Random Search Techniques for Optimal Bidding in Auction Markets

Shahram Tabandeh and Hannah Michalska *†

Abstract—**Evolutionary algorithms based on stochastic programming are proposed for learning of the optimum bid in auction markets. Sellers and buyers are attempting to learn their optimum bids that maximize their individual utility functions in the next round of the game. Examples of a second-price sealed-bid auction, and double auction markets, with random and merit-based matching processes are considered and good performance of this type of algorithms is confirmed by extensive simulations. The proposed algorithms need no assumptions about the stationary behavior of players, contrary to the needs of competitive algorithms in the class of fictitious play.**

Keywords: Auction market, Learning in games, Optimum bid, Stochastic programming, Matching process.

1 Introduction

Auctions are the basic framework of exchange in many markets and can be analyzed by game theoretical methods. In any game players attempt to maximize their individual utility functions. If the game is repetitive the players can use the past experience to learn to play better in the future.

Learning in games and particularly auctions has been studied extensively; see for example [5][7][16][15][9] , as it permits to improve the strategies of the players in the future based on the observation of the past actions. Diverse information patterns can be considered in this context by which an agent can have partial or complete access to the history of the actions and results of other players.

It is then of primary interest to determine whether the repeated game can converge to some kind of equilibrium. Learning and convergence are particularly difficult when utility functions exhibit discontinuities as is the case of utility functions used in auction markets. The references [12][3] provide a discussion of the existence of equilibria in discontinuous games while references [8][4] give conditions of existence of equilibria in auction markets specifically.

This paper considers different examples of auction markets and develops algorithms for iterative stochastically-based learning of their equilibria. The first algorithm is designed to implement learning of the optimum bid in a second-price sealed-bid auction where the valuation of buyers of the same object is unknown to other buyers. The second algorithm is suited for double auction markets with separate populations of buyers and sellers who attempt to optimize their individual utility functions during the game. In this algorithm buyers and sellers meet each other randomly. The third algorithm is also designed for learning in double auction markets, but buyers and sellers meet each other based on their merits.

All types of auctions presented in this paper are used widely in practice, NYSE and AMEX as examples, where double auctioning rather than the second price auctioning is employed to trade.

Evolutionary algorithms have long been used for learning in games, see e.g. [1][2] which discusses convergence of a genetic algorithm proposed for learning of the equilibrium in a double auction market.

Other approaches to solve the same kind of problem include fictitious play, see [13]. In a fictitious play, the players optimize their actions based on the empirical statistics estimated from the historical actions of their opponents, stationary behavior of opponents is assumed. In partial best response dynamic methods, only some players change their strategies to best reply to previous state of the game, [16]. In other methods, called replicator dynamic, the number of players using the same strategy grows proportionally to the success of that strategy [16].

Random search algorithms, [10][6], are preferred stochastic programming techniques in search of solutions to global or nonsmooth optimization problems. This also motivates their use as a tool for computation of equilibria in discontinuous games such as auction markets considered here.

The algorithm presented here can be adapted and extended to apply to other kinds of auctions and can, for example, prove useful to find equilibria of auctions in electricity markets as discussed in many recent papers, see e.g. [11][14].

The paper is organized as follows. Section II delivers the problem statement. Section III introduces examples of auction markets and presents the algorithm for evolutionary computation of their equilibria. Section IV, delivers conclusions and future re-

*S. Tabnadeh is with the Center for Intelligent Machines, McGill University, Montreal, Qc (email: shahram.tabandeh@mail.mcgill.ca).

†H. Michalska is with the Department of Electrical and Computer Engineering, McGill University, Montreal, Qc (email: hannah.michalska@mcgill.ca).

search directions.

2 Problem Statement

The algorithms presented below are designed to find optimum bids for different auction markets hence the problem statement is presented in the following general form. The assumption is that there are n buyers and n sellers in a market. Every buyer or seller employs his own utility function. When seller j meets buyer i their utility functions are denoted by: $u_{bi}(p_{bi}, p_{sj}) : A_b \times A_s \rightarrow \Re$ and $u_{sj}(p_{bi}, p_{sj}) : A_b \times A_s \rightarrow \Re$. Here, p_{bi} and p_{sj} are the bid and ask prices of buyer i and seller j, respectively, A_b is the feasible set of prices bid by the buyers, and A_s is the feasible set of prices asked by the sellers.

The utility function for buyer i as she meets with several random sellers is further defined as the expected value:

$$U_{bi}(p_{bi}, P_s, f^s) = E_{sj}(u_{bi}(p_{bi}, p_{sj})|p_{bi}) \quad (1)$$

where the P_s is the set of feasible prices offered by the sellers and f^s is the probability distribution function of these prices. Similarly, the utility function for seller j as she meets with several random buyers is:

$$U_{sj}(p_{sj}, P_b, f^b) = E_{bi}(u_{sj}(p_{bi}, p_{sj})|p_{sj}) \quad (2)$$

where the P_b is the set of feasible prices offered by the buyers and f^b is their probability distribution function.

The following definitions are essential for further developments in this paper.

Definition 1 *[1] (*Nash Equilibrium in an auction game*) The set, $\{P_b, P_s, f^b, f^s\}$, where P_b and P_s are sets of all prices bid by the buyers and asked by the sellers, respectively, and f^b and f^s are probability distribution functions over P_b and P_s, respectively, is called a Nash equilibrium of the game if and only if:*

$$U_{bi}(p_{bi}, P_s, f^s) \geq U_{bi}(p'_{bi}, P_s, f^s) \quad \forall i, p'_{bi} \in [0,1] \quad (3)$$

and

$$U_{sj}(p_{sj}, P_b, f^b) \geq U_{sj}(p'_{sj}, P_b, f^b) \quad \forall j, p'_{si} \in [0,1] \quad (4)$$

Definition 2 (Evolution) *Evolution is the process by which the values of the buyers' bids $p_{bi}(k)$ and the sellers' asks $p_{sj}(k)$, $i, j = 1, ..., n$ in round k are updated to their new values in round $k+1$. The evolution process is hence thought to be represented by two mappings*

$$E_b : p_{bi}(k) \mapsto p_{bi}(k+1) \text{ and } E_s : p_{si}(k) \mapsto p_{si}(k+1) \; \forall i, j$$

3 Stochastic Learning in Auction Markets

Two new algorithms implementing evolutionary learning of the optimum bid and ask prices for different types of auctions are presented below. Convergence properties of the novel random algorithms are studied by simulations. The case of a second-price sealed-bid auction market is discussed first. This is followed by three other examples of double auction markets. Extensive simulation results demonstrate the efficiency of the designed algorithms and deliver statistics for convergence to the market equilibria when stochastic learning is applied.

3.1 Second-Price Sealed-Bid Auction Market

A second-price sealed bid auction is considered in which different buyers bid to buy the same object. This object is assumed to have value v_i for buyer i. The valuations of objects are considered private , more precisely, any given buyer does not know the value that the other buyers are attributing to the object which she wants to buy.

The highest bidder gains the object and pays as much as the second highest bid, while anyone else pays as much as she has bid.The following notation is adopted :

– n, the cardinality of the populations of buyers.
– k, the index of the current round of the game, ($k \in \mathbb{Z}$.)
– $v_i \in [0, 1]$, the value of the product for buyer i.
– $p_{bi}(k) \in [0, v_i]$, the maximal price at which buyer i is willing to buy in round k.
– $\sigma > 0$, the variance of the random generator function.
– $N_T(\mu, \sigma)$, the normal distribution with mean μ and variance σ.
– $loop$, maximum number of rounds in the game.
– $mcount$, counter of the round in the game.
– ch, the index of the buyer who is allowed to change his bid.
– $bnew$, candidate for a mutated bid price.
– $u_{bch}(bnew; k)$, the value of the utility function for buyer ch when bidding price $bnew$ in round k of the game.

The algorithm is stated first and is followed by a discussion of its steps. The buyers are allowed to change their bids during the game as they see fit. No more than a single buyer is allowed to mutate his bidding price in any round of the game which places the learning methodology employed in the category of "learning by partial best response". The buyers are taking turns in changing their bids.

Algorithm 1 The Stochastic Optimizer Algorithm For Second-Price Sealed-Bid Auction Market.

Step 0: Set the counter for the round of the game $k = 0$. *Set* n *- the number of buyers and* $loop$ *- the maximal number of rounds in the game. Set the initial value for the bid randomizer* $\sigma > 0$. *For* $i = \{1, ..., n\}$ *draw samples of initial values of the bid prices from uniform distributions over the interval* $[0, v_i]$, *i.e.* $p_{bi}(0) \sim U(0, v_i)$.

Step 1: For $i = \{1, ..., n\}$ *calculate utility functions* $u_{bi}(p_{bi}(k); k)$.

Step 2: For buyer whose index is calculated as $ch = remainder(mcount/n) + 1$ *calculated the candidate for a mutated bid price as:*
$bnew \sim N_T(p_{bch}(k), \sigma)$.

Step 3: Update the bid price $p_{bch}(k) = bnew$ *if* $u_{bch}(bnew; k) > u_{bch}(p_{bi}(k); k)$.

Step 4: Update the counter of rounds $mcount = mcount + 1$, *and go to Step 1 if* $mcount \leq loop$ *.*

In Step 0 of the above algorithm, parameters of the algorithm are initialized, and the bids of the n buyers are drawn from uniform random distributions. In Step 1 the values of the utility functions for all buyers are calculated for the current bid prices. The index ch of Step 2 changes in a manner that allows the buyers to take turns in mutating their bidding prices in accordance with their proper utility function values. The fact that the buyers are allowed to change their prices one by one does not diminish the applicability of the algorithm in practical situations as one can assume that the real-time execution times of Steps 2 and 3 are negligible as compared to real time bidding process in the marketplace. The inequality of Step 3 implies that the bidding prices in the market will move towards achieving best utility values for all buyers.

To test the algorithm, an example with 5 buyers is considered. Values of the common object of interest for these 5 buyers are drawn from a uniform random distribution $U[0, 1]$. Figures 1 and 2 show how the bid prices and utility function values evolve when the private values of the desired object for the 5 players were randomly set to: 0.8785, 0.7110, 0.6611, 0.4396, 0.5628. Note that these happen to be set in favor of the first buyer who, in fact, gains possession of the object as his utility function approaches its valuation. During the evolution of the market, the values of the utility functions for the remaining players approach zero as they do not gain the object of bid.

The utility functions for the buyers in the game are assumed to be given by:

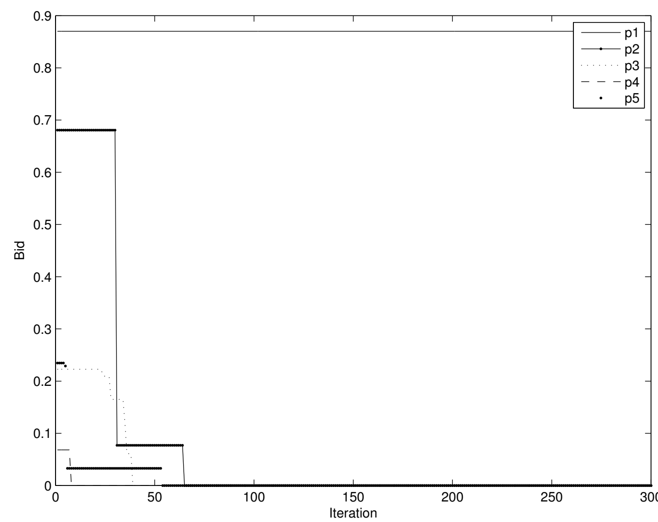

Figure 1: Evolution of the bids.

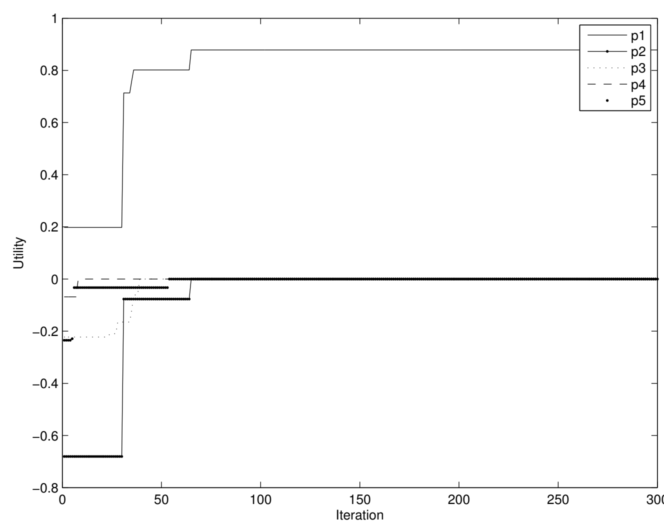

Figure 2: Evolution of the utility functions

$$u_{bi}(p_{bi}(k); k) = \begin{cases} \frac{1}{N(k)}(v_i - p'_b(k)) & \text{if } p_{bi}(k) = p_b(k) \\ -p_{bi}(k) & \text{otherwise} \end{cases} \tag{5}$$

where $p_b(k)$ is the highest bid, $N(k)$ is the number of buyers that bid the highest bid, and $p'_b(k)$ is the second highest bid, all in round k.

The other parameters in the algorithm were set to $loop = 300$ and $\sigma = 0.5$.

Figures 3 and 4 show the statistics of convergence when the values of the object for the five players are always the same and are equal to 0.8785, 0.7110, 0.6611, 0.4396, 0.5628. The his-

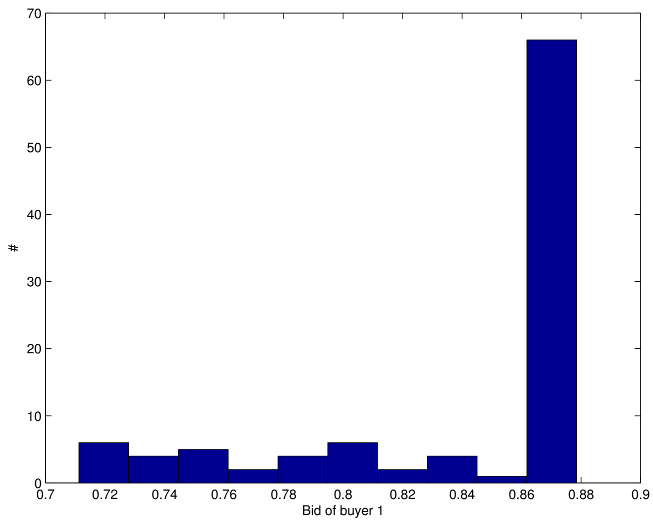

Figure 3: Empirical frequencies of convergence to different bids for the fist player (the buyer who gains the object of the auction).

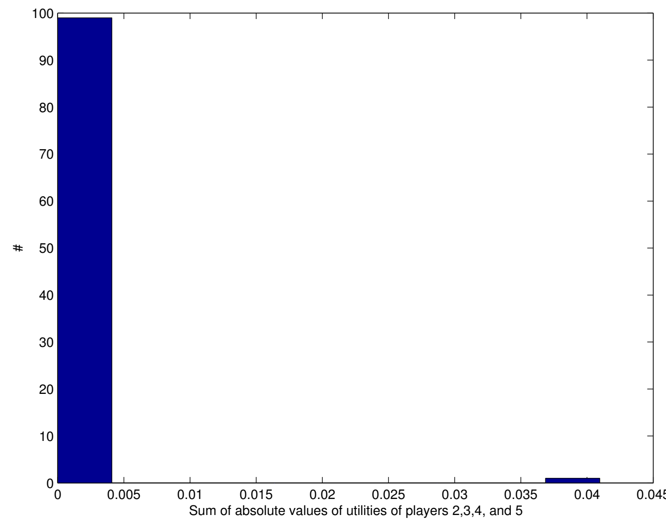

Figure 4: Empirical frequencies of the sums of absolute values of utility functions for the non-winning buyers (i=2,3,4) as an estimator of the error in reaching the equilibrium of the market (error reflecting inadequate learning in the game).

togram presented shows that different bid price equilibria can be achieved in the market depending on the particular course of the stochastic evolution of the game. As follows from the formula for the utility functions, all these prices are part of Nash equilibria when the remaining buyers bid zero. Figure 4 shows that perfect learning is not achieved as the sum of the absolute values of the utility functions for the non-winning buyers is non-zero in the region $[0, 0.04]$.

3.2 Double Auction Market

A double auction market is considered in which the number of buyers and sellers is the same and is equal to $n > 0$. It is further assumed that there is only one kind of good to trade and that in any round of the game a seller has a single unit of good to sell and a buyer can buy up to one unit of good. In any round of the game any buyer will be matched with a random seller. A transaction will take place, benefiting both the buyer and the seller, only if the price bid by the buyer exceeds the price asked by the seller.

If $c \in [0, 1]$ is the cost of the production and $v \in [0, 1]$ represents the value of good for the buyers, and under the assumption that a buyer and a seller will benefit from their transaction equivalently, the utility functions, u_{bi} and u_{sj}, of buyer i and the seller j in a single round of the game can be given by the formulae below [1], [2]:

$$u_{bi}(b_i, s_j) = \begin{cases} v - \frac{p_{bi}+p_{sj}}{2} & \text{if } p_{bi} \in [p_{sj}, v] \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$u_{sj}(b_i, s_j) = \begin{cases} \frac{p_{bi}+p_{sj}}{2} - c & \text{if } p_{sj} \in [c, p_{bi}] \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

in which p_{bi} and p_{sj} denote the prices of buyer i and seller j, respectively.

Another example of a double auction market will be also considered that is created by adopting a different set of utility functions:

$$u_{bi}(b_i, s_j) = \begin{cases} v - (\frac{p_{bi}+p_{sj}}{2})^2 & \text{if } p_{bi} \in [p_{sj}, v] \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$u_{sj}(b_i, s_j) = \begin{cases} (\frac{p_{bi}+p_{sj}}{2})^2 - c & \text{if } p_{sj} \in [c, p_{bi}] \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Proposition 1 *[1] A double auction game with employing any of the two sets of utility functions as above, with populations of buyers and sellers of equal cardinalities, is in equilibrium if all the players are bidding and offering the same price, i.e.* $p_{bi} = p_{sj} = \psi \in [c, v]$ *for all* $i, j \in \{1, ..., n\}$.

Justification :
Assume that all the players are biding/offering $\psi \in [c, v]$ i.e. $p_{bi} = p_{sj} = \psi, \forall i, j$. If bidder i decides to bid higher, $p_{bi} > \psi$, it is obvious that the utility function will decrease for that player. If the same bidder decides to bid lower, her utility function will be zero. A similar reasoning can be applied to a seller, implying that the market is in Nash equilibrium.

Algorithm Proposed For Double Auction Markets

The evolutionary iterative algorithm for learning in double auction markets developed here belongs to the general class of random search algorithms. The underlying idea of the algorithm is that buyers and sellers try to follow the most successful buyer or seller known to them from the previous iteration of the algorithm. This algorithm can hence be considered to belong to the class of algorithm of guided learning.

The following notation is adopted :

– n, the cardinality of the populations of buyers and sellers.
– k, the index of the current round of the game, ($k \in \mathbb{Z}$.)
– $p_{bi}(k) \in [c, v]$, the maximal price at which buyer i is willing to buy in round k.
– $p_{sj}(k) \in [c, v]$, the minimal price at which seller j is willing to sell in round k.
– $c \in [0, 1]$, the cost of production for sellers.
– $v \in [0, 1]$, the value of the product for buyers.
– $\bar{p}_b(k) \in [c, v]$, the average of the buyers' bid prices in round k.
– $\bar{p}_s(k) \in [c, v]$, the average of the sellers ask prices in round k.
– m, the number of buyers (or sellers) that any seller (or buyer) meets in any round of the game.
– $\alpha > 1$, a shrinking factor for the variance of the randomizer function used in the generation of the bid and offer prices.
– $\sigma_k > 0$, the variance of the random generator function in round k of the game.
– $\mu_{bk} \in [c, v]$ and $\mu_{sk} \in [c, v]$, the means for the random generator functions for buyers and sellers, respectively.
– $N_T(\mu, \sigma)$, the normal distribution with mean μ and variance σ truncated to the interval $[c, v]$, i.e. if $p \sim N(\mu, \sigma)$, the normal distribution, and $p \notin [c, v]$ then p is reset to c or v depending on whether the initial sample satisfies $p < c$ or else $p > v$.
– m_{count}, a counter by which a buyer meets exactly m sellers.
– $u'_{bi}(p_{bi}, p_{sj})$, variable that is used to show the value of the utility function for buyer i as she meets seller j.
– $u'_{sj}(p_{bi}, p_{sj})$, variable that is used to show the value of the utility function for seller j as she meets buyer i.
– $usum_{bi}$, variable that is used in averaging the utilities of buyers.
– $usum_{sj}$, variable that is used in averaging the utilities of sellers.
– cs_j, the counter of the number of times that seller j has a chance to participate in a transaction.
– $\epsilon \in (0, 1)$, algorithm termination threshold.
– i^* and j^*, the indices of the buyer and seller, respectively, who achieve the highest utility values in the current round of the game.

Before stating the steps of the algorithm it is helpful to explain the meaning behind them. The values of the algorithm parameters and the initial values of the buyers' and sellers' prices are selected in Steps 0 and 1. The latter are variables that are used in averaging the utility of every buyer and seller that participate in the market. Steps 3 - 6 constitute a loop in which each buyer meets m sellers in the current round of the algorithm. As a result of the meeting between buyer i and seller j, both of them claim utility values $u'_{bi}(p_{bi}, p_{sj})$, and $u'_{sj}(p_{bi}, p_{sj})$, that add up to: $usum_{bi}$ and $usum_{sj}$, respectively. The counter cs_j is upgraded to serve the averaging of utility values for every seller in Step 7. Buyers do not need a similar counter as there is always m values to average over for each buyer. Step 8, commences by determining the indices i^* and j^* of the buyer and seller, respectively, who achieve the highest utility values in the current round of the game. The prices of this buyer and seller are then selected as the averages μ_{bk} and μ_{sk} for the randomizer normal distribution employed to generate the prices for buyers and sellers in the next round of the game.

The variances of both probability distributions are shrunk by a factor $1/\alpha$ for the next round of the game. The variances of the randomizing distributions are decreasing as players learn about the market whose behavior is tightly related to the ensemble of their utility functions. The algorithm is exited if the prices of the buyers and sellers are sufficiently close to each other (close to the equilibrium of the game). Clearly, the information structure in this game is as follows: the players know their own utility functions, their own price and the current price of their opponents in the market game.

Algorithm 2 The Stochastic Optimizer Algorithm For Double Auction Market.

Step 0: Set the initial values of m, c, v, $\alpha > 1$, *and* $k = 0$. *Set initial values for* $\sigma_0 > 0$, $\mu_{b0} \in [c, v]$ *and* $\mu_{s0} \in [c, v]$. *For* $i, j = \{0, ..., n\}$ *draw samples of initial values of the ask and bid prices from uniform distributions over the interval* $[c, v]$, *i.e.*$p_{bi}(0) \sim U(c, v)$ *and* $p_{sj}(0) \sim U(c, v)$.

Step 1: For $i, j = \{1, ..., n\}$ *set* $usum_{bi} = 0$, $usum_{sj=0} = 0$, *and* $cs_j = 0$ *(parameters needed for averaging of utility values for all players).*

Step 2: Set $i = 1$, *indicating that the utility function is averaged for buyer* i. *Set* $m_{\text{count}} = 0$.

Step 3: Draw an integer $j^* \in \{1, 2, ..., n\}$ *from a uniform distribution (i.e.* $Pr(j^*) = 1/n$*) without repetitions in round* k. *Update the counter of the number of times that seller* j *participates in asking against all buyers:* $cs_j = cs_j + 1$.

Step 4: Calculate $u'_b(i, j)$ *and* $u'_s(i, j)$ *- the utility values for buyer* i *and seller* j *as they meet. Update the total sums:* $usum_{bi} = usum_{bi} + u'_{bi}(p_{bi}, p_{sj})$ *and* $usum_{sj} = usum_{sj} + u'_{sj}(p_{bi}, p_{sj})$.

Step 5: Increment counter $m_{\text{count}} = m_{\text{count}+1}$. If $m_{\text{count}} < m$, go to Step 3.

Step 6: $i = i + 1$, go to Step 3 if $i < n + 1$.

Step 7: For $i, j = \{1, ..., n\}$, set $u_{bi} = usum_{bi}/m$, and $u_{sj} = usum_{sj}/cs_j$, the average utilities of buyers and sellers.

Step 8: Update the price generator densities for the buyers and the sellers, as follows. First determine the indices i^* and j^* of the buyer and seller, respectively, who achieve the highest utility values in the current round of the game. Then set : $\mu_{bk} = p_{bi^*}$, and $\mu_{sk} = p_{sj^*}$.

Step 9: Evolve the price of each buyer and seller according to $p_{bi}(k+1) \sim N_T(\mu_{bk}, \sigma_k)$, $p_{sj}(k+1) \sim N_T(\mu_{sk}, \sigma_k)$; $i, j \in \{1, ..., n\}$.

Step 10: Contract the variance of the averages of the price generator densities for the buyers and the sellers: $\sigma_{k+1} = \sigma_k/\alpha$.

Step 11: Verify the algorithm's stopping condition. If $|\bar{p}_b(k) - \bar{p}_s(k)| > \epsilon$, then set $k = k + 1$, and go to Step 1, else exit the algorithm.

The evolution of the ask and bid prices of the players in the market during the first few rounds of the game are shown in figures 5, 6, and 7. It can be seen that the prices of both buyers and sellers are concentrating in the neighborhoods of their corresponding best bid or ask prices. In all these tests the parameters of the algorithm are set to $n = 100$, $\alpha = 1.1$, $\sigma_0 = 0.3$, $c = 0$, $v = 1$.

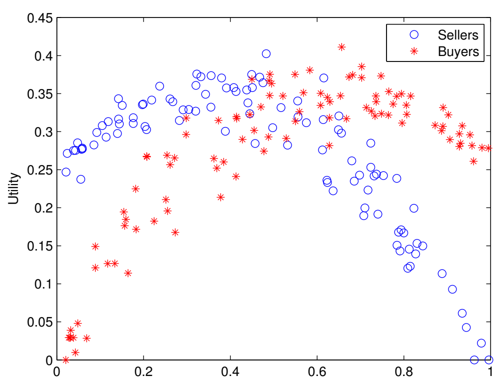

Figure 5: Diagram of prices, Step 0

The curves in Figures 8 to 11 represent the evolution of the average prices of the population of buyers and sellers during the game using the utility functions 6 to 9, respectively. It is seen that convergence to a Nash equilibrium of the game is achieved in each case. In figures 9, 10, 12, and 13, statistics of convergence are shown for the algorithm that terminates after 100 rounds of the auction game. It is seen that the spread between the average bid and ask prices $p_b - p_s$ is marginally small which essentially demonstrates convergence to a single market price.

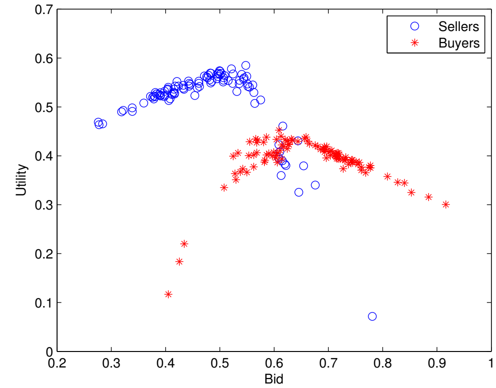

Figure 6: Diagram of prices, Step 1

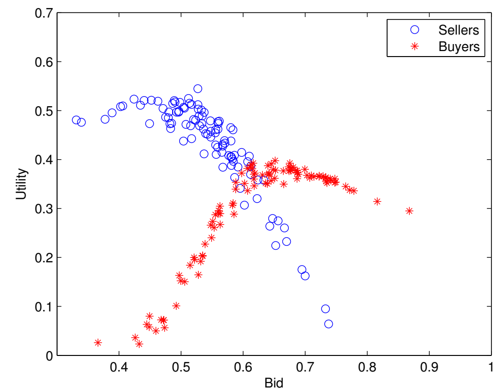

Figure 7: Diagram of prices, Step 2

3.3 Merit Based Matching

In real auction markets the buyers and sellers do not meet randomly, but the system selects the partners by their merits. An example of such a double auction market is considered by adopting a different set of utility functions:

$$u_{bi}(b_i, s_j) = \begin{cases} v - p_{bi} & \text{if } p_{bi} \in [p_{sj}, v] \\ 0 & \text{otherwise} \end{cases} \qquad (10)$$

$$u_{sj}(b_i, s_j) = \begin{cases} p_{sj} - c & \text{if } p_{sj} \in [c, p_{bi}] \\ 0 & \text{otherwise} \end{cases} \qquad (11)$$

The main idea here is that a buyer who bids a higher price re-

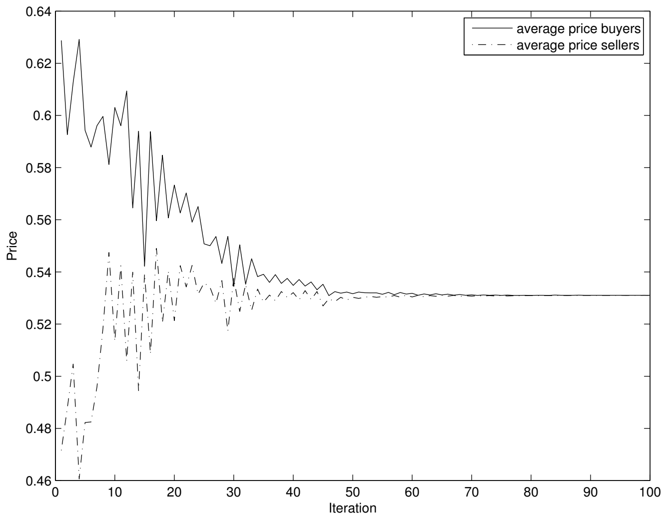

Figure 8: Convergence of the proposed algorithm, while using the utility functions of formulas (6) and (7).

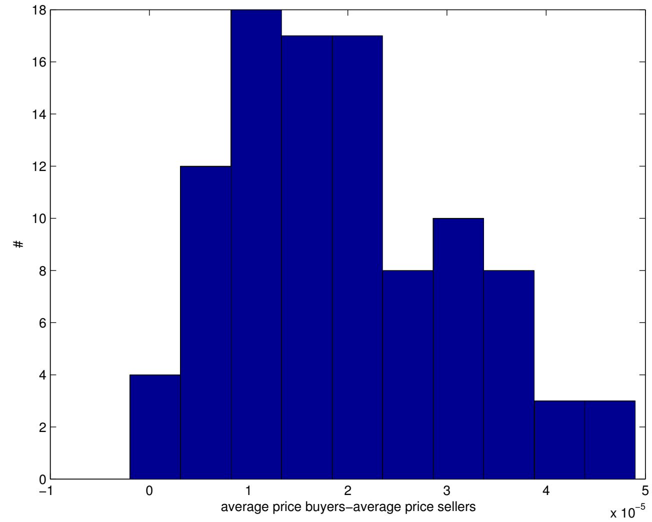

Figure 10: Empirical frequencies for the spread between the average bid and ask prices $p_b - p_s$, while using the utility functions of formulas (6) and (7).

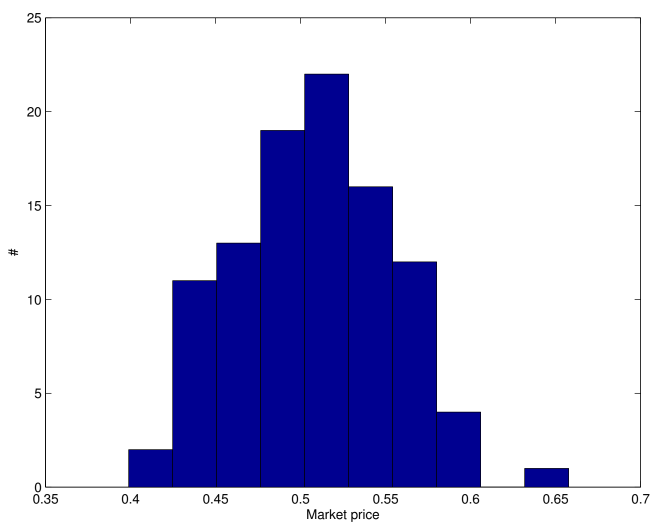

Figure 9: Empirical frequencies of convergence to different market equilibria, while using the utility functions of formulas (6) and (7).

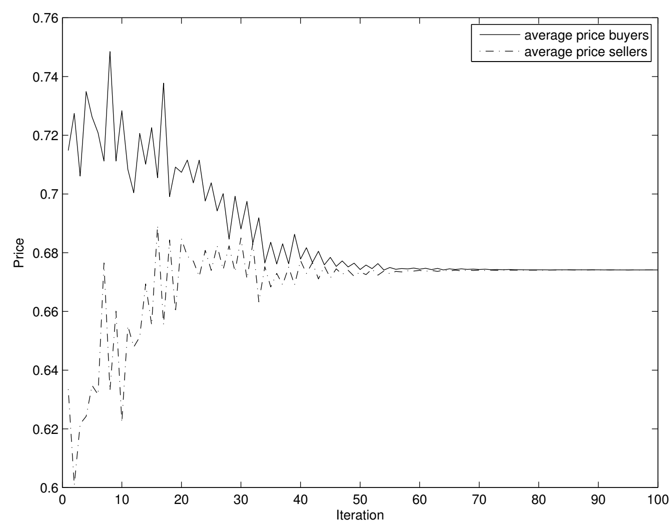

Figure 11: Convergence of the proposed algorithm, while using the utility functions of formulas (8) and (9).

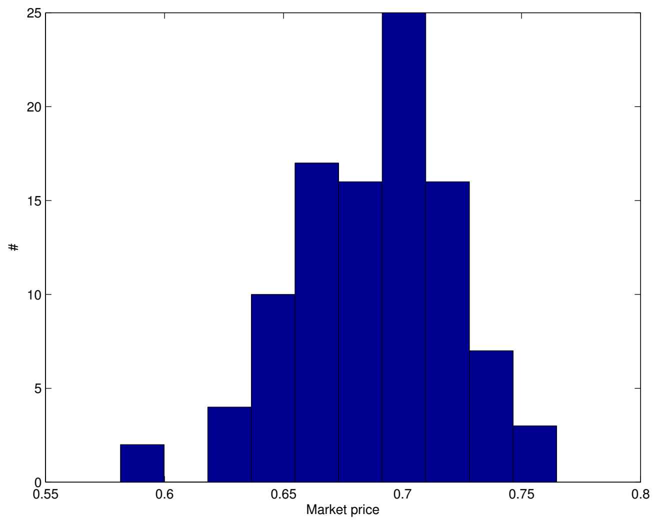

Figure 12: Empirical frequencies of convergence to different market equilibria, while using the utility functions of formulas (8) and (9)

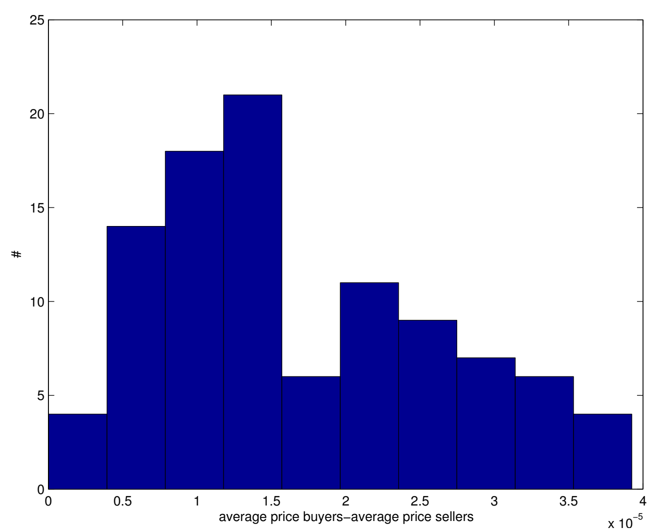

Figure 13: Empirical frequencies for the spread between the average bid and ask prices $p_b - p_s$, while using the utility functions of formulas (8) and (9)

ceives more merit points for transaction than a buyer who bids a lower price. Similarly, a seller who asks a lower price receives more merit points than the one who asks a higher price. The buyers and sellers are then matched according to their respective ranks of merit. Algorithm 3 is designed for learning in such a market. Unless otherwise stated, notation and definitions are the same as those used in Algorithm 2.

Algorithm 3 The Stochastic Optimizer Algorithm for Double Auction Market with Merit Based Matching.

Step 0: Set the initial values of c, v, $\alpha > 1$, *and* $k = 0$. *Set initial values for* $\sigma_0 > 0$, $\mu_{b0} \in [c, v]$ *and* $\mu_{s0} \in [c, v]$. *For* $i, j = \{1, ..., n\}$ *draw samples of initial values of the ask and bid prices from uniform distributions over the interval* $[c, v]$, *i.e.*$p_{bi}(0) \sim U(c, v)$ *and* $p_{sj}(0) \sim U(c, v)$.

Step 1: Set $i = 1$, *indicating that the utility function is calculated for buyer* i.

Step 2: Calculate r, *the merit rank of buyer* i *in the population of all buyers (buyer with the highest bid has rank 1). Find* j, *the seller that has rank* r *in the population of all sellers (seller with the lowest ask has rank 1).*

Step 3: Use utility functions of formulae 10, 11 to calculate u_{bi}, *and* u_{sj}, *utility function values for the buyer and the seller in Step 2.*

Step 4: $i = i + 1$, *go to Step 2 if* $i < n + 1$.

Step 5: Update the price generator densities for the buyers and the sellers, as follows. First determine the indices i^* *and* j^* *of the buyer and seller, respectively, who achieve the highest utility function values in the current round of the game. Then set :* $\mu_{bk} = p_{bi^*}$, *and* $\mu_{sk} = p_{sj^*}$.

Step 6: Evolve the price of each buyer and seller according to $p_{bi}(k+1) \sim N_T(\mu_{bk}, \sigma_k)$, $p_{sj}(k+1) \sim N_T(\mu_{sk}, \sigma_k)$; $i, j \in \{1, ..., n\}$.

Step 7: Contract the variance of the averages of the price generator densities for the buyers and the sellers: $\sigma_{k+1} = \sigma_k/\alpha$.

Step 8: Verify the algorithm's stopping condition. If $|\bar{p}_b(k) - \bar{p}_s(k)| > \epsilon$, *then set* $k = k + 1$, *and go to Step 1, else exit the algorithm.*

Figure 14 presents the evolution of the average prices of the populations of buyers and sellers to a Nash equilibrium of the game. All parameters of the algorithm are the same as those used to test

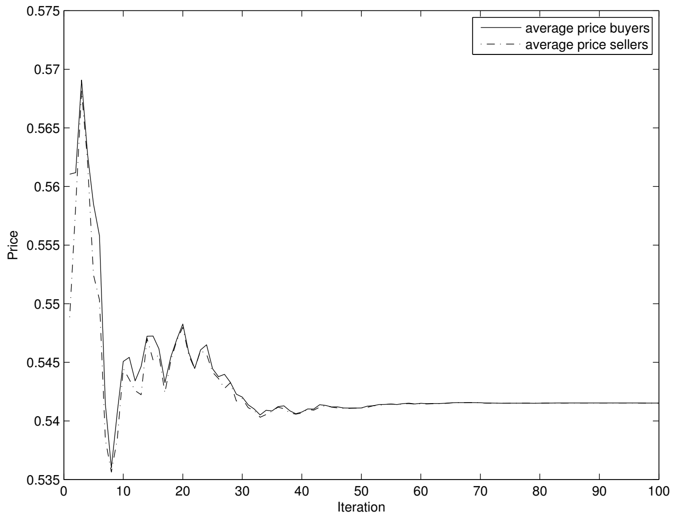

Figure 14: Convergence of the proposed algorithm, while using the utility functions of formulas (10) and (11)

Algorithm 2 in Subsection B. In figures 15, and 16 statistics of convergence are shown for the algorithm that terminates after 100 rounds of the auction game. It is seen that the spread between the average bid and ask prices $p_b - p_s$ is marginally small which essentially demonstrates convergence to a single market price.

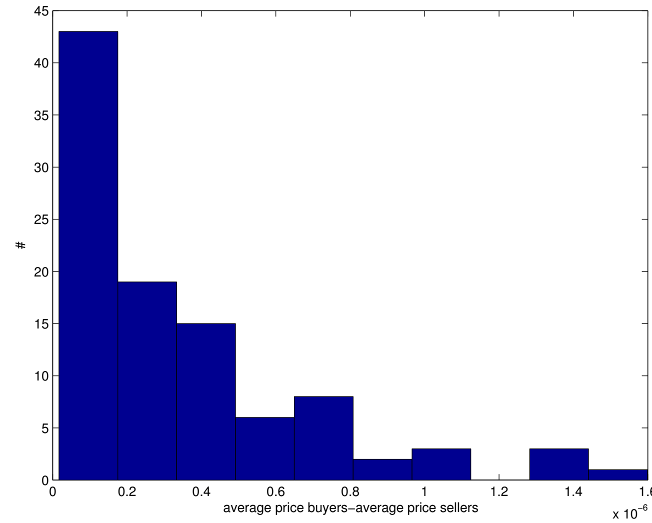

Figure 15: Empirical frequencies for the spread between the average bid and ask prices $p_b - p_s$, while using the utility functions of formulas(10) and (11)

4 Conclusions

Stochastic programming evolutionary algorithms are proposed in this paper and are applied to different cases of auctions such as a second-price sealed-bid auction and double auction markets. These algorithms do not require any a priori assumptions about the stationary behavior of bidders or sellers to be made. Also, no specific assumptions are made about the probability distribution functions as believed price behavior of the players. The utility functions are allowed to be discontinuous as is the usual case in real life auction markets. Thus, the presented algorithms are considered to compare favorably against the competitive classes of algorithms employed for learning in market games such as the "fictitious play" algorithms (which make assumptions about the stationary behavior of the players) and the "gradient play" algorithms (which cannot handle discontinuous utility functions).

Numerous simulations confirm that the new algorithms converge to equilibria of the market.

In the last part of this paper, a more realistic matching process is considered in the auctioning mechanism as the merit of buyers and sellers plays a central role in the matching process.

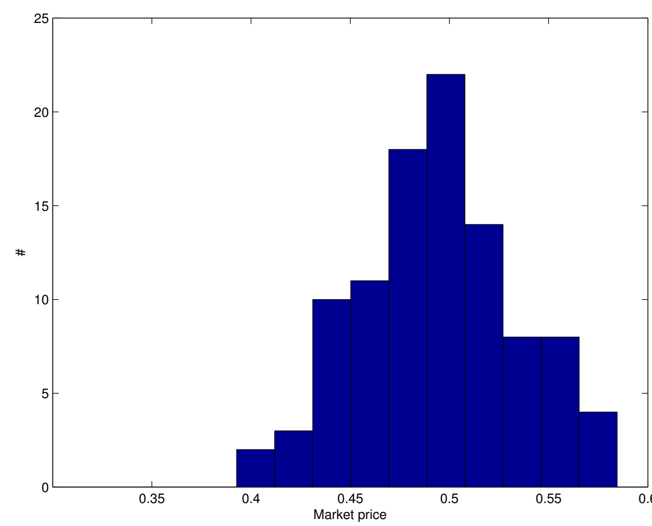

Figure 16: Empirical frequencies of convergence to different market equilibria, while using the utility functions of formulas (10) and (11)

The novel algorithms can be applied to other cases of auctions and discontinuous games. Further research should address the dependence of the values of the equilibria on initial market conditions and parameters of the algorithms.

References

[1] H. Dawid, "On the Convergence of genetic learning in a double auction market," *Journal of Economic Dynamics and Control*, vol. 23, pp. 1545-1567, 1999.

[2] H. Dawid, *Adaptive Learning by Genetic Algorithms: Analytical Results and Applications to Economic Models*, Springer-Verlag New York, LLC, 1999.

[3] M. O. Jackson, L.K. Simon, J.M. Swinkels, W.R. Zame, "Communication and Equilibrium in Discontinuous Games of Incomplete Information," *Econometrica*, vol. 70, pp. 1711-1740, 2002.

[4] D. Fudenberg, M. Mobius, and A. Szeid, "Existence of equilibrium in large double auction," *Journal of Economic Theory*, vol. 133, pp. 550-567, 2007.

[5] D. Fudenberg, *The Theory of Learning Games,* MIT Press, 1998.

[6] O. V. Guseva, "Convergence of a random-search algorithm," *Cybernetics and Systems Analysis*, vol 7, pp. 1104-1106, 1971.

[7] P. Herald, *Evolutionary Games And Equilibrium Selection,* MIT Press, 1997.

[8] M. O. Jackson, and J. M. Swinkels, "Existence of Equilibrium in Auctions and Discontinuous Bayesian Games: Endogenous and Incentive Compatibility Sharing Rules," *Econometrica,* vol. 73, pp. 93-140, 2005.

[9] J. H. Kagel, "Cross-game learning: Experimental evidence from first-price and English common value auctions," *Economics Letters,* vol. 49, pp. 163-170, 1995.

[10] D. C. Karnopp, "Random Search Techniques for Optimization Problems," *Automatica,* vol. 1, pp. 111-121, 1963.

[11] A. R. Kian, J. B. Cruz, Jr., and R. J. Thomas, "Bidding strategies in oligopolistic dynamic electricity double-sided auctions," *Power Systems, IEEE Transactions on,* vol. 20, pp. 50-58, 2005.

[12] P. J. Reny, "On the Existence of Pure and Mixed Strategy Nash Equilibria in Discontinuous Games," *Econometrica,* vol. 67, pp. 1029-1056, 1999.

[13] J. S. Shamma and G. Arslan, "Dynamic fictitious play, dynamic gradient play, and distributed convergence to Nash equilibria," *IEEE Transactions on Automatic Control*, vol. 50, pp. 312-327, 2005.

[14] J. I. Singer, "Double Auctions across a Constrained Transmission Line," *Operations Research,* vol. 50, pp. 449-461, 2002.

[15] S. Hon-Snir- and D. Monderer, "A Learning Approach to Auctions," *journal of economic theory,* vol. 82, 65-88, 1998.

[16] J. W. Weibull, *Evolutionary Game Theory,* MIT Press, 1995.