$ is determined. If $s.s_{ask} = oligopolist$ then $g.q_{ask}$ is set to the point at which the marginal revenue and the marginal cost functions intersect, where marginal revenue $s.MR(\cdot)$ is defined as $d(q \cdot s.f_{ed}(q))/dq$.⁵ Note that, in our model, for some a and b $s.f_{ed}(p) = a - bp$, $s.MR(p) = a - 2bp$. If $s.s_{ask} = competitor$ then $g.q_{ask}$ is set to the point where the price equals the marginal cost. If $s.a = CompetitiveOligopolist$ then $g.q_{ask}$ is uniformly randomly picked to be between the above two extremes. s on behalf of each GENERATOR $g \in s.G$ offers a quantity, (p_b, q_b) , somewhere between the two extremes of what a competitive and an oligopolistic SELLER would offer. These extremes are represented, respectively, by $g.q_{max}$ (q_c) and $g.q_{min}$ (q_o). Thus $g.q_{ask} \in [g.q_{min}, g.q_{max}]$. In contrast to the traditional Cournot oligopolist model, we assume that s does not know the real market demand but is able to estimate it via s 's estimated demand function $s.f_{ed}(\cdot)$.⁶ Each SELLER also has an estimate of the market share that it expects to sell $s.m_{share}$.

Algorithm RESETGENERATORS

Let Q be the total quantity of power traded on the markets since the last time RESET was called;

Foreach GENERATOR g ,

Reset $g.f_{cost}(\cdot)$ to its initial value;

Let $g.m_{share} = (g.m_{share} + Q_g/Q)/2$, where Q_g is the quantity of power sold by g since the last time RESET was called;

Let $g.q_{max}$ be the unique quantity satisfying $g.s.f_{ed}(q) = d(g.f_{cost}(\cdot))/dq$;

Let $g.q_{min}$ be the unique quantity satisfying $d(q \cdot s.f_{ed}(q))/dq = d(g.f_{cost}(\cdot))/dq$;

If $g.s.s_{ask} = oligopolist$, that is, if g 's SELLER'S asking strategy is oligopolistic,

Let $g.q_{ask} = q_{min}$;

⁵Although we denote marginal revenue in a way usually reserved for attributes, MR is not an attribute of the SUPPLIER class because we can easily derive it from the estimated demand function attribute f_{ed} .

⁶Given that the $f_{demand}(\cdot)$ attribute of the CONSUMER is nonlinear, the linear estimates that $f_{ed}(\cdot)$ provides allow for information asymmetry between the SELLERS and the BUYERS.

Elseif $g.s.s_{ask} = competitor$, that is, if g 's SELLER'S asking strategy is competitive,
Let $g.q_{ask} = q_{max}$;
Else
Choose $g.q_{ask}$ by sampling uniformly from the range $[q_{min}, q_{max}]$.

The running time of this algorithm is $\Theta(G)$, where G is the number of GENERATORS.

3.2.2 RunPoolcoMarket Event

The POOLCOMARKET was described in detail in section 2.2.5. Here we describe the process of running the POOLCOMARKET. In order to make it easier to understand, we break the running of the POOLCOMARKET into four phases: dispatching power, calculating the market clearing price, calculating the bids (respectively, asks) of each CONSUMER (respectively, GENERATOR). The following algorithm shows each of these four phases, which we describe below in greater detail.

Algorithm RUNPOOLCOMARKET

DISPATCHPOWER;
CALCULATEMCP;
CALCULATEBIDSANDASKS;
CONFIRMCONTRACTS;

The total running time of this algorithm is $\Theta(C + G \log G + n^2)$, where C is the number of CONSUMERS, G is the number of GENERATORS, and n is the number of BUSSES.

In the first phase, we determine which GENERATORS are called in.

Algorithm DISPATCHPOWER

Let $A = \{\text{ASKS } a \mid (a.g \in G) \wedge (a.q = g.q_{ask}) \wedge (a.p = g.f_{cost}(a.q))\}$, where G is the set of all GENERATORS in the simulation, that is, A is the set of all ASKS based on some GENERATOR'S asking price p_{ask} ;

Let (a_1, a_2, \dots, a_n) be the ASKS in A listed in economic order, i.e., $(\forall i, j \in \{0, 1, \dots, n\})[(a_i, a_j) \in A) \wedge (i \leq j \Rightarrow a_i.p \leq a_j.p)]$;

Define $F_{demand}(p) = \sum_{c \in C} c.f_{demand}(p)$, where C is the set of all CONSUMERS in the simulator, that is, $F_{demand}(\cdot)$ is the sum of all demand functions;

Let $Q_{target} = \sum_{c \in C} c.q_{target}$, that is, Q_{target} is the sum of all target demands;

Let $i = \max\{i \mid (i \leq n) \wedge (\sum_{l=1}^i a_l.q \leq \min\{F_{demand}(a_i.p), Q_{target}\})\}$, where $a_l.q$ (respectively, $a_l.p$) is the quantity (respectively, price) of ASK a_l ;

Thus, the integer i that is calculated in the last line of the above algorithm is the index of the ASK corresponding to the most expensive GENERATOR that is called in. The sum of the demand functions of the BUYERS are used to determine the value of i .

The running time of this algorithm is $\Theta(C + G \log G)$, where C is the number of CONSUMERS and G is the number of GENERATORS. (The $G \log G$ term is from the required sorting on the ASKS of each GENERATOR that occurs in step 2 of the preceding algorithm.)

In the next phase, we determine the MCP, which depends on whether the clearing is Normal, Vickrey or Weighted Average.

Algorithm COMPUTEMCP

Let $Q_{supply} = \sum_{l=1}^i a_l \cdot q$

If $i \leq n$,

If we are running a Normal auction,

Let $MCP = a_i \cdot p$;

Else if we are running a Vickrey auction,

Let $MCP = a_{i+1} \cdot p$;

Else (i.e., if we are running a Weighted Average auction),

Let $MCP = \sum_{j=1}^i (a_j \cdot p)(a_j \cdot q) / (\sum_{j=1}^i a_j \cdot q)$;

Else

Let $MCP = \min\{p_{cap}, F_{demand}^{-1}(Q_{supply})\}$;

The running time of this algorithm is $\Theta(G)$, where G is the number of GENERATORS.

Next, we determine the BIDS and ASKS that are sent to the ISO for approval. Since ASKS were used in the first two phases, we simply assign the MCP to the price attribute of the ASKS that were called in. We may also need to modify the quantity field if there is a surplus.

Algorithm CALCULATEBIDSANDASKS

Let $Q_{demand} = \min\{F_{demand}(MCP), Q_{target}\}$;

Let $Q_{trade} = \min\{Q_{demand}, Q_{supply}\}$;

Foreach $l \in \{1, \dots, i\}$

Let $a_l \cdot p = MCP$;

Let $a_l \cdot q = a_l \cdot q \cdot Q_{trade} / Q_{supply}$;

Let $B = \emptyset$;

Foreach CONSUMER c ,

Let $B = B \cup \{(c, c.f_{demand}(\text{MCP}) \cdot Q_{trade}/Q_{demand, \text{MCP}})\};$

The running time of this algorithm is $\Theta(G + C)$, where G is the number of GENERATORS and C is the number of CONSUMERS.

Finally, we send these BIDS and ASKS to the ISO. If the ISO approves, we update the demand (respectively, supply) attributes of the CONSUMERS (respectively, GENERATORS).

Algorithm CONFIRMCONTRACTS

Submit ASKS a_1, \dots, a_i and each BID $b \in B$ to the ISO for approval.

If the ISO approves,

Foreach $l \in \{1, \dots, i\}$,

Let $a_l.g.q_{ask} = a_l.g.q_{ask} - a_l.q;$

Define $a_l.g.f_{cost}(q) = a_l.g.f_{cost}(q + a.l.q);$

Foreach BID $b \in B$,

Let $b.c.q_{target} = b.c.q_{target} - b.q;$

Define $b.c.f_{demand}(p) = b.c.f_{demand}(p) - b.q;$

The running time of this algorithm is $\Theta(G + C + n^2)$, where G is the number of GENERATORS, C is the number of CONSUMERS, and n is the number of BUSSES.

3.2.3 Bind Event

The bind event is where SELLERS and BUYERS are matched for an upcoming runBilateralMarket event. There are two syntactical forms that bind event can take. In the first case, the form is

bind random *num*

where *num* is a floating point number in $[0, 1]$. In this case the algorithm is

Algorithm BIND

Let $M = \emptyset.$

Foreach (BUYER, SELLER) $(b, s),$

Let r be a floating point number sampled uniformly from $[0, 1];$

If $r \leq num,$

Let $M = M \cup \{(b, s)\};$

The other form is

bind $(b_1, s_1), (b_2, s_2), \dots, (b_n, s_n)$

where $(b_1, s_1), (b_2, s_2), \dots, (b_n, s_n)$ is a list of (BUYER, SELLER) pairs. In this case the algorithm is

Algorithm BIND

Let $M = \{(b_1, s_1), (b_2, s_2), \dots, (b_n, s_n)\}$.

In either case, the running time of this algorithm is $\Theta(C \cdot G)$, where C is the number of CONSUMERS and G is the number of GENERATORS.

3.2.4 RunBilateralMarket Event

The BILATERALMARKET was described in detail in section 2.2.6. Before a runBilateralMarket event occurs, we assume that a bind event has occurred. The RUNBILATERALMARKET algorithm takes one argument, the set of matchings M created by the most recent bind event.

Algorithm RUNBILATERALMARKET

Foreach $(b, s) \in M$, where M is the latest set of matchings created by the BIND algorithm,

Foreach $(c, g) \in b.C \times s.G$,

Let Q be the unique quantity satisfying $(Q = c.f_{demand}(g.f_{cost}(Q)) \wedge (f_{cost}(Q) \leq p_{cap}) \wedge (Q \leq \min\{c.q_{target}, g.q_{ask}\}) \wedge (\min\{Q, g.f_{cost}(Q)\} > 0))$;

If such a Q exists,

Submit BID $(c, Q, g.f_{cost}(Q))$ and ASK $(g, Q, g.f_{cost}(Q))$ to the ISO for approval;

If the ISO approves,

Let $g.q_{ask} = g.q_{ask} - Q$;

Define $g.f_{cost}(q) = g.f_{cost}(q + Q)$;

Let $c.q_{target} = c.q_{target} - Q$;

Define $c.f_{demand}(p) = c.f_{demand} - Q$;

Note that the order in which the (BUYER, SELLER) pairs are chosen to negotiate contracts is significant. Because the cost and demand functions are updated as soon as a contract is cleared, as a rule of thumb, contracts that clear earlier will be for more power. The running time of this algorithm is $\Theta(C \cdot G \cdot n^2)$, where C is the number of CONSUMERS, G is the number of GENERATORS, and n is the number of BUSES.

3.3 Computing Power flow

The power flow problem is to compute the load on each LINE of a POWERGRID given the power injected into the grid. We ask the reader to refer to a standard text, e.g., [36], for an introduction to the topic.

Let G be the POWERGRID. Recall that G has attributes U , i.e., its set of n BUSES, and L , its set of m LINES. One of the BUSES $b_1 \in U$ is designated the *reference*, or *swing*, BUS. Let $l_{i,j} \in L$ denote the LINE between BUSES i and j .

In some power flow models, each LINE $l_{i,j}$ has an impedance $z_{i,j} = r_{i,j} + \hat{j}x_{i,j}$ where $r_{i,j}$ is the resistance and $x_{i,j}$ is the reactance of the LINE. The input is a power injection vector $P \in \mathbb{C}^n$ specifying the power injected at each BUS in the POWERGRID; loads are simply negative injections. The output is a vector $L \in \mathbb{C}^m$ specifying the amount of current flowing through each LINE.

Our simulator considers only the “linearized,” or the “DC,” load flow problem since the linearized version scales up easily. The power flow through $l_{i,j}$ from BUS i to BUS j is

$$P_{i,j} = \frac{\theta_i - \theta_j}{x_{i,j}}$$

where θ_i and θ_j are the phase angles at BUSES i and j [11].

Now, for each BUS i , the power P_i injected at the BUS must equal the power flowing out of the BUS. Therefore,

$$P_i = \sum_j \frac{\theta_i - \theta_j}{x_{i,j}}.$$

This gives us a system of linear equations $P = BT$, where $P = [P_1, \dots, P_n]$ is the vector of power flow at each BUS, $T = [\theta_1, \dots, \theta_n]$ is the vector of phase angles at each BUS, and B is the matrix LINE susceptances given by

$$\begin{aligned} B_{i,j} &= -\frac{1}{x_{i,j}} & i, j \neq 0 \\ B_{i,i} &= \sum_{j \neq i} \frac{1}{x_{i,j}} & i \neq 0 \\ B_{i,1} &= 0 & i \neq 0 \\ B_{1,i} &= 0 & i \neq 0 \\ B_{1,1} &= 1 \end{aligned}$$

The solution to the system of equations is obtained by computing the inverse B^{-1} of B and substituting in $T = B^{-1}P$ to solve for T . Knowing θ_i and θ_j and given $x_{i,j}$, we can compute the power flow through LINE $l_{i,j}$. Each LINE has a maximum power flow capacity. If the actual power flow through the LINE as computed above exceeds this capacity, then we say that the physical constraints of the POWERGRID are violated.

Inverting B occurs once during the simulation, during the initialization phase. It is $\Theta(n^3)$, where n is the number of BUSES.

The basic algorithm for computing power flow is run by the ISO when contracts are submitted to it for approval. It updates the loads on the POWERGRID BUSES with the values specified by the BIDS and ASKS (subtracting power for a BID, adding power for an ASK). It then solves the power flow equations, where the power vector P is the vector of the q_{inj} attributes of the BUSES. Note that, even though BIDS and ASKS do not explicitly have a BUS attribute, the corresponding BUS can be found from the BUS attribute u of the BID'S CONSUMER c or ASK'S GENERATOR g . If any of the lines exceed their capacity, the ISO removes the power just loaded onto the POWERGRID and disapproves the contracts. Otherwise, it approves them. This algorithm takes two arguments: a list of the ASKS A and a list of BIDS B that need to be approved.

Algorithm APPROVECONTRACTS

Foreach ASK $a \in A$,

Let $a.g.u.q_{inj} = a.g.u.q_{inj} + a.q$;

Foreach BID $b \in B$,

Let $b.c.u.q_{inj} = b.c.u.q_{inj} - b.q$;

Solve the powerflow equations;

Foreach LINE $l \in L$, where L is the set of LINES in the POWERGRID,

If $l.q_{flow} > l.q_{max}$
 Foreach ASK $a \in A$,
 Let $a.g.u.q_{inj} = a.g.u.q_{inj} - a.q$;
 Foreach BID $b \in B$,
 Let $b.c.u = b.c.u.q_{inj} + b.q$;
Reject contracts;
Exit algorithm;

Approve contracts (note: if the algorithm gets this far, i.e., if it does not exit, then no line constraints have been violated);

This algorithm is $\Theta(A + B + n^2)$, where A (respectively, B) is the number of ASKS (respectively, BIDS) and n is the number of BUSSES. $\Theta(n^2)$ is the amount of time it takes to solve the powerflow equations, assuming that B^{-1} has already been computed.

The preceding discussion of the modules, entities and algorithms used in *Marketecture* aims to show the level of scale and detail that it can capture. This is a large-scale system, capable of handling transactions between thousands of buyers and SELLER who further represent millions of consumers and hundreds of generators, respectively. Previously designed large-scale systems either do not represent individual behavior or model individuals in such a way that their behavior is unrealistic [12]. Market simulators that model realistic human behavior [24] have modeled it at such a high level of detail that they cannot easily scale up to the size of our simulation. Our approach is a compromise between these two kinds of approaches. We model individual behavior by providing market entities with parameters. These parameters allow each entity an individual character but these are not so detailed that the simulator does not scale. The goal is to capture different behavior of the system using different sets of parameters. A parameterized representation allows efficient use of computational resources. Such representations are also valuable from the standpoint of extensibility. The design and architect of our system is set up in such a manner that other commodity markets can be studied with minimal modifications.

4 A Case Study

We now use *Marketecture* to perform a study that evaluates the performance and efficiency of the POOL-COMARKET and BILATERALMARKET markets under different trading institutions and orders of market execution. This study analyzes different strategies of the SELLER and a range of different trading arrangements to determine which combinations lead to higher market efficiency and social welfare through their impact on market clearing price, market clearing quantity, buyers' surplus, sellers' surplus and dead weight loss to the society.

There have been several experimental studies done in the literature which address similar issues. [13] uses a game theoretic approach to show that if the pool settlement price is modified to use Vickrey auction clearing, it would remove the ability of the SELLER to influence the clearing price and make the marginal cost bidding the dominant bidding strategy. Work by [22] uses an agent based modeling approach to understand the issues of market power and efficiency in a wholesale electricity market. [10] compares results of a *sealed bid offer* market mechanism and a uniform price double auction mechanism in a spot electricity market. The comparison of performance is done in terms of market efficiency, buyer and seller profitability and the delivery price. Their results show that the sealed bid offer mechanism performs better than the uniform price double auction mechanism. [23] uses cash motivated human laboratory to compare the two

alternative institutional arrangements for the trading of electric power. Day-ahead sealed bid trading and the continuous double auction trading is used to measure the efficiency, distribution of surplus between BUYER and SELLER and the effect of location on prices and profitability. [4] uses an agent based simulation model of the wholesale market for electricity in England and Wales to study the impact of uniform versus discriminatory clearing prices which were implemented as SMP (System Marginal Pricing) and pay-as-bid clearing rules respectively.

Our study uses multiple market clearing mechanisms and multiple seller's strategies to determine the performance of the different markets. Several different combinations of markets, strategies and clearing mechanisms are analyzed with a high level of detail, accuracy and realism.

4.1 Market Performance and Efficiency

The parameters used in this study to measure the market performance and efficiency are (i) buyers' surplus, (ii) sellers' surplus, (iii) pareto efficiency/social welfare/deadweight loss, (iv) market clearing price and (v) market clearing quantity. Similar measures have also been considered by other researchers in the electricity literature [10], [23], [4].

Buyers' surplus is measured as the difference between the amount of money that a BUYER actually pays for quantity x and the amount he would be willing to pay for quantity x rather than do without it. Sellers' surplus is calculated as the difference between the SELLER's total revenue and its variable costs. Pareto efficiency is measured by the percentage of dead weight loss to the society or the proportion of maximum surplus captured by the BUYER and SELLER. If theoretically, the maximum surplus available to the society is 100\$ and sum of SELLER and buyers' surplus is only 95\$, then the dead weight loss to the society is 5% and the pareto efficiency is 95%. We use the total surplus available under the *competitive* strategy and *normal* clearing as the theoretically maximum surplus possible under the POOLCOMARKET. All other strategies are compared with the *competitive* strategy to determine the dead weight loss. In the BILATERALMARKET also, the *competitive* strategy is considered to be pareto efficient. This definition of efficiency is consistent with the measure used in [23]. The market clearing algorithms used for the POOLCOMARKET clearing are *normal*, *Vickrey auction* and *weighted average* clearing. For the BILATERALMARKET, this study uses the random matching probability of 1 to pair the BUYER and SELLER which means that all BUYER and SELLER have access to each other and can potentially pair with each other given their demand and supply functions. In both BILATERALMARKET and POOLCOMARKET the SELLER can choose between the previously mentioned three business strategies; *competitor* (C), *oligopolist* (O) and *competitive-oligopolist* (B).

4.2 Input Parameters

This study uses the synthetic population data on 1.6 million individuals for the city of Portland, Oregon. The previously mentioned UPMoST (Urban Population Mobility Simulation Technology) tool created a regional population imitation of Portland with demographics closely matching the real population [29]. The spatio-temporal demand profiles for every hour, total of 24 hours, were constructed for each of the 1.6 million individuals using the technique given in Section 2.1.

The 1.6 million individuals in Portland are distributed over 243,000 spatial locations. These locations are represented by the entity *Consumer* in our simulation. Each of the 243,000 locations are further assigned to 400 buyers on a random basis. On the supply side, 40 generators were assigned to 40 different sellers. Based on the cost function, an estimated market share and an estimated linear market demand function, each SELLER calculates its profit maximizing price and output level for the three strategies explained above.

Each simulation run is performed for 24 hours. In each hour, the market is run 4 times to give multiple chances to the BUYER and SELLER to meet their targets for the hour. The BUYER aim is to meet the sum of

the target demand for all its CONSUMER and the sellers' aim is to sell the profit maximizing output calculated for the hour. A price cap is set at 100\$ per MW for both the POOLCOMARKET and BILATERALMARKET.

4.3 Summary of Results

Table 1 shows the market clearing price (MCP), market clearing quantity (MCQ), buyers' surplus, sellers' surplus and the dead weight loss (DWL) to the society for the POOLCOMARKET and BILATERALMARKET under each of three strategies of the SELLER. Each of the numbers shown in the table are hourly averages calculated over a 24 hour period. The buyer's surplus is calculated by summing up all its corresponding consumers' surplus. Note that the surplus for the inelastic part of the demand curve is bounded by the price cap. For SELLER, the profits plus the fixed cost determine the surplus. The fixed cost has been added to the surplus to reflect that the SELLER consider the fixed costs to be sunk cost which will be incurred no matter whether the SELLER actually sells anything or not. A SELLER who does not get called in the market to serve the load will have zero surplus by this definition. The dead weight loss is that part of the surplus that neither goes to the BUYER nor to the SELLER.

• Poolco Model

1. The results for the POOLCOMARKET are summarized in Table 1. Panel 1 shows that the system is pareto-efficient under *normal* clearing and *competitive* strategy. The surplus generated here is used as a benchmark to calculate the DWL in the other POOLCOMARKET clearing mechanisms and strategies. The DWL to the society under competition is zero and quantity cleared is the highest.
2. If *Vickrey* auction clearing mechanism really induces the SELLER to reveal their true production costs and bid at competitive level, the market performance is almost pareto-efficient. This is shown in the first row of Panel 2 where *Vickrey* clearing and competitive strategy is followed. The MCP is only marginally higher and the dead weight loss is almost zero. Higher MCP results in slightly higher sellers' surplus and lower buyers' surplus.
3. Even if *Vickrey* clearing does not lead to revelation of true costs but induces the SELLER to move from the *oligopolistic* strategy to *competitive-oligopolist* strategy, it reduces the dead weight loss to the society by more than 11% and increases the buyers' surplus by 13%.
4. If the POOLCOMARKET clearing algorithm is *Vickrey* and yet the SELLER continue to behave like oligopolists, the market performance is worse than it is under *normal* and *weighted average* clearing.
5. If *weighted average* clearing method is used to clear the POOLCOMARKET, the MCP is the lowest resulting in the highest buyers' surplus and lowest sellers' surplus. The dead weight loss and the MCQ are only marginally different as compared to other POOLCOMARKET clearing mechanisms. This holds for all three different strategies.
6. Among the four different POOLCOMARKET clearing algorithms, the worst outcome is realized when *Vickrey* auction with 'O' strategy is used. The MCP is the highest, the MCQ is the lowest, the dead weight loss is the highest and the consumer surplus is the lowest.

• Bilateral Model

1. In the BILATERALMARKET, the average total cost (ATC) curve is offered as the supply curve no matter what the strategy of the SELLER is. Different business strategies are used to only create different upper bounds on the quantity offered. If the strategy is competitive, i.e. $g.s_{ask} = competitive$, the SELLER offers the ATC curve bounded by $g.q_{ask} = q_{max}$, if the strategy is

- oligopolistic i.e. $g.s.s_{ask} = oligopolist$, the ATC curve is bounded by $g.q_{ask} = q_{min}$, and if the strategy is competitive-oligopolist, it is bounded by $g.q_{ask}$ where $g.q_{ask}$ is sampled uniformly from the range $[q_{min}, q_{max}]$.
2. The average contract price per unit in the BILATERALMARKET is much less than the average price in the POOLCOMARKET. The reason being that in the POOLCOMARKET, every SELLER gets paid the same price as bid by the marginal SELLER. This results in a more inefficient uniform pricing.
 3. The absolute value of the buyers' and sellers' surplus in the BILATERALMARKET is much lower than in the POOLCOMARKET. This is due to the fact that individual contracts are being negotiated in the BILATERALMARKET resulting in more efficient pricing of contracts.
 4. The relative value of the surplus is much higher for the BUYER than for the SELLER in the BILATERALMARKET as compared to the POOLCOMARKET.⁷ This is because the SELLER are bidding at the ATC in the BILATERALMARKET. Also, due to the uniform pricing rule in the POOLCOMARKET SELLER get paid much higher than their bid price resulting in a higher surplus for the SELLER.
 5. Note that the average clearing price under *competitive* strategy is higher than the *oligopolist* strategy. This is because when more quantity is sold, the price cap is triggered more often resulting in a higher average price. To confirm that this indeed is the reason, we removed the price cap and found that the average price under 'C' strategy is lower than the average price under 'O' strategy.

Table 1

Market Performance and Efficiency Results

Strategy	MCP	MCQ	Buyer Surplus(%)	Seller Surplus(%)	Deadweight Loss(%)
Panel 1: Poolco Model - Normal Clearing					
C	12.53	1800	78.03	21.97	0
B	22.83	998	55.61	31.67	12.71
O	32.01	715	42.96	33.12	23.90
Panel 2: Poolco Model - Vickrey Auction Clearing					
C	12.76	1791	77.33	22.57	0.09
B	22.59	1011	55.92	31.69	12.38
O	32.15	714	42.79	33.25	23.94
Panel 3: Poolco Model - Weighted Average Clearing					
C	5.94	1769	92.70	5.12	2.17
B	13.75	1034	70.50	18.32	11.17
O	28.48	715	47.09	28.99	23.91
Panel 5: Bilateral Model					
C	7.18	1806	95.17	4.83	0
B	7.22	1354	71.49	4.23	24.27
O	6.94	903	47.36	3.20	49.42

Table 2 shows the market performance when both BILATERALMARKET and POOLCOMARKET are allowed to run in a certain order within the hour. Each market player is given a chance to play in the

⁷We use the total surplus generated in the BILATERALMARKET with C strategy to calculate the DWL for the B and O strategies in the BILATERALMARKET.

BILATERALMARKET and POOLCOMARKET to fulfill the target demand and sell the desired output. Two iterations of BILATERALMARKET and two iterations of POOLCOMARKET are run in each hour. Different scenarios are run using different orders of market execution to analyze the effect of ordering on the performance of the market. In the first scenario, two iterations of the BILATERALMARKET are run before the two POOLCOMARKET iterations. The results from this scenario are shown in the first three panels of Table 2. The only difference in these panels is that different POOLCOMARKET clearing algorithms are used in the POOLCOMARKET. *Normal*, *Vickrey* and *weighted average* clearing results are shown in panels 1, 2 and 3 respectively.

In the second scenario the order of execution for the BILATERALMARKET and POOLCOMARKET is reversed. Now the two POOLCOMARKET iterations are run before the two BILATERALMARKET iterations. Results from the second scenario are shown in panels 4, 5 and 6 of Table 2. Again, *normal*, *Vickrey* and *weighted average* clearing is used in the POOLCOMARKET and the results are displayed in panels 4, 5 and 6 respectively. The DWL for each of the panels in Table 2 has been calculated by using a benchmark total surplus generated when both BILATERALMARKET and *normal* POOLCOMARKET were run under ‘C’ strategy. For example, Panels 1-3 use the benchmark total surplus generated when BILATERALMARKET followed by *normal* POOLCOMARKET is run with a ‘C’ strategy. Panels 4-6 use the benchmark total surplus generated when *normal* POOLCOMARKET followed by BILATERALMARKET is run with a ‘C’ strategy.

Order of Market Execution

- If BILATERALMARKET is run first, more volume get traded at a cheaper price since prices on average are lower in the BILATERALMARKET market than POOLCOMARKET.
- The buyers’ surplus is much higher when BILATERALMARKET is run before POOLCOMARKET. This is because when BILATERALMARKET is run first, all the inelastic demand is served in this market. Given that the SELLER are only charging average total cost and there is no uniform pricing rule like the POOLCOMARKET, the sellers’ surplus is much lower in the BILATERALMARKET.
- The total volume traded i.e. $(MCQ_{bi} + MCQ_{po})$ is higher when POOLCOMARKET is run before the BILATERALMARKET. The reason being that all the inelastic demand has to be met in the market that runs first even if it is expensive. Once the inelastic demand is served in the POOLCOMARKET, the BILATERALMARKET is run which is usually much cheaper than the POOLCOMARKET. The downward sloping demand functions allow higher quantities to be bought in the BILATERALMARKET. The flip side of this behavior is shown in the first three panels where POOLCOMARKET is run after BILATERALMARKET. The volume traded in the BILATERALMARKET is high but in the POOLCOMARKET it is really low. This is because all the inelastic demand is met in the BILATERALMARKET which had run first. The rest of the demand is price elastic. This, when offered in the relatively expensive POOLCOMARKET, results in low volume traded in the POOLCOMARKET.
- The DWL to the society increases consistently as SELLER move from being *competitive* to *competitive-oligopolist* to just *oligopolists* no matter what the order of execution is. When the BILATERALMARKET is run first, the DWL is higher than the DWL when POOLCOMARKET is run first. This is true under all three strategies. This is because when BILATERALMARKET is run first total volume traded is much lower.
- Panel 6 shows that when *weighted average* POOLCOMARKET is run before the BILATERALMARKET with a ‘C’ strategy, the DWL is -4.19. This simply means that the benchmark total surplus that was chosen with the expectation that *normal* POOLCOMARKET will always lead to highest quantity traded and hence most efficiency is not true. The DWL is the loss generated from both the markets. When *weighted average* POOLCOMARKET is run first, the volume traded in that market was the lowest in

comparison with the other POOLCOMARKET clearing mechanisms under ‘C’ strategy. This allowed more quantity to be traded under BILATERALMARKET (after the *weighted average* POOLCOMARKET) resulting in surplus higher than what was generated in Panel 5 under *normal* POOLCOMARKET with a ‘C’ strategy.

Table 2

Results from Different Order of Market Execution

Strategy	MCP_{bi}	MCQ_{bi}	MCP_{po}	MCQ_{po}	B.Surplus(%)	S.Surplus (%)	DWL(%)
Panel 1: Bilateral Followed by Normal Poolco							
C	7.17	1799	12.91	70	83.60	16.40	0
B	6.97	1359	16.48	67	71.07	4.53	24.39
O	6.97	901	29.23	18	46.83	3.41	49.75
Panel 2: Bilateral Followed by Vickrey Poolco							
C	7.18	1806	13.77	0	92.30	4.69	3.00
B	7.28	1353	19.75	0	73.38	3.77	22.84
O	6.96	902	34.20	4	46.16	3.17	50.66
Panel 3: Bilateral Followed by Weighted Average Poolco							
C	7.16	1799	1	68	94.18	4.68	1.13
B	6.92	1363	8.54	21	70.30	3.86	25.83
O	6.97	901	26.04	18	46.86	3.38	49.75
Strategy	MCP_{bi}	MCQ_{bi}	MCP_{po}	MCQ_{po}	B.Surplus	S.Surplus	DWL
Panel 5: Normal Poolco Followed by Bilateral							
C	5.42	1776	12.41	1819	76.97	23.02	0
B	5.96	1365	22.85	997	55.85	31.81	12.33
O	7.70	870	34.39	667	40.78	35.37	23.84
Panel 6: Vickrey Poolco Followed by Bilateral							
C	5.43	1778	12.68	1803	76.22	23.55	0.22
B	6.25	1361	23.41	977	54.96	32.32	12.70
O	7.71	870	34.49	666	40.78	35.37	23.87
Panel 7: Weighted Average Poolco Followed by Bilateral							
C	4.28	1880	3.82	1561	98.24	5.95	-4.19
B	7.00	1301	14.49	935	68.23	17.02	14.74
O	7.70	870	30.19	666	45.17	30.97	23.85

5 Computational Issues

In this section, we describe our experiences with the computational issues involved in designing and implementing our simulation tool in order to perform large-scale simulations. The running time of each of the component algorithms was discussed in Section 3.

5.1 Computing power demand

We compute the power demand at any given time of day as a function of the activities and demographics of approximately 1.6 million INDIVIDUALS spread over roughly a quarter million distinct locations. This input data is obtained from the micro-simulation performed by the UPMOSTINTERFACE for INDIVIDUALS in a medium-sized urban area.

For reasons of modularity of the code, we currently perform this step as a pre-processing phase. The input to the pre-processing phase is a set of the activities and demographic information generated by the UPMOSTINTERFACE. The output is a set of 24 files giving the aggregated demand profile (essentially, the coefficients of a demand function) for each hour of the day at all locations. In order to discover the demand profile at a given location during a given hour, one can simply load the demand profiles from the appropriate hour.

It is possible to perform this step as a separate pre-processing phase because we have an *a priori* fixed time step (1 hour) which defines the smallest granularity of simulated time during the course of the simulation. However, we would like to implement the more elegant solution of querying a database server for the demand profile at a given location during an arbitrary window of time. In a parallel computing environment, this database could even be distributed by partitioning it according to the topology of the grid. The demand at a given location changes over time only as a result of a change in the activities performed at the location or the occupancy of the location. If such changes are small, a very efficient caching strategy can be developed to reduce the database lookup time.

5.2 Demand Functions

The aggregation step described in step 3 of algorithm DISPATCHPOWER computes the sum $F_{demand}(\cdot)$ of individual demand functions. The implication is that at any price per unit power p , the value of $F(p)$ denotes the aggregate demand at p . However, with the introduction of discontinuous constraints, such as the maximum target demand attribute q_{target} for each CONSUMER, this is no longer true. Consequently, we have to carefully consider the semantics of the disaggregation step in algorithm CALCULATEBIDSANDASKS when the net power cleared in the market is partitioned among individual CONSUMERS.

Consider a small example with two demand functions $f_1(p) = 20 + 400/p$ and $f_2(p) = 30 + 300/p$ and target demands $t_1 = 50$ and $t_2 = 70$. Then, the aggregate target demand is $T = t_1 + t_2 = 120$ and the aggregate demand function is $F(p) = 50 + 700/p$. At $p = 10$, we compute $F(10) = 120$ which does not exceed the aggregate target demand T . Hence, we conclude that the aggregate demand in the market is at least 120 units of power. However, we find that $\min\{f_1(10), t_1\} = \min\{60, 50\} = 50$ and $\min\{f_2(10), t_2\} = \min\{60, 70\} = 60$ so that the actual aggregate demand is at most $50 + 60 = 110$. This example illustrates the case precisely when, for some p ,

$$\sum_i \min\{f_i(p), t_i\} < \min \left\{ \sum_i f_i(p), \sum_i t_i \right\}$$

even though $\sum f_i(p) \leq \sum t_i$. We then have a discrepancy between the actual aggregate demand and the aggregate demand computed by evaluating the aggregate demand function.

Another issue is that the function $F(p)$ is the sum of demands of *all* BUYER at price p . However, for sufficiently large p , some CONSUMERS' demand functions may evaluate to a negative value. Whenever a CONSUMER'S demand function is negative, it means that the CONSUMER is not willing to buy any power at that price. Therefore, the actual aggregate demand at price p is really the sum of the demands of only the CONSUMERS whose demand at price p is non negative.

The problems arise from the fact that the demand functions are only piecewise continuous, but we represent the demand function as a sum of continuous functions. In order to solve the problem in its full generality, we would require an implementation of functions that are piecewise continuous and an additional (horizontal sum) operation for a set of such functions. The actual aggregate demand at a given price p would then be computed by summing the actual demand of each CONSUMER (the minimum of the value of the demand function and the target demand of the CONSUMER) and summing over the CONSUMERS. This can be prohibitively expensive if the number of CONSUMERS is large.

5.3 Power flow

Computing the load on each LINE in the power grid using the “linearized” DC method involves solving a system of linear equations. The coefficients in this system are the susceptances of the power LINES, which do not change during the course of the simulation. Hence, the inverse, B^{-1} , of the matrix B of coefficients is computed only once and cached in a separate file. Further runs of the simulation that use the same power grid simply load the inverse matrix, B^{-1} , from this file without having to recompute it. This method can provide substantial savings in computation time for large grids. We thus avoid the $\Theta(n^3)$ matrix inversion operation via either LU decomposition or Gaussian elimination at every run of the simulator. We still incur the $\Theta(n^2)$ running time of back substitution for computing the actual load on each LINE whenever the power injected or power extracted changes at a substation. The running time of the DC powerflow code on an i86-based PC running Linux was less than a minute.

In future work, we would like to take advantage of the sparse nature of the system of equations to reduce the memory requirements and to parallelize the code.

6 Conclusions

This paper introduces *Marketecture*, and describes its various features in detail. Marketecture is a simulation based tool for analyzing electricity markets. It can provide insight into the qualitative dynamics of the markets and verify commonly held intuitions about how markets respond to changes in the pricing strategies and gaming opportunities with a lot of detail, accuracy and realism. It provides a large degree of flexibility via its ability to vary levels of aggregation, cost functions, demand functions, market clearing rules and matching algorithms. These capabilities, combined with the use of a simple language for configuring a simulation run, enable a wide range of market studies.

We perform a case study for the city of Portland, Oregon using the BILATERALMARKET and POOLCOMARKET. We evaluate the efficiency and performance of these markets under different sellers’ strategies and market clearing algorithms. Our results show that the markets are pareto efficient if sellers’ bid competitively and *normal* clearing is used in the POOLCOMARKET. The second best outcome in the POOLCOMARKET occurs when *Vickrey* auction clearing mechanism is able to induce SELLER to reveal their true production costs and bid at competitive levels. The market clearing price can be reduced further if *weighted average* clearing method is used to clear the POOLCOMARKET. However, *weighted average* clearing results in lower MCQ and higher dead weight loss making it an unattractive alternative.

The average contract price in the BILATERALMARKET is significantly less than the average price in the POOLCOMARKET for all three strategies. This is due to the fact that in our model, BILATERALMARKET SELLER use their ATC curves to negotiate a price. Also, in the POOLCOMARKET, every SELLER gets paid the same price per unit as bid by the marginal SELLER. This results in a more inefficient uniform pricing. Our study also shows that different orders of market execution can be important in significantly affecting the performance of the markets. Based on two different execution orders, our model finds that the overall market performance is better when POOLCOMARKET is run before the BILATERALMARKET.

Acknowledgments

Many people in addition to the authors have contributed towards this project. Foremost among them are Brian Bush, G. Tom Denniston, Martin Drozda, Stephen Eubank, Kriste Henson, Andy McCown, Susan Mniszewski, Fred Roach, Doug Roberts and James Smith.

References

- [1] J. Bernard, R. Zimmerman, W. Schulze, R. Thomas, T. Mount and R. Schuler. Alternative Auction Institutions for Purchasing Electric Power: An Experimental Examination, Bulk Power System Dynamics and Control IV: Restructuring, August, Santorini, Greece. 1998.
- [2] S. Borenstein, J. Bushnell and F. Wolak. Diagnosing Market Power in California's Restructured Wholesale Electricity Market. <http://www.stanford.edu/wolak> (2000).
- [3] S. Borenstein and J. Bushnell. An Empirical Analysis of Market Power in a Deregulated California Electricity Market. University of California Energy Institute, Berkeley (1996).
- [4] J. Bower and D. Bunn. A Model-Based Comparison of Pool and Bilateral Market Mechanisms for Electricity Trading. London Business School, Energy Markets Group, (1999).
- [5] D. Bunn and J. Bower. Experimental analysis of the efficiency of uniform-price versus discriminatory auctions in the England and Wales electricity market. *Journal of Economic Dynamics and Control*, 2001, Vol 25:3,4 p 561-592.
- [6] D. Bunn and F.S. Oliveira. Agent-based simulation - an application to the new electricity trading arrangements of England and Wales. *IEEE Transactions on Evolutionary Computation*, 2001, Vol 5:5 p 493-503.
- [7] J. Bushnell, C. Knittel and F. Wolak. Estimating the Opportunities for Market Power in a Deregulated Wisconsin Electricity Market. http://www.customersfirst.org/marketpower_es.shtml.
- [8] J. Cardell, C. Hitt and W.W. Hogan. Market Power and Strategic Interaction in Electricity Networks. *Resource and Energy Economics*, vol 19, 109-137, (1997).
- [9] J. Carnes. Why Use UTM Coordinates. <http://www.maptools.com/UsingUTM/whyUTM.html> (2001).
- [10] M. J. Denton, S. J. Rassenti and V. L. Smith. Spot Market Mechanism Design and Competitive Issues in Electric Power. *Proceedings of the 31st International Conference on System Sciences: Restructuring of Electric Power Industry*, (1998).
- [11] L.J. Dowell. How Diakoptics Works: An Example Power-Flow Problem *Los Alamos National Laboratory, LA-UR-99-6847*, (1999).
- [12] J. Dowell, M. Drozda, D. Hendersen, V. Loose, M. Marathe, D. Roberts. Electric Industry Simulation System (ELISIMS), *Los Alamos National Laboratory*, technical report, LA-UR-00-1572, 2000.
- [13] N.M. Fehr and D. Harbord. Spot Market Competition in the UK Electricity Industry. *The Economic Journal*, vol. 103, Issue 418, 531-546, (1993).
- [14] D. Friedman. On the Efficiency of Experimental Double Auction Markets. *The American Economic Review*, vol. 74, Issue 1, 60-72, (1984).
- [15] D.K. Gode and S. Sunder. Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality. *Journal of Political Economy*, vol. 101, no. 1, 119-137, (1993).
- [16] J. E. Goodman and J. O'Rourke. Handbook of Discrete and Computational Geometry. CRC Press, (1997).
- [17] R. Green and D. Newbery. Competition in the British Electricity Spot Market. *Journal of Political Economy*, vol. 100, No. 5, pp. 929-953, (1992).

- [18] R. Green. "Increasing Competition in British Electricity Spot Market", *Journal of Industrial Economics*, vol 44(2), pp. 205-216, (1997).
- [19] W. Hogan. A Market Power Model with Strategic Interaction in Electricity Networks. Working Paper, Center for Business and Government, J.F. Kennedy School of Government, Harvard University (1997).
- [20] LCG Consulting. Competitive Power Market Analysis, October 1996. <http://www.energyonline.com/reports/reportsLCG2.asp>
- [21] Mathew Morey. Power Market Auction Design: Rules and Lessons in Market Based Control for the New Electricity Industry. Prepared for Edison Electric Institute, September 2001.
- [22] J. Nicolaisen, V. Petrov and L. Tesfatsion. Market Power and Efficiency in a Computational Electricity Market with Discriminatory Double-Auction Pricing. *IEEE Transactions on Evolutionary Computation*, vol. 5, No. 5, October 2001, pp. 546-560 (2001).
- [23] M. A. Olson, S. J. Rassenti, V. L. Smith, M. L. Rigdon and M. J. Ziegler. Market Design and Motivated Human Trading Behavior in Electricity Markets. *Forthcoming in IIE Transactions on Operations Engineering*, (2001).
- [24] C. R. Plot and S. Sunder. Rational Expectations and the Aggregation of Diverse Information in Laboratory Security Markets. *Econometrica*, vol. 56, No. 5, 1085-1118, (1988).
- [25] K. Rose. Competition in Wholesale Power Markets, April 2001. <http://www.nga.org/cda/files/ElecRose.pdf>
- [26] A. Rudkevich, M. Duckworth and R. Rosen. Modeling Electricity pricing in a Deregulated Generation Industry: The Potential for Oligopoly Pricing in a Poolco. <http://www.tellus.org/energy/>, (1997).
- [27] J. Rust, J. H. Miller and R. Palmer. Characterizing Effective Trading Strategies: Insights from a Computerized Double Auction Tournament. *Journal of Economic Dynamics and Control*, 18, 61-96, (1994).
- [28] V. L. Smith. An Experimental Study of Competitive Market Behavior. *The Journal of Political Economy*, Vol 70, Issue 2, 111-137, (1962).
- [29] Urban Population Simulation Technology. Los Alamos National Laboratory. <http://truchas.lanl.gov/jmh/transims.html>
- [30] Public Service Commission of Utah. Report to the Electrical Deregulation and Customer Choice Task Force", <http://www.psc.state.ut.us/elec/Mktpwrpt.htm>, (1998).
- [31] USGS Mapping Applications Center The Universal Transverse Mercator (UTM) Grid: Fact Sheet 077-01, <http://mac.usgs.gov/mac/isb/pubs/factsheets/fs07701.html> August (2001).
- [32] W. Vickrey. Counterspeculation, Auctions and Competitive Sealed Bidders. *Journal of Finance*, Vol. 16, no. 1, 8-37, (1961).
- [33] F. Wolak. An Empirical Analysis of the Impact of Hedge Contracts on Bidding Behavior in a Competitive Electricity Market. <http://www.stanford.edu/wolak>, (2000).
- [34] J. Weiss. Market Power Issues in a Restructuring of the Electricity Industry: An Experimental Investigation. <http://www.ksg.harvard.edu/hepg/Jurgen2.htm>.
- [35] C. Wolfram. Measuring Duopoly Power in the British Electricity Spot Market. *American Economic Review*, 89, pages 805-826, (1999).

[36] A. J. Wood and B. F. Wollenberg. Power Generation, Operation, and Control *John Wiley & Sons*, (1984).