# Deep learning approach to pricing of food processors in agricultural spatial markets

# Abstract

Pricing behavior of agricultural processing firms in agricultural input markets has large impacts on the farmers and processors prosperity as well as the overall structure of the market. Despite existing analytical contributions with regard to explanation of food processors’ pricing policies in agricultural markets, the need for models, which can reflect real complex market environment features is contemporary. Agent-based models ABMs serve by now as computational laboratories to help understand market outcomes emerging from autonomously interacting agents. Yet, individual agents within ABMs must be equipped with appropriate intelligent learning algorithms. The objective of this paper is to development robust deep learning agents to simulate the pricing behavior of agricultural processing firms. Our simulations provide rigorous insights to improve the explanation of agricultural processors’ spatial pricing policies in agricultural input markets.

# Introduction

Although microeconomic textbooks often introduce agricultural markets as examples for perfectly competitive markets, a large number of studies have shown that such markets are of imperfect-competitive nature (Sexton, 1990 and 2012). In light of dramatically increased concentration in food processing, large number of spatially dispersed farms supply primary input to a small number of large processing firms. Spatial pricing of processing firms in input markets -competing on farmers’ products- is investigated in the existing literature, often based on the models of oligopsonistic competition (Rogers and Sexton, 1994; Durham et al., 1996; Alvarez et al., 2000; Huck et al., 2006; Graubner et al., 2011a). The food processors’ spatial competition has short to long term impacts on the emerging prices in procurement markets. This determines the farmers and processors prosperity as well as the overall structure of the market. This paper’s objective is to discover computational solutions by means of artificial intelligent methods to improve the explanation of agricultural processors’ spatial pricing policies in agricultural input markets.

Most of the existing theoretical approaches in the literature (e.g. Espinoza, 1992; Kats and Thisse 1993; Zhang and Sexton, 2001; Fousekis, 2011) utilize *analytical* methods to understand the pricing policies of processing firms. Although these approaches develop solid mathematical models to explain the emerging prices, they are not suited to describe pricing policies in the real complex market environments. Whereas these models often

* assume one dimensional (linear) spatial markets, the real life procurement markets are two-dimensional markets.
* presume only two extreme pricing policies, namely Free On Board(FOB) pricing (where the processors set the firm-gate price and farms must pay the entire transportation cost from farm gate to the processing company gate) and Uniform Delivered (UD) pricing (where the processors set the farm-gate price and bear the whole transportation cost), in the real life markets, the processing firms are free to choose prices with various possible degrees of absorbing transport costs comprising not only the FOB to UD but also in-between degrees of shared transport costs to be absorbed by both the purchasing firms and the farmers.
* Assume that interaction of firms takes places in one stage games, the real life markets can incorporate infinitely dynamic firms’ interactions.

Agent based models ABMs (Tesfatsion, 2006) are recently proposed approaches to cope with the deficiencies mentioned above. These frameworks are utilized to facilitate the simulation of interaction of autonomous agents in complex environments (Grimm and Railsback, 2005). Yet, individual agents within ABMs must be equipped with appropriate decision making mechanisms in order to enable the entire model to successfully simulate emergent behavior at the system level (Kirman, 2011). Thus, agents must discover a solution autonomously through observations and relying on their own knowledge by means of *learning* (Weiss, 2000).

While learning algorithms in the context of multi agent systems (MASs) are predominantly applicable to just small problems (Busoniu et al., 2010) e.g. small stochastic games with 2 players and 2 possible actions per each player and or small grid worlds, modelling complex economic systems i.e. agricultural input markets in ABMs contexts requires learning methods, which are able to overcome the adaptation problem with rich decision spaces (e.g. with firms, which are free to choose hundred’s possibilities of pricing policies and live on non-trivial grid worlds). In the field of MASs there is continuous research towards the development of robust agents for large-scale and dynamic environments often by incorporating individual based reinforcement learning (Sutton and Barto, 2005) methods (Busoniu et al., 2010). In the field of computational economics, researchers often prefer to analyze the firms’ behavior einther by means of evolutionary (Mitchell, 1996) methods e.g. the studies of Graubner et al. (2011a, 2011b and 2022) or on the basis of the individual based reinforcement learning RL (Sutton and Barto, 2005) methods e.g. the study of Nicolaisen et al. (2001) for the case of whole sale electricity market. In this paper we opt for using reinforcement learning methods, which are based on deep neural networks (DNNs). Choosing DNNs in our study is on one hand because interpreting the dynamics of genetic algorithms as individual learning processes are not in all cases clear (Brenner, 2005). On the other hand, DNNs are powerful computational approaches that can process unprecedented amounts of data (Bishop, 2006; Krizhevsky et al., 2012). Hence, we make use of the DNNs potentials to develop reinforce learning pricing agents in agricultural input markets. In addition, we combine the application of RL with DNNs. The joint application of RL and DNNs is a solution first presented by Mnih, et al. (2015) to overcome the mentioned course of dimensionality issue in market interactions with rich decision spaces.

To examine the results of our study we constitute two types of agents: agents who we name *supervised agents* throughout and agents who we name *unsupervised agents*. S*upervised agents* only confront *supervised agents* through our study and the *unsupervised agents* only confront *unsupervised agents.* The *unsupervised agents* are deep reinforcement learning agents in line with the aforementioned logic of Mnih, et al. (2015). The *supervised agents* are more straightforward ones and act in line with a learning theory analogous to the *sequential move adjustment* process in Fudenberg and Levine (1998) and Maskin and Tirole (2001). These agents are programmed to know exactly the most profitable pricing policy from all possible range of their pricing policies given the pricing policy of the opponent. Indeed, in contrast to the *unsupervised agents*, who are involved in a course of interaction with the competitors, without any pre knowledge, the *supervised agents* are fed with a complete list of *best response* policies given the price of the opponent, which is computationally prepared before their pricing interaction begins. As we expect that the sequential process of mutually *best responding* will lead the system towards the same or close termination points as the *Nash equilibrium* (Fudenberg and Levine, 1998), the *supervised agents* are constituted as the benchmark in our study to use the outcomes of their interactions as reliable criteria to evaluate the performance of the *unsupervised agents* with respect to achieving (or not achieving) the targeted games Nash equilibria.

The remainder of the paper is organized as follows. Section 2 presents the literature background with regard to pricing theories in agricultural markets as well as learning theories within multi agent systems. Section 3 presents the spatial market environment of this study together with the processors’ and farms’ attributes. Section 4 conveys the algorithms, which the supervised agents and unsupervised agents use, respectively. The results of the simulations of supervised and unsupervised agents, each within the elastic as well as the non-elastic market environments are presented in section 5. Section 6 conveys conclusions and further thoughts.

# Background

There are three well-known pricing policies practiced by the processors in agricultural input markets: free on board pricing FOB, uniform delivered pricing UD, and optimal discriminatory OD (where the freight costs are shared equally between the processor and the farmer) pricing (Beckman, 1976). Until the early 1990s, the researchers’ views on the choice of pricing policies by processors and their market implications are not fundamentally relying on mathematical models (Scherer, 1980; Greenhut, 1981; Greenhut et al. 1987) but showed the more frequent practices of UD and OD pricing relative to FOB pricing. Espinoza (1992) and Kats and Thisse (1993) are the first well-known studies analytically modelling the spatial pricing of firms. Both studies suggest that UD pricing policies are likely to be observed in highly non-competitive industries (industries where the transportation cost is high) but also in highly competitive industries (industries with low transportation cost), while FOB is likely for intermediate market settings. Zhang and Sexton (2001) highlight that the presumed farmers’ output supply elasticity of price in the models of Espinoza (1992) and Kats and Thisse (1993) is assumed to be perfectly inelastic leading to bias the processors’ policies in favor of UD pricing. Using a supply function with strictly positive (unitary) price elasticity for farms, Zhang and Sexton (2001) suggest that contrary to the Espinoza’s model, the FOB pricing policies emerge as equilibrium under very competitive market settings. Mixed FOB–UD policies (e.g. one of processors chooses the FOB and the other the UD policy) are Nash equilibria in less competitive markets and UD pricing emerge when shipping costs are high relative to the value of the finished product, for example for markets that are nearly monopsonistic in nature. Fousekis (2011) adopts Zhang and Sexton’s model to specific processor objectives and shows that the coexistence of processors with different objective functions (e.g. one of processors is profit maximizers and the other aims at simultaneously maximizing its profit and the farmers’ profit) are likely to give rise to some mixed pricing policies. According to Fousekis (2011) UD (FOB) pricing is chosen in markets where transportation costs are small (large) relative to the net value of the primary product. A mixed FOB–UD pricing equilibrium emerges for an intermediate market setting.

Although these models present strong analytical basics, they fail to reflect complex environment features like dynamic interactions, two-dimensional market shapes, asymmetric pricing policies of firms or pricing policies with various degrees of freight absorptions. Agent-based models (ABMs) serve by now as computational laboratories as bottom-up approaches to help understand market outcomes emerging from autonomously deciding and interacting agents. ABMs employ various learning methods. However, individual agents must be equipped with appropriate adaptive decision mechanisms to successfully simulate such emergent behavior at the system level (Kirman, 2011). In spatial competition context, each processor agent needs to dynamically keep up with the changes in the prices of other agents. According to Panait and Luke (2005) there are three main approaches to learning: supervised, unsupervised and reward based learning. In supervised learning, an agent deals with the problem of learning the optimal function mapping inputs to outputs by training with a series of input and output pairs. A teacher or supervisor steers the learning progress through providing feedback on the success. Deep Neural Networks DNNs (Bishop, 2006; Krizhevsky et al., 2012) are typically examples of supervised learning. In unsupervised learning, no feedback is provided. Data mining methods, clustering and discovery are examples of unsupervised learning. The reward-based learning methods are divided into two subsets: reinforcement learning (RL) and stochastic search methods such as evolutionary algorithms e.g. genetic algorithm (GA).

The study of market power in a broad range of studies in computational economics is often done using genetic algorithms (Vallée and Basar, 1999; Alemdar and Sirakaya, 2003; Arifovic, 1994; Vriend, 2000; Graubner et al. 2011). Agents using a genetic algorithm require less prior competence in the specific task (Arifovic, 1994). Such evolutionary algorithms can be quite useful for some classes of complex problems especially when the problem is non-trivial to deal with. Graubner et al. (2011a), Graubner et al. (2011b) are among the first studies who apply ABMs with incorporation of *genetic algorithms* to investigate the spatial pricing policy of firms in agricultural procurement markets. They find that UD pricing is an equilibrium behavior under high and mild spatial competition between processors and OD emerges under less intense competition. In contrast, FOB pricing does not emerge in equilibrium. Graubner and Sexton (2022) investigate the joint selection of location and price policy by processing firms and find that, when buyers have the flexibility to jointly choose their locations and pricing policies, farm product procurement markets are both more competitive and more efficient than has been predicted by prior studies. They especially come to the result that pricing policies close to FOB pricing *re-emerge* as equilibrium strategies in the market settings where competition is intense. The group of representative in the context of the firms’ spatial pricing are depicted in table 1.

Table 1: Firms’ pricing policy in agricultural input market from different agricultural economics studies’ point of view

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Work by | Pricing Game | Supply elasticity | Specific firm character | Equilibrium by low competition | Equilibrium by medium competition | Equilibrium by high competition |
| Espinosa, 1992 | Repeated  Game | Constant =0 | No | UD | FOB | UD |
| Zhang and Sexton, 2001 | Static | Constant=1 | No | UD | FOB-UD | FOB |
| Fousekis, 2011 | Static | Constant=1 | IOF or COOP | FOB | FOB-UD | UD |
| Graubner et al., 2011 | Repeated  Game | Variable | No | OD | UD | UD |
| Graubner et al., 2022 | Repeated  Location & Pricing Game | Variable | No | OD | UD | Close to FOB |

From a critical point of view, despite the prevalence of GAs in the literature, interpreting the dynamics of genetic algorithms as *individual* learning processes are not in all cases clear (Brenner, 2005). In general, understanding the dynamics of evolutionary algorithms is complicated by the fact that the internal fitness measures (, which is the main component of the GA and measures e.g. how well a particular pricing policy yields a higher pay-off for a firm) for evaluation of policies are *subjective* (Luke and Wiegand, 2002 and Watson and Pollack, 2001). Vriend (2000) and Riechmann (2006) show how the learning dynamics of agents through defining fitness functions in evolutionary algorithms can substantially influence the outcome of the game. For example, if we assume that the vital criterion for measuring the fitness of policies are relative payoffs compared to a competitor, in this case it pays for an agent to hurt himself in terms of absolute payoff as long as he hurts its opponent even more. This kind of behavior is a result of the implied algorithm dynamics by defining the fitness functions, not of the game itself. In Graubner et al. (2011a), Graubner et al. (2011b) and Graubner and Sexton (2022) the main criterion for measuring the fitness of pricing policies is thought to select the pricing policies, which yield the highest average pay-offs when a processing firm is repeatedly opposed with different combinations of the competitor's pricing policies. While choosing this fitness function is logically expected to result in selection of *on average* solid pricing policies by each firm, it is not clear whether the concluded equilibria policies in can be interpreted as pure or mixed Nash equilibria in all cases.

Instead of elaborating on GAs, in this paper we opt for using a reward based method i.e. reinforcement learning. Thereby, we make use of the deep neural networks DNNs to develop RL pricing agents in agricultural input markets. Processors use basically a well-known RL algorithm i.e. Q-learning algorithm (Watkins1992). Q-learning agents elaborate decisions based on the notion of *dynamic programming* (Bellman, 1957) to solve optimization problems by combining solutions to sub-problems. Each agent solves each sub-problem just once and saves its answer in a memory table (named Q-table), to avoid the re-computation. The entries of tables are called *Q-values*. The evaluations of Q-values are then iteratively reinforced and improved, more and more the game is played. However, by increasing the interaction domain (e.g. in games with rich decision spaces), tabular storage of Q-values for agents becomes economically infeasible i.e. impractical. Hence, as the basic Q-learning algorithm needs huge values of storage tables capacities, it is predominantly applicable to just problems with small decision spaces (Busoniu et al., 2010). Mnih, et al. (2015) are the first, who introduced the jointly usage of Q-learning together with utilizing deep artificial neural networks DNNs to overcome the issue of memory storage capacity. Hereby, the deep learning model takes the role of storage tables and can predict the Q-values without usage of any tables.

To examine the results of deep reinforcement learning agents in our study, we simulate two types of agents in separate games. *Unsupervised agents* are deep reinforcement learning agents, which act in line with Mnih, et al. (2015). *Supervised agents* are alternative *agents* who act in line with a learning theory analogous to the *sequential move adjustment* process in Fudenberg and Levine (1998) and Maskin and Tirole (2001). The *Supervised agents* exactly know how to *best response* their competitors. Hence, their behavior at the equilibrium serves as a benchmark in our study. We simulate games by means of the both defined agents’ types in a market environment presented in the next section and use the outcome of the supervised agents’ type’s games as equilibrium criteria to assess the performance of the unsupervised agents’ type.

# Market environment

In this section we describe the spatial environment of our study. We presume 2 price setting processors (as purchaser) located on a one-dimensional space. The region is discrete in space consisting of a grid of cells and each cell occupied by exactly one farm (as supplier) and locations are accessible by X-Y coordinates such that X = {-100, …, 1, 0, 1, …, 100}and Y=0. The location of processors in each simulation is fixed on the points (-100,0) and (+100,0). To normalize the factor distance, each distance between two next to each other points of the grid world is normalized by dividing by 100, so that each farm’s distance to its direct neighbor is equal to 0.01 and the fixed processors’ distances to each other is equal to 2.

Processors are price setting profit maximizing processors. A general price equation is assumed to describe the net price per unit quantity of supply (local price) received by farmers at each location:

(1)

The vector ( is in our study the representative of a processor’s pricing policy describing the price for a farmer at processor’s location by the term and the share of transportation cost absorbed by each farm due to spatial differences of agents expressed by the term . is the local price of processor each supplier farm receives at its location point *s*, and is the distance between processor *p* and the supplier farm *s,* and *t* describes a global variable for transportation cost rate. We limit the maximum possible values for product price of processors in the downstream market via normalization equal to 1. The price policy parameters of agents and have discrete values between 0 and 1 with predetermined increments equal to 0.01. We assume that suppliers are price takers and aim for the highest local price offered by processors. The cost function of supplier farms is

(2)

where is the production cost of producing amount of raw product and *ε* is the price elasticity of supply. Following (2) each farm will produce the amount which maximizes its profit function :

(3)

(4)

In the case of supply elasticity equal to zero, the production cost and the production quantity of each farm are set to 0 and 1 respectively. Note that the local prices received by each farm must be positive. However, the processors in our study are free to purchase the raw product even if it does not yield a positive local profit for them. Hence the set of potential suppliers for each processor is not limited within the space by any marginal location.

After submitting the processors’ bids to potential suppliers, each processor will earn the local profit knowing its ultimate supplier calculated as

(5)

Ultimately, each processor’s profit in our model is the sum of all local profits of its contracted suppliers. If two processors submit equal local bids to a farm, the supply of that farm is shared between the contracted processors (market overlap).

# Learning model

In this section, we convey the description of the two types of agents in our study i.e. the unsupervised agents and the supervised agents.

## 4.1 The unsupervised agents

The unsupervised agents in our study our deep Q-learning processor agents. The theory of Markov Decision Processes (MDPs) offers a framework for modeling the decision-making procedure by agents in the context of Q-learning. Q-learning uses MDPs for world representation. A MDP (Howard, 1960) is a tuple *(S, A, T, R)*, where S is the set of states, *A* is the set of actions, *T* is a transition function *[0 1]*, and *R* is a reward function *S×AR*. The transition function defines a probability distribution over the next states as a function of the current state and the agent’s action. The reward function defines the reward the agent receives when selecting an action at given state. Solving MDPs consists of finding a policy function , , which maps states to actions. An optimal policy maximizes the sum of future rewards *r*, discounted by factor , over time *t*. The optimal way for agents to learn the optimal policy is learning the optimal *value function* (Sutton and Barto, 2005). A *Q-function* is defined as the expected discounted reward given the agent takes a certain action *in state following policy*.

(6)

The optimal Q-function is defined as . It satisfies the Bellman optimality equation:

(7)

Equation (7) states that the optimal value of taking *a* in *s* is the expected immediate reward from undertaking *a* plus the expected discounted maximum value attainable from the next state . Once values corresponded to actions in each state are available, the optimal policy will be returned in every state by reinforcing the action with the largest optimal q-value.

(8)

The optimal policy of agent in each state would be typically assigning probabilities to actions that obtain higher Q-values. A broad range of single and multi-agent RL algorithms are derived from the basic Q-learning developed by Watkins (1992). A Q-learning agent maintains the value of each possible action in every state of the environment. These are called *Q-values* and are stored in a table. The evaluations of the quality of particular actions at particular states are iteratively improved. The agent, subject to some *error*, selects the most favorable action (the action, that gives already the maximum Q-value in his current state) *a* in its current state *s*. For example, in a so called *epsilon greedy* policy (which is used in our simulations) an agent chooses a random action (error) with a small probability *epsilon* and with a probability equal to *1- epsilon* decides to take the action, which gives already the maximum Q-value in his current state. This parameter is set to 0.2 in our study. Then it perceives the consequence of this action in form of the new state of the environment and its reward *r*. Through this reward, the agent validates the significance of its last action and updates its Q-value. Hence, Q-learning turns into an iterative approximation procedure. The agent starts with an arbitrary Q-function, observes transitions , and after each transition updates the Q-function according to

(9)

The term within the bracket at the right hand side of the equation 9 is the difference between the current estimate of Q-value of and the updated estimate of . Parameter setting influences the quality of learning. For example setting factor to 0 means that the Q-values are never updated, hence nothing is learned. Setting a high value such as 0.9 means that learning can occur quickly. This parameter is set to 1.0 in our study. The discount factor γ describes how an agent will evaluate the rewards, which he gets afterwards. If the discount factor meets or exceeds 1, the q-values may diverge. This parameter is set to 0.5 in our study.

The classical Q-learning algorithms are predominantly applicable to small problems only, e.g. games with 2 players and 2 possible actions per each player. By increasing the interaction domain tabular storage of q-functions for agents becomes economically infeasible i.e. impractical. The number of actions and states in a real-life environment can be thousands of thousands, making it extremely inefficient to manage Q-values in a table. Recent advance in deep neural networks DNNs especially in deep learning has enabled the application of Q-learning algorithms to large-scale decision problems (Silver et al. , 2016; Mnih, et al., 2015). In this case one can use DNNs to predict Q-values for actions in a given state instead of using tables. As an alternative for initializing and updating a Q-tables in the Q-learning process, we’ll initialize and train a neural network model to predict Q-values. DNNs consist of artificial neurons that receive and process input data. Data is passed through the input layer, the hidden layer, and the output layer to predict complex patterns (LeCun, Bengio, & Hinton, 2015; Schmidhuber, 2015). The layers of the neural network used in our study comprise the following. The input layer consists of 4 nodes comprising a tuple of and of both players, which represents the state of the world in our study. There are 3 hidden layers in the neural network architecture of our study each of them consisting of 50 neurons respectively. The output layer consists of 5 nodes comprising the number of actions, which each processor can undertake by observing the state of the world. Each of the 5 actions enables processors to decide upon the change in each element of their pricing vectors ( with the preferred increment sizes = 0.01 in line with the following gradients: [(-,0) , (0,-), (0,0), (0,), (,0)].

The DNN minimizes the error function (called loss function) presented in equation 10 through the course of learning i.e. the square value of the difference between the predicted and the target q values ( :

(10)

The unsupervised agents’ (deep Q-learning processor agents) learning algorithm is shown in the following in pseudo code format:

1. Initialize a DNN model
2. Initialize a list for memorizing (state of the world, action, new state of the world, reward) in each step of the game
3. In each step of the game:
   1. Observe the state of the world comprising all processor firms’ prices
   2. Demand the DNN model to predict the Q-value of each action from the state of the world
   3. If you are not in error mode:
      1. Choose the action with the highest Q-value
   4. Else:
      1. Choose a random action
   5. Adjust the pricing policy based on the chosen action
   6. Participate in the spatial competition by applying the determined pricing policy
   7. Each farmer decides whether to connect and deliver to which Processor based on the processors’ determined pricing policies
   8. Collect the input product from the connected farmers based on the pricing policy
   9. Pay the transportation cost according to distance to each farmer
   10. Process the input product and sell the processed product in the Down-stream market
   11. Calculate the final pay-off
   12. Set the final pay-off as reward
   13. Observe the new state of the world comprising all processor firms’ prices
   14. Extend memory based on new information: (state of the world, action, new state of the world, reward)
   15. For states of the worlds in the memory list:
       1. Demand the DNN model to predict the Q-value of each action from the new state of the world
       2. Set the highest Q-value among actions the Max\_New\_state\_Q\_Value
       3. Compute the Q-value of the chosen action from the state of the world according to equation: reward + discount\_factor \* Max\_New\_state\_Q\_Value
   16. Train the DNN model (1 epoch) by using states of the worlds as input and computed Q-values of each action from the state of the world as output

The training code to replicate the training process is appended to the supplementary material of this paper.

## 4.1 The supervised agents

In order to examine the performance of the unsupervised agents introduced in the beforehand section, we use a game theoretic approach analogous to the sequential move adjustment process in Maskin and Tirole (2001) and Fudenberg and Levine, (1998). We presume, when processors decide to become involved in spatial competition, they start pricing from an arbitrary price point but proceed to sequentially applying best response priced to each other. The processors mutually best responding is expected to lead us to the same termination points as the Nash equilibrium of the game implies (Fudenberg and Levine, 1998). Here, we initiate a course of best response sequential play between agents from the point ( and let the processors undertake moves subject to the assumption, that both agents know exactly to choose the most profitable pricing policy from all 10e+2\*10e+2 possible combinations of their ( given the pricing policy of the opponent. A complete list of best response policies given the price of the opponent in the market environment presented in section 3 is computationally prepared in tables and is fed into the agents by us as the supervisor before the sequential move game begins. The supervised agents’ (Best-response knowing processor agents) learning algorithm is shown in the following in pseudo code format:

1. In each step of the game:
   1. Observe the state of the world comprising all processor firms’ prices
   2. Select the Best response pricing policy given the opponent’s prices based on the information provided by supervisor
   3. If the same state of the world is twice observed:
      1. Report the sequence of repeated states of the world comprising all processor firms’ prices as equilibria

The training code to replicate the training process is appended to the supplementary material of this paper. Note that we have not presumed that the supervised agents report necessarily one unique state of the world as the Nash equilibrium in the market. In spatial competition with incorporation of freight rate policies, there is no guaranty for existing a unique verifiable price combination of processors as the mutually best response of players to each other. In such circumstances the cyclic price behaviors can take the place of Nash equilibria. This phenomenon happens due to the discontinuous nature of best response functions of players in agricultural markets with freight absorptions and is discussed in the literature (Dasgupta and Maskin, 1986; Beckmann, 1973; Shubik, 1980; Schuler and Hobbs, 1982; Tesauro and Kephart, 1998).

# Simulation results

The dependent variable of our study is set to be each of two processors’ pricing vectors ( in competition with the opponent in the market environment. The explanatory variable for the pricing policies of the processor agents in the existing literature of spatial competition is a ratio called *importance-of-space* measured by *s=t\*D/ρ* i.e. transport costs (*t*) multiplied distance to the competitor (*D*) divided by net value of product being sold at downstream market (*ρ*) (Alvarez et al., 2000). More the ratio *s* increases, competition between processors diminishes to the point where eventually they are spatially isolated monopsonies. More the ratio *s* decreases, competition between processors becomes more intensive. In order to alter the value of *s* as an explanatory variable in our study, we exogenously change the parameter *t* in the range of the values within the list *t = [0.01 ,0.2, 0.4, 0.6, 0.8, 1.0]*. In addition, we conduct the simulations, once with the factor price elasticity of supply *ε* equal to 1 (which represents a strictly positive (unitary) price elasticity for farms) and once with the factor price elasticity of supply *ε* equal to 0.01 (which represents an extremely non-elastic market supply). The near to zero *ε* parameter can reflect the fact that farmers in the reality might have just limited flexibility to substitute outputs creating relatively inelastic supply in the short run (Gardner, 1992). Each simulation is conducted through 10000 steps and repeated by changing the parameters, agents’ types (supervised or unsupervised), *t* and *ε* according to the above mentioned values. Figure 1 (2) shows the finalized outcome of the simulations by *unsupervised* agents at the final 1000 steps of the agents’ interaction as well as the equilibrium values of obtained by *supervised* agents by setting the *price elasticity of supply* equal to 1 (0.01) for the selected values of transport costs.

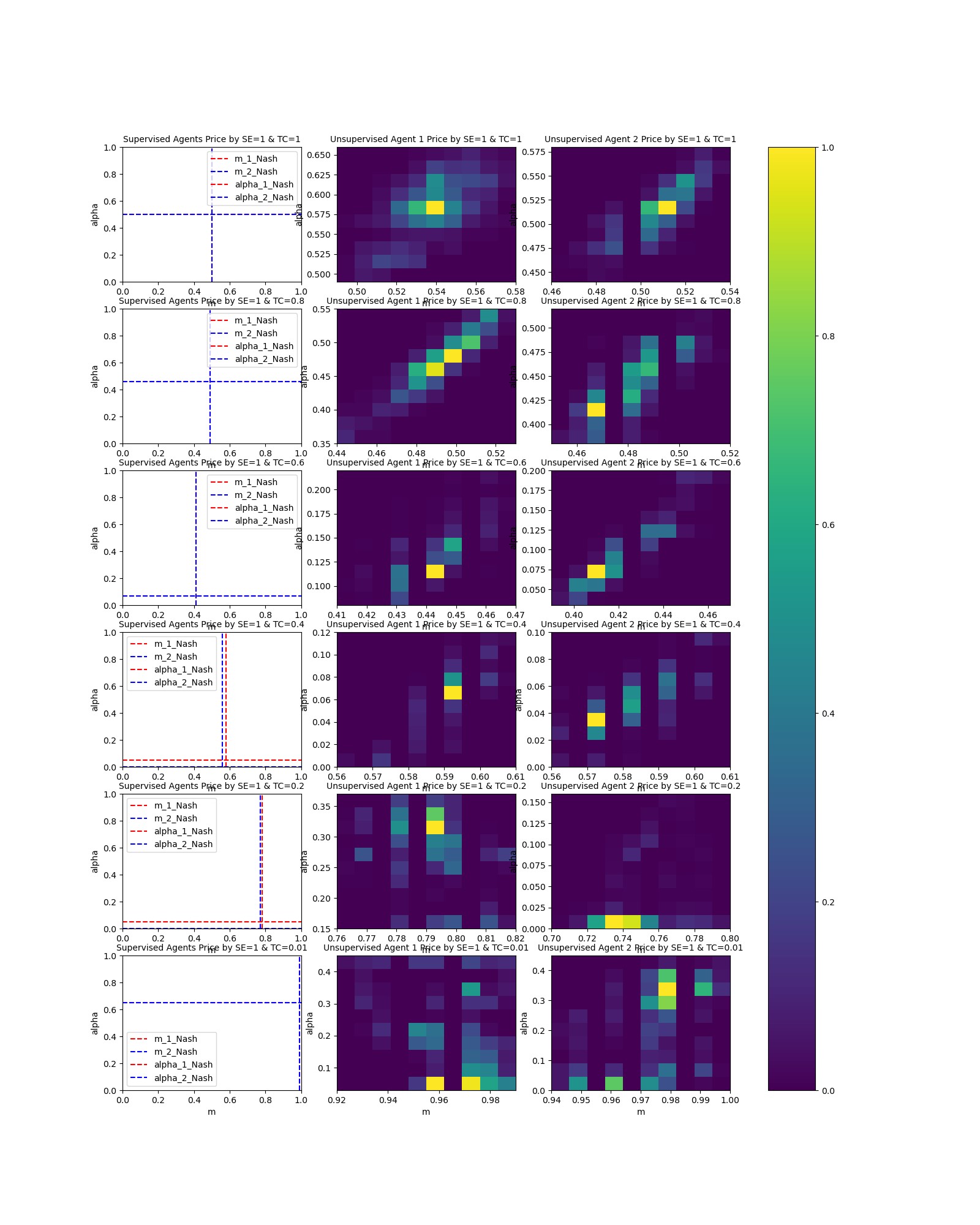


Figure 1. pricing of the duopsony for selected values of *t* in the case of elastic market (*ε=1*)

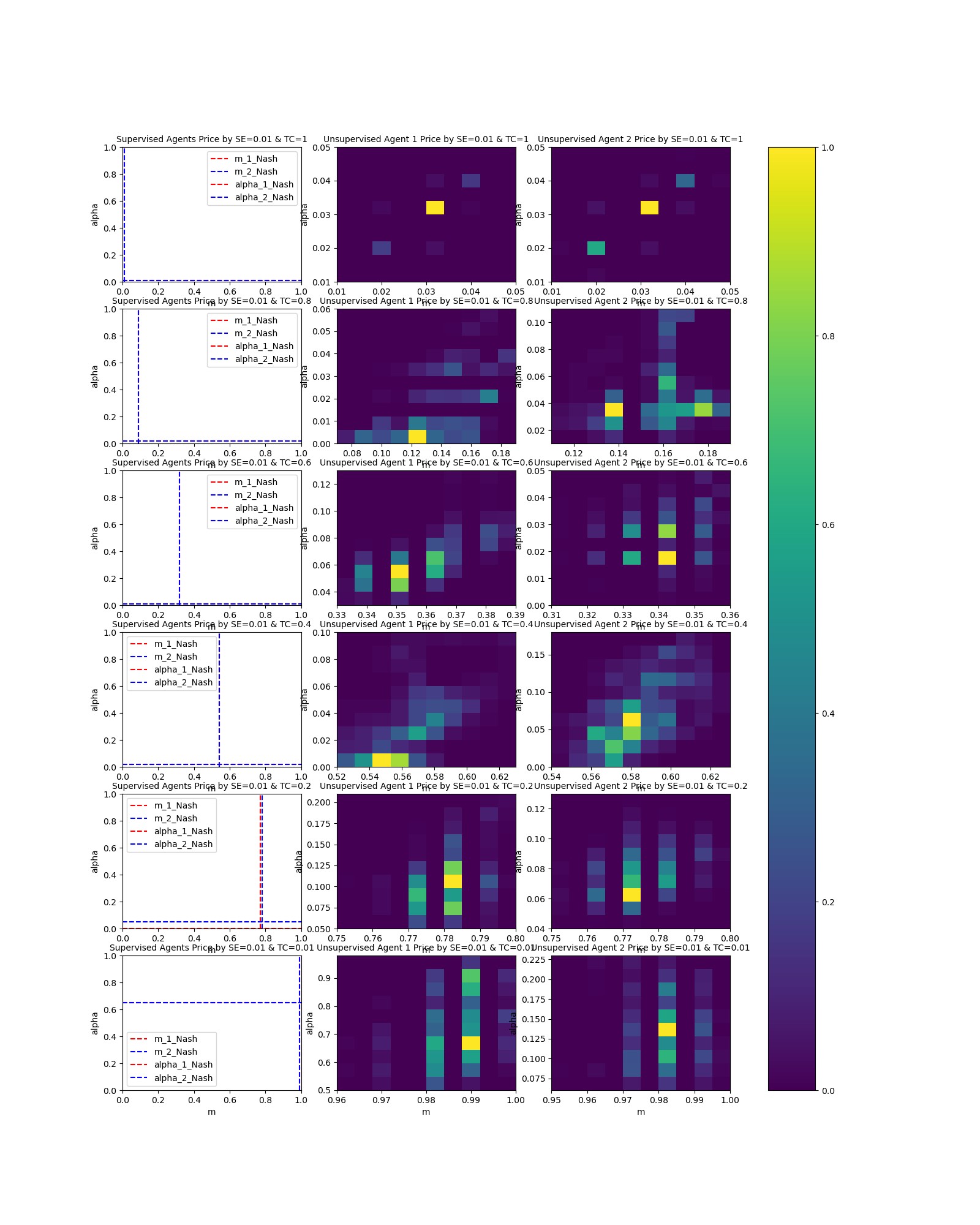


Figure 2. pricing of the duopsony for selected values of *t* in the case of non-elastic market (*ε=0.01*)

The left hand panels of the figures 1 and 2 reveal that the spatial market system does comprise a verifiable *unique* *Nash equilibrium* with practically *symmetric* prices for both processor firms in the entire market settings. That means the case of cyclic price wars i.e. Non-existence of Nash equilibrium is not observed in our study by the underlying market environment and the ranges of selected parameters for transport rate and the price elasticity of supply.

The middle and the right hand panels of the figures 1 and 2 demonstrate the frequency of variable combinations obtained by *unsupervised* competitor agents. The lighter a point within the *m-α* plane, the more frequent the variable combination are observed as the outcome of the game by *unsupervised* agents. The points outside of the depicted minimum and maximum of the *m-α* plane are not-observed pricing policies at the final steps of the agents’ interaction.

The findings with regard to the obtained equilibrium prices in figures 1 and 2 are twofold: first, in the entire market settings within the selected parameters for transport rate and the price elasticity of supply, the deep learning *unsupervised* agents have approximated the Nash equilibrium target points in line with supervised agents’ outcomes, in the course of market interactions by *learning*. This approximation is accomplished with a minor exception when the parameter *t* is set extremely small (*t=0.01*) both in elastic (*ε=1*) and in-elastic (*ε=0.01*) markets. The simulations show that in the extreme cases of setting the lowest possible transportation rate, unsupervised agents proceed in the course of overbidding the mill prices , until they achieve the highest levels of their mill prices i.e. which is completely in line with the Nash equilibrium outcome of the game by supervised agents, however, whereas the *supervised* processors both in the case of elastic market as well as in the inelastic market incorporate (with regard to the absorption of transportation costs), the most frequently observed and values by *unsupervised* agents vary around *0.0-0.4* in the elastic market and around *0.0-0.2* by one of the processor agents in the non-elastic market (whereas the other *unsupervised* agent in figure 2, has approximated the target level similar to the supervised agents’ Nash equilibria points). We can interpret this outcome in relation to the little weight the deep learning might have put on the importance of the (as the coefficient of transport cost) in a market where the role of transport (*t=0.01*) is not significant. In this case, the equilibrium of the market with extremely small transportation cost is expected to recap the *Bertrand solution* with maximum mill prices set by both processors agents and a little incentive to regulate the together with roughly zero profits obtained by both processors. In despite of this deliberation, it would not be surprising if the deep learning processor agents can exactly settle on the target Nash equilibrium points by further learning in extra simulation steps.

The second finding with regard to the obtained equilibrium prices in our study is related to question which prices emerge as Nash equilibrium in spatial agricultural procurement markets?

From figures 1 and 2 it is evident that for the cases of setting the transportation cost rate *t* equal to 1.0 and 0.8, both in the elastic market and the non-elastic market, the market prices go around the monopsonistic optimal discriminatory prices. The monopsonistic optimal discriminatory prices (OD) policy comprises mill price and the freight cost absorption by the monopsony processor equal to the tuple , = (1/(1+ε), 1/(ε+1)) (Löfgren, 1986). Thus where ε=1.0, we expect the *m* and *alpha* variables to converge around the point , and where ε=0.01, we expect the *m* and *alpha* variables to converge around the point , . By decreasing the transportation cost rate *t* to 0.6, 0.4 and 0.2 we observe clearly that the processors’ pricing policies both in the elastic market and the non-elastic market converge towards setting the farm-gate price and bearing the whole transportation costs i.e. . Only in the extremely competitive market setting where the spatial feature of the market almost doesn’t matter i.e. *t=0.01*, processors’ prices tend to involves high absorption of (the actually little) transport costs to be carried by the farms i.e. close to FOB policies.

These outcomes are in one hand thorough with regard to the individual learning- as well as game theoretic foundation of the firms’ policies. On the other hand, our simulation outcomes support the prevalence of pricing policies with absorption of high portions or the entire freight charges through the processing firms in a broad range of market settings, while admitting the emergence of close to FOB policies both in extremely competitive elastic and inelastic market settings. The drawn results are the closest replication of the results of Graubner and Sexton (2022), while relying on a totally different approach i.e. deep learning.

# Conclusion

Price formation in agricultural procurement markets is a complex dynamic process with multiple agents’ interaction. Analytical approaches are confronted with basic limitations to deal with computational complexities. Agent based simulation models are able to simulate actions and interaction of autonomous agents in complex environments. However, individual agents must be equipped with appropriate adaptive learning mechanisms to successfully simulate the emergent pricing policies at the system level. The majority of algorithms in multi agent systems are particularly suited to deal with small games and only few approaches e.g. evolutionary learning models offer a framework for multiple learning agents in rich strategy space. In this paper, we introduce an adaptive dynamic model of pricing in the context of agricultural markets in large-scale strategy spaces, which comprises reinforcement deep learning. We designed experiment runs to examine if the learner agents converge to Nash or close to Nash policies. The results show that deep learning processor agents are capable to converge to the policies, which significantly correlate with robust theoretical basics in line with Nash equilibria. Note that the significance of the deep learning agents is not only restricted to convergence to the targeted theoretical points. It is more significant that the deep learning agents are capable to keep up with the changes in the environment by modeling the state of the environment as input, which consists of the pricing policy vectors of all market processors. Such agents constantly move towards the optimal action in their model’s output to find the optimal behavior. For example, if one of the processors’ policies changes to another direction and settles on a stationary pricing vector, then one competitor deep learning processor agent will update its policy to converge to a policy that is a best-response to the opponent processor’s policy. A further finding of our research provides insights to the research question of emerging discriminatory pricing policies in agricultural input markets. While our simulations support the theoretical and empirical reflections of the most recent existing studies with regard to the emergence of optimal discriminatory and uniform delivery pricing policies in a broad ranges of market settings, they show the emergence of close to FOB policies both in extremely competitive elastic and inelastic market settings.

While our study’s objective with regard to the research design within this paper is attained, there exist limits in the scope of our study, which need to be addressed in future research. For example, our simulations are still limited to a linear duopsony market, just 2 agents and fixed firm locations. The scope of this research can be extended to further examine the capability of deep learning firms to simulate more complex environments. By increasing the complexity of the market environment, more enhanced versions of deep learning models will be required e.g. those deep learning techniques, which can elaborate on hierarchical levels of analyzing to cope with additional market features’ computational burden. Such market environment features of the real world markets comprise e.g. two dimensional market spaces, pricing with presence of multiple agents and or with incorporating agents, who decide upon the joint selection of pricing and location.

# Supplementary material

You can find further material related to this paper consisting of code, data, results and figures in the [GitLab account](https://gitlab.uni-koblenz.de/hamedkhalili/covid_ai_project/-/tree/main/p_1) corresponded to this paper, which is provided by University of.

# References

Alvarez, A.M., Fidalgo, E.G., Sexton, R.J. and Zhang, M. (2000). Oligopsony Power with Uniform Spatial Pricing: Theory and Application to Milk Processing in Spain. European Review of Agricultural Economics, Vol. 27(3), pp. 347-364.

Beckmann, M.J. (1976). Spatial Price Policies Revisited. Bell Journal of Economics 7, pp. 619-630.

Beckmann, M.J. (1973). Spatial Oligopoly as a Noncooperative Game. International Journal of Game Theory 2, pp. 263–268.

Bellman, R. (1957). Dynamic Programming. Princton, New Jersey: Princton University press.

Brenner, T. (2005). Agent Learning Representation –Advice in Modelling Economic Learning. Germany: Max Planck Institute for Research into Economic Systems.

Busoniu, L., Babuska, R. and De Schutter, B. (2008). A comprehensive survey of multiagent reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 38 (2), pp. 156-172.

Busoniu, L., Babuska, R. and De Schutter, B. (2010). Multi-agent reinforcement learning: An overview. In: D. Srinivasan, L. Jain (eds.) Innovations in Multi-Agent Systems and Applications - 1, Studies in Computational Intelligence, vol. 310, pp. 183–221. Berlin Heidelberg: Springer.

Dasgupta, P. and Maskin, E. (1986). The Existence of Equilibrium in Discontinuous Games. Applications. Review of Economic Studies 53, pp. 27-41.

Durham, C.A., Sexton R.J. and Song J.H. (1996). Spatial Competition, Uniform Pricing and Transportation Efficiency in the California Processing Tomato Industry. American Journal of Agricultural Economics 78, pp. 115-125.

Espinoza, M. P. (1992). Delivered pricing, FOB pricing, and collusion in spatial markets. Rand Journal of Economics 23, pp. 64-85.

Fousekis, P. (2011). Free-on-board and Uniform Delivery Pricing Policies in a Mixed Duopsony. European Review of Agricultural Economics 40, pp. 119-139.

Fousekis, P. (2011). Spatial Price Competition Between Cooperatives Under Hotelling- Smithies Conjectures. Agricultural Economics Review, 12, pp. 4-15.

Fudenberg, D. and Levine, D. K., The theory of learning in games. (1998). Cambridge, MIT Press.

Gardner, B. (1992). Changning Economic Perspectives on the farm problem. Journal of Economic Literature 30, pp. 62-101.

Graubner, M., Koller, I., Salhofer, K. and Balmann, A. (2011a). Cooperative versus Non-Cooperative Spatial Competition for Milk. European Review of Agricultural Economics. European Review of Agricultural Economics, Vol. 38(1), pp. 99-118.

Graubner, M., Balmann, A. and Sexton, R.J. (2011b). Spatial Price Discrimination in Agricultural Product Procurement Markets: A Computational Economics Approach. American Journal of Agricultural Economics, Vol. 93(4), pp. 949-967.

Graubner, M., Sexton, R.J., More competitive than you think? Pricing and location of processing firms in agricultural markets, American journal of agricultural economics: 22 July 2022 <https://doi.org/10.1111/ajae.12336>

Greenhut, M.L. and Ohta, H. (1972). Monopoly Output under Alternative Spatial Pricing Techniques. American Economic Review 62, pp. 705-713.

Greenhut, G. (1981). Spatial pricing in the USA, West Germany and Japan. Economica 48, pp. 79-86.

Hotelling, H. (1929). Stability in Competition. Economic Journal 39, pp. 41-57.

Howard, R. (1960). *Dynamic programming and Markov process.* MIT Press.

Huck, P., Salhofer, K. and Tribl C. (2006). Spatial Competition of Milk Processing Cooperatives in Northern Germany. International Association of Agricultural Economists Conference.

Kats, A. and Thisse, J.F. (1993). Spatial oligopolies with uniform delivered pricing. In H. Ohta and J.-F. Thisse (eds), Does Economic Space Matter? New York: St Martins Press, pp. 274-296.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature* , 436:521.

Löfgren, K. (1986). The Spatial Monopsony:A Theoretical Analysis. Journal of Regional science 26, pp. 7070-730.

Luke, S. and Wiegand, R. P. (2002). Guaranteeing coevolutionary objective measures. See Poli, Rowe, and Jong (2002), pp. 237–251.

Maskin, E. and Tirole,J. (2001). Markov perfect equilibrium. Journal of Economic Theory, Vol.20, pp. 191-215.

Mitchell, M., 1996. An Introduction to Genetic Algorithm. MIT Press, Cambridge (MA).

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Bellemare, M., & . et al. (2015). Human-level control through deep reinforcement learning. *Nature* , 518-529.

Riechmann, T. (2006). Cournot or Walras? Agent based learning, rationality, and long run results in oligopoly games, Journal of Institutional and Theoretical Economics 19, pp. 702-720.

Rogers, R. T. and Sexton, R.J. (1994). Assessing the Importance of Oligopsony. American Journal of Agricultural Economics 76, pp. 1143–1150.

Scherer, F. M. (1980). Industrial Market Structure and Economic Performance. 2nd edn. Chicago: Rand-McNally College Publishing co.

Schmidhuber, J. (2015). Deep learning in neural networks: an overview. *Neural Netw. 61*, 85–117.

Schuler, R. E., and Hobbs, B. F. (1982). Spatial Price Duopoly under Uniform Delivered Pricing. Journal of Industrial Economics 31, pp. 175–187.

Sexton, R. J. (1990). Imperfect Competition in Agricultural Markets and the Role of Cooperatives. American Journal of Agricultural Economics, Vol.72 (3), pp. 709-720.

Sexton, R. J. (2012). American Journal of Agricultural Economics, Volume 95, Issue 2, 1 January 2013, pp. 209–219.

Sexton, R. J., Kling, C. L. and Carman, H. F. (1991). Market integration, efficiency of arbitrage, and imperfect competition: Methodology and application to U.S. celery. American Journal of Agricultural Economics 73 (3), pp. 568-580.

Silver, D., Huang, A., Maddison, C., Guez, A., & et al. . (2016). Mastering the game of go with deep neural networks and tree search. *Nature 529* , 484–489.

Sutton, R.S. and Barto, A. (1998). Reinforcement Learning: An Introduction. Cambridge: MIT Press.

Vriend, N. J. (2000). An illustration of the essential differences between individual and social learning, and its consequences for computational analyses. Journal of Economic Dynamics & Control 24, pp. 1-19.

Watkins, C. J. (1989). Learning from delayed rewards. Ph.D. thesis, Cambridge University.

Zhang, M. and Sexton, R. J. (2001). FOB or Uniform Delivered Prices: Strategic Choice and Welfare Effects. Journal of Industrial Economics 49, pp. 197-221.