Market-based Mechanism for Option Pricing

Sarvar Abdullaev[1], Peter McBurney[1], and Katarzyna Musial[1]

Department of Informatics, King's College London
{sarvar.abdullaev,peter.mcburney,katarzyna.musial}@kcl.ac.uk

Abstract. This paper provides a new insight to option pricing from a mechanism design perspective. We simulate a real market environment using double-auction (DA) and combinatorial exchange (CE) mechanisms with different market clearing rules. In particular, we test VCG mechanism in calculating the option prices. We use direct mechanisms relying on the revelation principle which assumes traders' truthfulness. While implementing these market mechanisms, we borrowed ideas from commonly referenced works in auction design such as McAffee's DA and Parkes *et al.* 's ICE. Implementation of our mechanisms involved the use of modern ILP tools such as Gurobi and CVX Research for solving winner determination and price clearing problems efficiently. An important aspect of our work is to understand how option trading strategies frequently used by option traders may influence option prices. Therefore we simulate option prices in DA where goods are traded in single-item bids, and in CE where goods are traded in bundles. For simulating the underlying asset prices, we run a geometric Wiener process. We also compare the resulting option prices with standard Black-Scholes prices.

1 Introduction

Standard financial theory provides a number of methods for calculating option prices based on the market performance of an underlying asset. But there are few models that take into account self-interested agents trading options, and their role in forming the prices. It is commonly assumed that an individual trader is mostly a price-taker, and therefore her influence on the market is insignificant. But in reality, traders with their aggregated utilities form the market prices. Although it is almost impossible to know how each individual agent would evaluate the risk in the market, we can still model them with reasonable properties such as rationality, competitive behaviour, and risk-neutrality. This would provide a testable environment where various market mechanisms, trading behaviours can be simulated. In fact, the main goal of this paper is to show how close option prices can be when they are derived from an option market simulated using self-interested agents or calculated using a Black-Scholes model.

There has been a growing interest in the research of markets as complex game-theoretic systems since Myerson coined *mechanism design* as a framework for strategic interactions between self-interested agents [MY83]. A new discipline of *auction theory* emerged as a part of mechanism design, and it found its applications in solving many of well-known problems such as resource allocation, scheduling, supply chain optimization, operations control and multi-agent system implementation [SP11,YV10]. Similarly it has been also used in simulating financial markets [SK01].

In this paper, we will combine two disciplines, auction theory and option pricing, to build a mechanism which could be used for pricing options. We will consider two specific mechanisms: Double Auction (DA) and Combinatorial Exchange (CE), and their theoretical frameworks. We will also propose tractable implementations for them. We use modern software tools such as CVX[1] for describing LP problems, and Gurobi solver [2] for computing the results. For running the overall simulation, we use MATLAB and its core components. We consider only European options on asset prices generated through a geometric Wiener process. We also review some of the basic trading strategies such as bullish, bearish and butterfly spreads, and use them for determining agents' supply and demand for options. We generate option prices using a Black-Scholes model and compare them with our simulation results from DA and CE.

The paper is organised as follows. In Section 2, we review relevant literature and recently published work in the field of auction theory and option pricing. Section 3 provides the basic framework within which we will construct our simulation model. We define fundamental concepts used in auction theory and review the main aspects of option pricing. In Section 4, we present our simulation model, and walk through its flow. Section 5 provides experimental results obtained from our simulation. Finally, in Section 6 we make concluding remarks for presented work, and provide our future plans towards enhancing our understanding on this topic.

2 Related Works

The ultimate goal of any auction is the allocation of scarce resources to agents. The space of auction types is limitless, because they may vary in their initial settings, bidding rules, market clearing methods etc. Parsons describes more than 30 variations of auctions based on properties such as dimensionality, quantity and heterogeneity of traded items; direction, sidedness, openness of accepted bids; and kth order prices in determining winners[SP11].

We will focus more on DAs and CEs in this paper. DA is an auction mechanism which involves sellers and buyers trading identical goods using single-item bids. McAfee laid the foundations of DAs specifying the dominant strategy implementation of its mechanism, and defined the optimal rules for aggregating and clearing the bids [MC92]. A little later, Friedman and Rust coined Continuous Double Auctions (CDA) where orders are cleared as soon as they arrive[FR93]. Another subtype of DAs is Periodic Double Auctions (PDA) where the market mechanism is split into subsequent phases in which a market-maker first collects

[1] CVX Research, Inc. CVX: Matlab software for disciplined convex programming, version 2.0. http://cvxr.com/cvx, February 2014.

[2] Gurobi 5.60, Commercial parallel MILP solver, http://www.gurobi.com, February 2014.

bids and asks, and then provides provisional allocations and prices. In this way, a market-maker iteratively elicits preferences from traders, and optimally clears the market. This sort of clearinghouse-like mechanism gives rise to generalization of DAs - CEs which involve a combinatorial matching of bids and asks that may have complementarity or substitutability values when dealt in bundles. There are substantial differences in solving DA and CE problems, as they involve different mathematical and computational challenges. In this sense, CE is more complex in its theoretical formulation and implementation. For example, unlike DA, CE has to implement a bidding language which should express exponentially large number of combinatorial alternatives in a concise and comprehensive way. Propositional logic is often used to join trader preferences and encode them into a well-defined bidding language [NN06]. Bidding languages in CE transform to the formulation of yet another complex problem called the winner determination (WD) problem whose solution is an efficient allocation of goods. Weighted set-packing problem can be reduced to a WD problem for CE due to its nature of settling overlapping sets of bids and asks. It is well-known that this problem belongs to the class of NP-hard problems. We will discuss more about the WD problem and its formal definition in the next section of this paper.

There have been a number of researches accomplished in applying market-based mechanisms to financial derivatives markets. King *et al.* has described a multi-agent model for derivatives market which used Gaia methodology [AK05]. Espinosa developed a software solution for resource allocation using the power of options and market-based systems [OE08].

3 Preliminaries

In this section, we define a formal framework for our proposed simulation model and explain some of the concepts that we use throughout this paper. For a given quasilinear auction setting (N, G, λ^0, v), we will implement a direct mechanism (χ, ρ) with an objective function W. Now let us go through each element of this statement:

- N - is the set of traders, indexed by $i \in \{1, 2, ..., n\}$
- G - is the set of goods (in the context of this paper, options), indexed by $j \in \{1, 2, ..., m\}$
- $\lambda^0 \in \mathbb{Z}^{m \times n}$ represents the initial allocation matrix (i.e. supply and demand matrix) where rows are the goods, and columns are the agents. The elements of the matrix are quantities demanded or supplied. Positive quantities mean buying (i.e. demand), negative quantities mean selling (i.e. supply). Initial allocation can be also written in terms of non-negative integers as $\lambda^0 = \lambda^0_B - \lambda^0_A$ where λ^0_B represents demand, and λ^0_A represents supply.
- v is a valuation function whose output changes based on mechanism type. DA requires linear prices for each item, therefore $v : \mathbb{Z}^{m \times n} \to \mathbb{R}^{m \times n}$ takes some allocation matrix λ and maps it to another matrix of the same size with real values. This will represent each agent's valuation for each item.

However, in CE, $v : \mathbb{Z}^{m \times n} \to \mathbb{R}^n$ accepts λ of the same size, but returns only a vector $\mathbb{R}^n$, as agents evaluate items in bundles.
- $\lambda = \chi(v, \lambda^0)$ is an allocation function χ which returns feasible allocation λ for any given valuation function v and initial allocation λ^0.
- $p = \rho(v, \lambda)$ is a payment function ρ which returns anonymous (i.e. same for all agents) linear prices $p \in \mathbb{R}^m$ for each item.
- W is an objective function of a mechanism which implements an auction setting. The mechanism is said to be implemented, if W is either maximised or minimised.

The reason why we use matrix λ where traders disclose their demand and supply for each item is that we will not implement a bidding language for representing trader preferences in a sophisticated way as it has been proposed by Nisan [NN06] using propositional logic or using a tree-based bidding language proposed by Parkes [PK08]. We will simply assume that trader bids can be either divisible or indivisible, and the quantities indicate their preferences over given items.

We will implement a direct mechanism based on the *revelation principle* which states that for any indirect mechanism, there exists a truthful and direct mechanism such that its outcome is at least as good as in an indirect mechanism [YV08]. This leads to another assumption about the truthfulness of agents in declaring their valuations. Nonetheless, we will also use a strategy-proof mechanism such as VCG to make incentive compatibility a dominant strategy for agents. As it has been stated above, auctions will be implemented in quasilinear setting which implies that the utilities of agents must be as follows:

$$u_i(\lambda_i) = v_i(\lambda_i) - p^T \lambda_i, \forall i \in N \tag{1}$$

The utility function requires two types of outcomes for given mechanism: λ_i allocation of goods and p clearing prices. $\lambda_{ij} \in \lambda_i$ determines if trader i wishes to buy or sell item j based on whether it is positive or negative respectively. So linear prices for a pure seller will be negative, and for a pure buyer positive. We will assume that the sign of the valuation function $v_i(\lambda_i)$ will depend on the sign of $\lambda_{ij} \in \lambda_i$, which means that for a pure buyer it has to be positive, for a pure seller negative. If trader i increases his volume for any given item j, it should not decrease his valuation for a previous allocation. In this way, we assume that the valuation function is monotonic. Formally, we define it as follows:

Definition 1. *The valuation function v_i is* monotonic*, if for given two quantities $\lambda_{ij} \in \lambda$ and $\lambda'_{ij} \in \lambda'$ such that $\lambda_{ij} \geq \lambda'_{ij}$ holds for every j, then it must be the case that $v_i(\lambda_i) \geq v_i(\lambda'_i)$.*

In the utility function, we assume that agents are risk-neutral; therefore we use linear pricing $p^T \lambda_i$ to be neutral in amounts and dismiss any budget constraints. However risk-neutrality in the context of option pricing must not be confused. We will assume that every agent will have her own future price expectation (which might not be a risk-neutral forecast) and agent evaluates options based on this factor.

3.1 Auction Design

Auction mechanism is supposed to change the initial allocation to a more preferred one, thus increase utility for every agent. Let us define $\Delta\lambda \in \mathbb{Z}^{m \times n}$ to denote the change in allocation, or just call it *trade*. It can also be written with non-negative integers $\Delta\lambda = \Delta\lambda_A - \Delta\lambda_B$ where $\Delta\lambda_A$ denote change in asks, and $\Delta\lambda_B$ change in bids. Then the final allocation is equal to $\lambda = \lambda^0 + \Delta\lambda$. Let $v(\Delta\lambda)$ denote change in value for each agent. In quasilinear setting, efficient strategy must maximise the social welfare which is the maximisation of the sum of all trader valuations. We can drop constant term $v(Q^0)$ from objective function, and consider efficient trade as a maximised growth in social welfare.

Definition 2. *For given auction* (v, λ^0)*, efficient trade* $\Delta\lambda^*$ *solves:*

$$\max_{\Delta\lambda} \sum_i v_i(\Delta\lambda_i) \tag{2}$$

$$s.t. \quad \lambda^0_{A,ij} - \Delta\lambda_{A,ij} \geq 0, \quad \forall i, \forall j \tag{3}$$

$$\lambda^0_{B,ij} - \Delta\lambda_{B,ij} \geq 0, \quad \forall i, \forall j \tag{4}$$

$$\sum_i \Delta\lambda_{ij} = 0, \quad \forall j \tag{5}$$

$$\Delta\lambda_{ij} \in \mathbb{Z}$$

Constraint (3) ensures that no trader sells more than she owns, and (4) ensures that no buyer buys more than she wants. Constraint (5) enforces that every sold item must be bought by some agent. All trades satisfying constraints (3), (4) and (5) constitute the set of *feasible trades* which is denoted as $\mathcal{F}(\lambda^0)$. We will consider using χ as a function which returns the efficient trade for given initial allocation and valuation profile.

Now we will review *competitive equilibrium* prices which maximise everyone's utility. We will denote equilibrium prices as p^*, and corresponding cost part of the quasilinear utility can be written as $p^{*T}\Delta\lambda_i$. Formally, competitive equilibrium is defined below [PK08]:

Definition 3. p^* *is said to be in competitive equilibrium for* (v, λ^0) *if there is some feasible trade* $\Delta\lambda \in \mathcal{F}(\lambda^0)$ *such that:*

$$v_i(\Delta\lambda_i) - p^{*T}\Delta\lambda_i \geq v_i(\Delta\lambda'_i) - p^{*T}\Delta\lambda'_i, \quad \forall \Delta\lambda'_i \in \mathcal{F}_i(\lambda^0) \tag{6}$$

It is very unlikely that equilibrium prices are found in practice, therefore δ-approximate competitive equilibrium prices are used more often.

Definition 4. p^* *is said to be in* δ*-competitive equilibrium for* (v, λ^0) *and* $\delta \in \mathbb{R}_{\geq 0}$ *if there is some feasible trade* $\Delta\lambda \in \mathcal{F}(\lambda^0)$ *such that:*

$$v_i(\Delta\lambda_i) - p^{*T}\Delta\lambda_i + \delta \geq v_i(\Delta\lambda'_i) - p^{*T}\Delta\lambda'_i, \quad \forall \Delta\lambda'_i \in \mathcal{F}_i(\lambda^0) \tag{7}$$

Consider a function $v(\lambda^0)$ that returns valuation matrix $m \times n$, where each agent i discloses his value for every individual item j. Then the surplus-maximising WD problem for DA can be written as following ILP:

Definition 5. *For given double auction* (v, λ^0)*, surplus-maximising WD problem is*

$$\max_{\Delta\lambda} \sum_i \sum_j v_{ij} \Delta\lambda_{ij} \tag{8}$$

$$s.t. \quad \Delta\lambda_{ij} = \lambda^0_{ij} x_{ij} \quad \forall i, \forall j \tag{9}$$

$$\sum_i \Delta\lambda_{ij} = 0, \quad \forall j \tag{10}$$

$$\Delta\lambda_{ij} \in \mathbb{Z}, \quad x_{ij} \in \{0, 1\}$$

where x_{ij} *is a binary decision variable.*

Constraint (9) can be derived directly from feasibility constraints (3) and (4). $\lambda^0_{A,ij} - x_{ij}\Delta\lambda_{A,ij} \geq 0$ is true where x_{ij} can take only $\{0, 1\}$. It is symmetric to (4) as well. As it is seen from formulation, in DA, quantities are not fractional, so individual bid or ask on item j must satisfy the requested amount, otherwise it is not accepted. Another way of finding an efficient allocation in DAs is using overlapping bids and asks. In this policy, mechanism first sorts bids in descending order $v_{B,1}(\{j\}) \geq v_{B,2}(\{j\}) \geq \cdots \geq v_{B,n}(\{j\})$ and asks in ascending order $v_{A,1}(\{j\}) \leq v_{A,2}(\{j\}) \leq \cdots \leq v_{A,n}(\{j\})$. Then it picks the first $l > 0$ bids and asks, where l is the maximal index for which $B_l \geq A_l$. These overlapping bids will determine the winners of DA [PK04,MC92]. This will maximise the surplus of the allocation.

However in CE, all combination of bids and asks must be satisfied. The formulation of WD problem for CE is similar to DA, however the decision variable is not a matrix, but a vector of $x \in \{0, 1\}^n$ for each trader. Similarly, the valuation function for given trade $\Delta\lambda$ returns a vector of $\mathbb{R}^n$, taking every agent's bid as an indivisible bundle, and assigning joint valuation to it. Mechanism decides whose bids to satisfy fully in order to maximise the surplus. Below is the formal definition:

Definition 6. *For given combinatorial exchange* (v, λ^0)*, surplus-maximising WD problem is*

$$\max_{\Delta\lambda} \sum_i v_i(\Delta\lambda_i) \tag{11}$$

$$s.t. \quad \Delta\lambda_i = \lambda^0_i x_i \quad \forall i \tag{12}$$

$$\sum_i \Delta\lambda_{ij} = 0, \quad \forall j \tag{13}$$

$$\Delta\lambda_{ij} \in \mathbb{Z}, \quad x_i \in \{0, 1\}$$

where x_i *is a binary decision variable,* λ^0_i *is an initial allocation for agent* i*.*

We will use McAfee's price clearing method [MC92] for DA and VCG for CE. McAfee DA pricing is budget-balanced, strategy-proof and individual rational mechanism. However VCG is not a budget-balanced mechanism [MY83], although it is efficient, incentive compatible and individual rational. Therefore we will find clearing prices that minimise the worst error between VCG payments and make them fixed for every trader. This will naturally make exchange budget-balanced. This idea is slightly modified version of Parkes *et al.* 's *Threshold rule* [PK08].

Definition 7. *For given lists of ordered bids (in descending order) and asks (in ascending order) if there exists $l > 0$ such that $v_{B,l}(\{j\}) \geq v_{A,l}(\{j\})$ and $\lambda^0_{A,lj} \geq \lambda^0_{B,lj}$ (free disposal rule), then McAfee DA price for item j is:*

$$\rho^*_j = \frac{v_{A,l+1}(\{j\}) + v_{B,l+1}(\{j\})}{2}, \quad \forall j \in G \tag{14}$$

In VCG mechanism prices are computed as the overall social cost that agent i introduces to market. Formally it is defined below:

Definition 8. *For given combinatorial exchange (v, λ^0), VCG payments are*

$$\rho_{vcg,i} = \sum_{t \neq i} v_t(\chi(v_{-i})) - \sum_{t \neq i} v_t(\chi(v)), \quad \forall t, i \in N \tag{15}$$

$$s.t. \quad \chi(v) = \underset{\Delta\lambda \in \mathcal{F}(\lambda^0)}{\arg\max} \sum_i v_i(\Delta\lambda_i) \tag{16}$$

where v_{-i} is a valuation profile without agent i.

It can be seen from VCG payments that it generates specific payment for each trader and these payments once aggregated may result in surplus or deficit for the mechanism. In order to distribute surplus among traders, we will minimise the worst-case difference between VCG payment $\rho_{vcg,i}$ and clearing linear prices $\rho^{*T}\Delta\lambda_i$. This will approximate competitive equilibrium prices with ε error.

Definition 9. *For given combinatorial exchange (v, λ^0) and VCG payments $\rho_{vcg} \in \mathbb{R}^n$, there are anonymous clearing prices $\rho^* \in \mathbb{R}^m$ defined as:*

$$\rho^* = \arg\min_{\rho} \varepsilon \tag{17}$$

$$s.t. \quad \rho^T\Delta\lambda_i - \rho_{vcg,i} \leq \varepsilon, \quad \forall i \tag{18}$$

$$-(\rho^T\Delta\lambda_i - \rho_{vcg,i}) \leq \varepsilon, \quad \forall i \tag{19}$$

$$\rho_j \geq 0, \quad \forall j \in G, \quad \varepsilon > 0 \tag{20}$$

where ρ^* is the vector of clearing prices for each item j.

When prices are anonymous, and supply and demand quantities match (5), all payments will be fully received or paid and thus will make mechanism generate zero revenue. This makes it strictly budget-balanced. Mechanism is also incentive compatible because it satisfies necessity and sufficiency conditions for ρ and χ functions. Payment for each agent i is calculated as $\rho_i = \rho^{*T}\Delta\lambda_i$. Necessity condition for incentive compatibility states [YV08] that payment function for agent i must involve only the valuations of other agents v_{-i} and agent i's own choice (i.e. bid) $\Delta\lambda_i$. From (17), we know that ρ^* is calculated directly from the output of VCG payments, which in turn involves only v_{-i} in computing payment for agent i. Agent's own choice $\Delta\lambda_i$ is derived from his initial bid λ_i^0 using a decision variable x_i (12). Hence the necessity condition is met. Sufficiency condition states [YV08] that a choice function (i.e. allocation function) χ must provide utility maximising allocation for each agent i. From the definition of WD problem (11), we can see that the solution of the WD problem finds the best allocation which maximises overall utility, thus satisfying the sufficiency condition.

3.2 Options

In this section, we will provide some basic notions about European options and how they are priced. An option is a financial contract which provides to its holder the right of buying or selling certain assets at an agreed future price (i.e. strike price). The one who sells (writes) them takes the liability to fulfil buy or sell requests in exchange for the premium he receives. European options are exercised upon their maturity date. An option allowing its holder to buy is named a *call option*, and allowing to sell is a *put option* [HL01]. Depending on the present value of its strike price K and the current price of its underlying asset S_t, options can be classified into *Out-of-The-Money* (OTM), *At-The-Money* (ATM) and *In-The-Money* (ITM) options. The table below illustrates the types of options that are traded in exchanges. We can also define the upper and lower boundaries for option valuation in equations (21), (22).

Option Types

	OTM	ATM	ITM
CALL	$K > S_t$	$K = S_t$	$K < S_t$
PUT	$K < S_t$	$K = S_t$	$K > S_t$

$$\max(S_t - K, 0) \leq c \leq S_t \qquad (21)$$

$$\max(K - S_t, 0) \leq p \leq K \qquad (22)$$

For simplicity reasons, we will assume that the risk-free interest rate is zero, so money has no time value. Also there is no friction in the market, so options can be sold and bought at the same price without any transaction costs involved.

There is an established relationship between put and call options with the same strike price and maturity date. This relationship results from the possibility of buying the one and selling the other. Consider a case, when trader buys a call option at K strike price, and at the same time sells a put option with K strike price, and both have the same maturity T. In some sense, it seems that trader can compensate the cost of a call option he bought for with the premium he received for selling put. So on maturity date, S_T turns out to be higher than strike price K, so the trader can benefit profit as a difference of $S_T - K$. However if S_T appears to be less than K, then trader has a liability to fulfil the put option that he sold, so he incurs a loss of $K - S_T$. This market position actually simulates a forward contract which could be obtained for free. This type of contract is free because it involves future possible liability or profit at the same time, so the risk for both parties is even. Once the combination of put and call options can replicate the liabilities of a forward contract, the prices for put and call options must hold the *put-call parity* relationship: $(c + K = p + S_T)$ [HM04]. Using the put-call parity relationship, we can easily convert call prices to put prices, and vice versa.

4 Simulation Model

We break down our simulation into three steps. Firstly, we generate bids for agents with heterogeneous beliefs on future asset prices. To obtain these forecasts, we simulate a Wiener process for each agent. Then agents are required to pick one of the *option trading strategies* based on their beliefs. Option trading strategy (OTS) is a vector of quantities of options that each agent needs to buy or sell in order to make a profit based on his forecasted asset price. Then agents calculate their private values for each option type and submit their bids to the DA and CE mechanisms. The second step solves DA and CE problems to find linear prices for each option type. In DA, winners are determined by solving (8), and clearing prices are calculated using McAfee method (14). In CE, winners are determined through surplus-maximisation formulated in (11). Then VCG payments are calculated using (15), and the resulting payments are used to find minimised worst-error prices (17) to clear the market. Finally, in the third step, we collect linear prices from both mechanisms and compare them with Black-Scholes prices.

4.1 Bid Generation

We first generate asset price forecasts for each agent i using a geometric Wiener process. Based on predefined volatility σ, initial asset price S_0 and white noise $z \sim \mathcal{N}(0, 1)$ we can calculate possible asset price outcome at time T. Also we assumed that risk-free rate r is zero. Below is the formula that we use:

$$S_{i,T} = S_0 \exp(-\frac{1}{2}\sigma^2 T + \sigma\sqrt{T}z) \quad \forall i \in N \tag{23}$$

Then every agent calculates his own values for OTM, ATM and ITM puts and calls separately. We use some ϵ variation from ATM strike price K to create OTM and ITM options. Afterwards agents pick one of the OTSs listed in Table 1 where the quantity of option type to buy or sell is specified in positive or negative numbers respectively. They will choose OTS based on its direction that matches their forecast. For example if agent's forecast in between $S_0 - \epsilon \leq S_{i,T} \leq S_0 + \epsilon$, then agent will choose *neutral strategy.* If $S_{i,T} > S_0 + \epsilon$, then agent will choose *bullish strategy.* And finally if $S_{i,T} < S_0 - \epsilon$, the agent will choose bearish strategy. Traders pick random strategy among strategies with same direction. However some OTSs can be both bullish and bearish such as Long Straddle, so both bullish and bearish traders can be interested in this OTS. It is also possible that OTS is more bullish, than bearish, and vice versa. For example, Strip generates greater payoff when prices go up. Therefore there is a biased chance for a bearish trader to choose Strip among other bearish OTSs because it is less bullish.

Algorithm 1 Bid Generation

$Ks \leftarrow \{S_0, S_0+\epsilon, S_0-\epsilon\}$ {Array of strike prices}
$priceFuncs \leftarrow \{(s_t,k)(\min(\max(s_t-k,0),S_0)),(s_t,k)(\min(\max(k-s_t,0),k))\}$
for $i \in N$ **do**
  $z \leftarrow randn(0,1)$ {random variable from $\mathcal{N}(0,1)$}
  $S_i \leftarrow S_0 \exp(-\frac{1}{2}\sigma^2 T + \sigma\sqrt{T}z), j \leftarrow 1$
  for $k \in Ks$ **do**
    for $val \in priceFuncs$ **do**
      $v_{ij} \leftarrow val(S_i, k)$
      $j \leftarrow j+1$
    end for
  end for
  $\lambda_i = pickOTS(S_0, S_T, \epsilon)$
  $DA_i \leftarrow \{\lambda_i, v_i\}$
  $CE_i \leftarrow \{\lambda_i, \sum_j \lambda_{ij} v_{ij}\}$
end for
return $\{DA, CE\}$

We sum up the process of bid generation for all agents in Algorithm 1. In this algorithm, bidders evaluate their forecasts, calculate their private values for each option, and then pick an appropriate OTS. Finally, they submit their bids to DA and CE. They submit individual option valuations to DA, and bundled valuations to CE.

Table 1. Option Trading Strategies

Name	c_{ATM}	p_{ATM}	c_{OTM}	p_{OTM}	c_{ITM}	p_{ITM}	Direction
Long Call	1	0	0	0	0	0	bullish
Long Put	0	1	0	0	0	0	bearish
Bull Call Spread	0	0	-1	0	1	0	bullish
Bear Call Spread	0	0	1	0	-1	0	bearish
Butterfly Put Spread	0	-2	0	1	0	1	neutral
Short Put Ladder	0	1	0	1	0	-1	bearish > bullish
Iron Butterfly	-1	-1	1	1	0	0	neutral
Short Straddle	-1	-1	0	0	0	0	neutral
Long Strangle	0	0	1	1	0	0	bearish and bullish
Short Strangle	0	0	-1	-1	0	0	neutral
Strip	1	2	0	0	0	0	bullish > bearish
Strap	2	1	0	0	0	0	bearish > bullish

4.2 Running Mechanisms

Once we have bids generated, we can feed them into both DA and CE mechanisms. We will run mechanisms several times per each time step, and then average the resulted prices. This will smooth up some noise associated with randomised selection of OTSs, and experimental errors. We continue this routine for every time step t till options reach their maturity date T. Also we update asset price S_t for each time step t through simulating a global Wiener process. Here is the update rule for asset prices:

$$S_{t+1} = S_t \exp(-\frac{1}{2}\sigma^2 T + \sigma\sqrt{T}z) \tag{24}$$

Algorithm 2 is a subroutine for computing DA prices for given DA bids. As it was mentioned before, we solve (8) to determine the winning bids and asks, and then find offsetting maximum bid and minimum ask to calculate their average as a clearing price. In CE, we follow the same routine to find winning allocations. But we calculate bids as bundles, and satisfy them fully. Then we calculate VCG payments, and minimise the worst-case error for these payments to find the final clearing prices. Algorithm 3 illustrates the steps of doing this.

Algorithm 2 Running DA Mechanism

$\Delta\lambda^* \leftarrow \arg\max_{\Delta\lambda\in\mathcal{F}(\lambda^0)} \sum_i \sum_j v_{ij}\Delta\lambda_{ij}$
for $j \in G$ **do**
  $offset\_bid \leftarrow \max(find(v_j|\Delta\lambda^*_j = 0$ **and** $\lambda^0_j > 0))$
  $offset\_ask \leftarrow \min(find(v_j|\Delta\lambda^*_j = 0$ **and** $\lambda^0_j < 0))$
  $\rho^*_j \leftarrow (offset\_bid + offset\_ask)/2$
end for
return ρ^*

Algorithm 3 Running CE Mechanism

$\Delta\lambda^* \leftarrow \arg\max_{\Delta\lambda\in\mathcal{F}(\lambda^0)} \sum_i v_i(\Delta\lambda_i)$
for $i \in N$ **do**
  $\lambda^0_{-i} \leftarrow \lambda^0 \backslash \{\lambda^0_i\}$
  $\Delta\lambda_{-i} \leftarrow \arg\max_{\Delta\lambda\in\mathcal{F}(\lambda^0_{-i})} \sum_{t\neq i} v_t \Delta\lambda_t$
  $\rho_{vcg,i} = \sum_{t\neq i} v_t(\Delta\lambda_{-i,t}) - \sum_{t\neq i} v_t(\Delta\lambda^*_t)$
end for
$\rho^* \leftarrow \arg\min_\rho \delta$, s.t. (18),(19) and (20)
return ρ^*

The output prices of the mechanism may not conform with put-call parity due to surplus-maximising objective in WD problem. In order to enforce this relationship, we will pick the prices only for one type of option, say call, and use put-call parity to find put price. In this way, we can avoid arbitrage associated with buying and selling calls and puts with incompatible prices. This arbitrage opportunity exists because there are other derivatives such as forward contracts which can be obtained for free, and guarantee the same payoff.

5 Experimental Results

We fix several parameters for the experiment. For example, options have constant strike prices throughout the timeline. This means that agents will trade only with predefined set of options at the beginning of the simulation, and no new type of option with new strike price will enter the market. Also option maturity date will be constant, and it will approach its maturity date through the timeline of the simulation. Asset price volatility will also be fixed. We have set the following parameters for the experiment:

Table 2. Parameters of the experiment

Name	Value
Initial Asset Price	$S_0 = 100$
Strike Price	$K = 100$
Deviation from Strike price	$\epsilon = 10$
Asset Price Volatility	$\sigma = 0.05$
Risk-free rate	$r = 0$
Time to maturity	$T = 100$
Number of agents	$N = 100$
Number of option types	$G = 6$
Number of tests per mechanism	$w = 30$

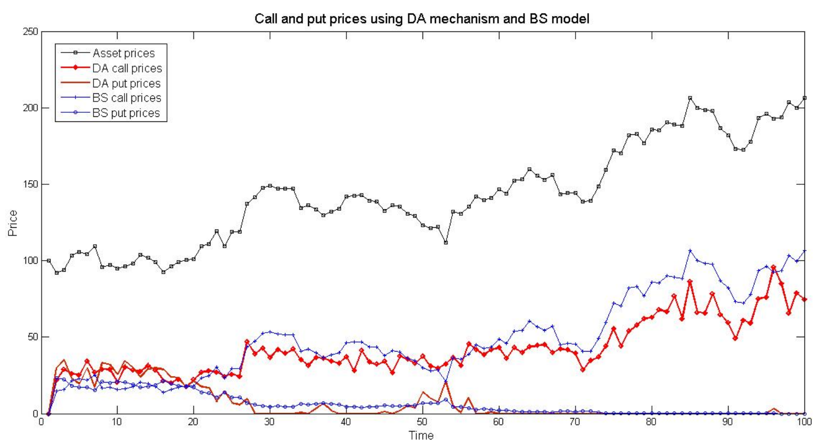

Fig. 1. Call and put prices for DA mechanism and Black-Scholes

In Figure 1 we can see call and put option prices from DA in comparison with Black-Scholes prices. It illustrates that DA mechanism can effectively simulate Black-Scholes prices, as call and put prices for both models move in tandem. Call prices rise because of a continuous increase in asset prices, and this is reflected both in Black-Scholes and the DA models. On the other hand, put prices go down, finally converging to zero. We use put-call parity relationship to covert DA call prices to corresponding put prices, and so does the Black-Scholes model. Figure 2 also shows converging prices in the simulation of CE mechanism and Black-Scholes model. It can be seen that CE call and put prices almost repeat Black-Scholes, and converge on maturity. CE prices are somewhat volatile

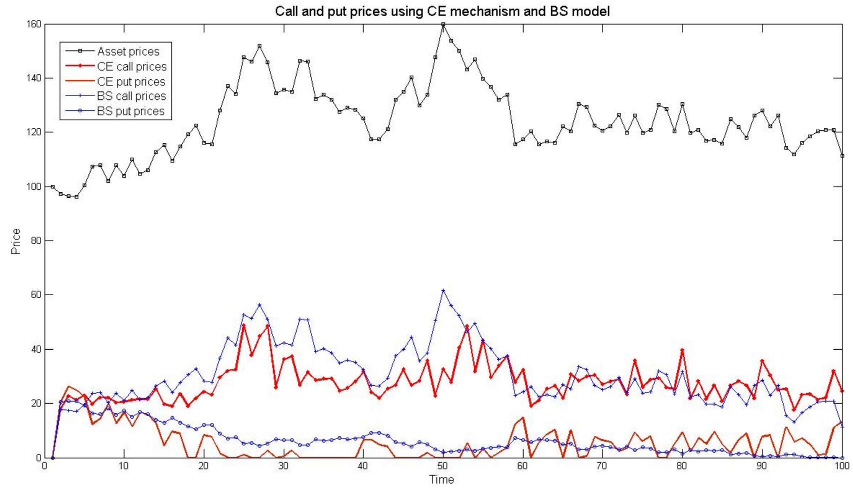

Fig. 2. Call and put prices for CE mechanism and Black-Scholes

compared to Black-Scholes, and this could be the result of bundled valuation of options, as the number of participating traders decrease, and their heterogeneous forecasts have greater effect on price changes. Table 3 presents the aggregated

Table 3. Summary of Key Indicators

	DA	CE
Absolute Error on Black-Scholes	11	6.5
Relative Error on Black-Scholes	24%	20%
Overall Utility: Mean	2547	2146
Trader Participation: Mean	68/100	43/100
Trader Participation: Standard Deviation	7	5
Option Volume Traded: Mean	57	46
Option Volume Traded: Standard Deviation	1	3
Delta of calls ($\Delta c \backslash \Delta S$)	0.35	1.43
Delta of puts ($\Delta p \backslash \Delta S$)	-0.61	-0.26

indicators of the simulation per each timestep. Error-wise both mechanisms performed equally well, as they deviate about 20% from Black-Schole prices. It can be seen from given results that trader participation is higher in DA, because it addresses every bid individually. Therefore more traders can participate in the market, and thus increase the overall utility and the volume. Due to central limit theorem, the more traders get involved in the market, the closer are the prices to an equilibrium point, which makes them more stable. However in CE mechanism, bids are indivisible and must be fully covered. This makes bid sourcing more complex leaving a number of high value bids unsatisfied. As a result the number of active traders decrease and prices become highly dependent on individual minor spikes. This makes option prices in CE more volatile which can be seen from option's high delta too.

6 Concluding Remarks

We have proposed a framework for pricing options in a simulated market environment. We formulated basic problems such WD and price clearing as core element of the mechanism. Then we implemented two popular mechanisms DA and CE to generate option prices. As it can be seen from our experimental results, both mechanisms performs as well as Black-Scholes. In our future work, we would like to study results more thoroughly by conducting sensitivity analysis on various parameters of the simulation such as asset price volatility and trends, trader beliefs, choice of OTSs, and bidding and price clearing methods. We will also closely look into VCG mechanism separately by analysing how VCG payments affect the overall well-being of traders in an option market, and what role it plays in forming the clearing prices. Also it would be interesting to compare our simulation results with real data in order to verify accuracy of the predictions.

References

MY83. Satterthwaite, M. A., Myerson, R. B., Efficient Mechanisms for Biliteral Trading. *Journal of Economic Theory*, 28, 265-281 (1983)

SP11. Parsons, S., Rodriguez-Aguilar, J. a., Klein, M., Auctions and bidding. *ACM Computing Surveys*, 43(2), 1-59. doi:10.1145/1883612.1883617 (2011)

YV10. Shoham, Y., Crampton, P., Steinberg, R., Combinatorial Auctions, MIT Press, US (2010)

OE08. O. Baqueiro Espinosa. *Agent Risk Management in Electronic Markets using Option Derivatives.* PhD thesis, Department of Computer Science, University of Liverpool (2008)

FR93. D. Friedman. The double auction institution: A survey. In *The Double Auction Market: Institutions, Theories and Evidence.* Perseus Publishing (1993)

HM04. D. Higham. *An Introduction to Financial Option Valuation: Mathematics, Stochastics and Computation.* Cambridge University Press (2004)

HL01. J. C. Hull. *Options, Futures and Other Derivatives.* Prentice Hall, 5 edition (2001)

AK05. A. J. King, O. Streltchenko, and Y.Yesha. Using multi-agent simulation to understand trading dynamics of a derivatives market. *Annals of Mathematics and Artificial Intelligence*, 44-3:233–253 (2005)

NN06. N. Nisan. Bidding languages for combinatorial auctions. In P. Crampton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions.* The MIT Press (2006).

PK04. D. Parkes and J. Kalagnanam. Auctions, bidding and exchange design. Technical report, IBM Research Division (2004).

PK08. D. C. Parkes, B. Lubin, A.I. Juda, R. Cavallo, S. Lahaie, and J. Shneidman. ICE: An expressive iterative combinatorial exchange. *Journal of Artificial Intelligence Research*, 33:33–77 (2008)

SK01. K. Saatcioglu, J. Stallaert, and A. Whinston. Design of a financial portal. In *Communications of the ACM*, volume 44, pages 33–38 (2007)

MC92. McAfee, R. P., A dominant strategy double auction. *Journal of Economic Theory, 56, 434-450, (1992)*

YV08. Shoham, Y., Leyton-Brown, K., *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations* Cambridge University Press (2008)