AGENT-BASED ARTIFICIAL MARKETS

By

TONG ZHANG

Bachelor of Science in Physics
Jilin University
Changchun, P.R. China
1995

Master of Science in Mechanical Engineering
Beijing Institute of Technology
Beijing, P.R. China
2001

Master of Science in Agricultural Economics
Oklahoma State University
Stillwater, Oklahoma
2005

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
In partial fulfillment of
The requirements for
The Degree of
DOCTOR OF PHILOSOPY
July, 2008

AGENT-BASED ARTIFICIAL MARKETS

Dissertation Approved:

Dr. B. Wade Brorsen

Dissertation Advisor

Dr. James R. Fain

Dr. Jayson L. Lusk

Dr. Clement E. Ward

Dr. A. Gordon Emslie

Dean of the Graduate College

PREFACE

This dissertation is composed of three essays. The first essay, "A Particle Swarm Optimization Algorithm (PSO) for Agent-Based Artificial Markets," adapts PSO to simulate dynamic economic games and compares the robustness and speed of the PSO algorithm to a genetic algorithm (GA) in a Cournot oligopsony market. Artificial agents with the PSO learning algorithm find the optimal strategies that are predicted by theory. PSO is simpler and more robust to changes in algorithm parameters than GA. PSO also converges faster and gives more precise answers than the GA method which was used by some previous economic studies. The agent-based model is a suitable tool to study complex economic problems that are hard to solve with analytical methods and it is also new to agricultural economics.

The second essay, "Collusion and Competition of Oligopsony Firms with Quantity-Price Strategic Decisions: An Agent-Based Artificial Market," uses an agent-based model to determine market equilibrium with price-setting firms in oligopsony markets. With price setting firms, the Bertrand solution is perfect competition and the Bertrand-Edgeworth model has a mixed strategy solution in which firms keep on changing prices. But experiments with human subjects find the market equilibrium varies with the number of firms in the industry. In our simulation, the learning of agents is modeled with the particle swarm optimization algorithm. The results show that with one or two firms prices are at the monopsony level and with four firms prices are always at the perfectly competitive level. The triopsony market, however, changes from mostly monopsony to perfect competition when capacity cost increases from zero to a higher level. The results also show that firms tend to have excess capacity. The results are similar to results observed in experiments with human subjects. This suggests that people use the heuristic rules assumed in the agent-based model rather being fully rational as assumed in the traditional Bertrand and Bertrand-Edgeworth models.

The third essay, “The Long Run and Short Run Impact of Captive Supplies on Spot Market Price: An Agent-Based Artificial Market,” uses an agent-based model to determine the impact of captive supplies under short run and long run assumptions in the fed cattle market. In the simulated market, packers purchase cattle from feeders with both exclusive captive supply contracts and in the spot market. The price of captive contracts is linked to the spot market price. The captive contracts are fixed in the short run but flexible in the long run. Simulation results indicate that packers can depress the spot market price in the short run if the contracts are fixed. This result matches Xia and Sexton’s model. But, this is a short run effect. In the long run when the packers can adjust the number of captive supply contracts and feeders have a supply response for contract quantity, the price depression phenomena disappears.

ACKNOWLEDGEMENTS

My transformation from a software engineer to a future economics faculty member seems as unimaginable now as when I worked in China. I owe my gratitude to all those people who have helped along the road and because of whom my graduate experience has been one that I will cherish forever.

My deepest gratitude is to my advisor, Dr. B. Wade Brorsen. I have been amazingly fortunate to have Dr. Brorsen as my dissertation advisor. He supported my interest in agent-based models, suggested important ideas, and had the patience to get me back on the right track away from my latest fancy new ideas. His sharp advice and insight guided my research. I learned so much from him and I would say that he changed my life by teaching me how to do research, which let me have the confidence and opportunity to step into the academic arena. I hope that one day I will become as good an advisor to my students as Dr. Brorsen has been to me.

I am thankful also to Dr. Francis Epplin for accepting me as a master's student in the agricultural economics department 6 years ago. Maybe he saw potential in me or just wanted to give me a chance even though at that time my spoken English was horrible. He is always kind and helpful when I seek suggestions from him.

I am also thankful to my master's thesis advisor, Dr. Chanjin Chung. Because of my weak economics background and bad English 5 years ago, he had to try his best to communicate with me and guide my thesis, which helped me finish my thesis successfully and got me into the PhD program at OSU.

My committee members, Dr. Jayson Lusk, Dr. Clement Ward, and Dr. James Fain, have been always supportive and given valuable advice. I am deeply grateful to them for their discussions and insightful comments on my research. Also, Dr. Lusk's course helped me understand the ideas of auctions and game theory that are an integral part of my dissertation.

I am also indebted to other faculty, Dr. Art Stoecker, Dr. Shida Henneberry, Dr. Brain Adam, Dr. Joe Schatzer, Dr. Antonio Camara, and Dr. Bailey Norwood. I either benefited from their courses or from valuable discussions with them.

Further, I would like to acknowledge Anna Whitney, Gracie Teague, Joyce Grizzle, Judy Rudin, Wanda Dollarhide, Andy Lowery, and other staff at Oklahoma State University, for their help, kindness and various forms of support during my graduate study.

My friends' support helped me overcome the difficult times and to stay focused on my graduate study. I greatly value their friendship.

Most importantly, none of this would have been possible without the love and support of my husband Yu Wu and my parents. Without you, I would not accomplish this degree.

Finally, I thank Dr. James Trapp and appreciate the financial support from the agricultural economics department that funded the research of this dissertation.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

A PARTICLE SWARM OPTIMIZATION ALGORITHM FOR AGENT-BASED ARTIFICIAL MARKETS

Introduction

Agent-based models are increasingly used to study economic phenomena and are especially suitable to simulate economic games in which agents interact with each other with bounded rationality and adaptive learning rules. Work to date has shown that such models can obtain the same results as theoretical models (Arifovic 1994; Alkemade, Poutre, and Amman 2006). Agent-based models offer considerable potential to study auctions and market mechanism designs as well as more traditional industrial organization topics. They have the potential to study much more complex economic problems than can be analyzed theoretically, such as markets containing heterogeneous agents, or agents using combinatorial strategies. They also have a potentially much lower cost than experimental markets with human subjects.

To date, however, the complexity of the agent-based models has been limited. One limitation of these models is the time it takes to find an optimum, the others are the algorithm complexity and robustness to algorithm parameters. Previous research using agent-based models in economics have used either a genetic algorithm (GA) (Arifovic 1994 and 1996; Axelrod 1987; Bullard and Duffy 1999; Riechmann 2001; Vriend 2000) or reinforcement learning (RL) (Erev and Roth 1998; Kutschinski, Uthmann, and Polani 2003). With GA, researchers have to be very careful to choose parameters and methods for each problem or it may cause premature convergence. The large population size required also makes GA slow to find equilibrium. RL is a sub-area of machine learning and the environment is typically formulated as a finite-state Markov decision process in which an agent increases the probability of choosing successful strategies under the possible strategy spaces of its rivals. When the possible strategy space is large or continuous, the computational cost increases exponentially. To avoid the problems of GA and RL, we use a particle swarm optimization (PSO) to model the learning behavior of agents.

PSO is a stochastic optimization technique developed by Eberhart and Kennedy (1995). The idea of PSO came from watching the way flocks of birds, fish or other animals adapt to avoid predators and find food by sharing information. In PSO, a set of randomly generated solutions moves towards the optimal solution over a number of iterations by assimilating and sharing information among all members of the swarm.

PSO has been shown to have the same ability to find a global optimum as genetic algorithms, but to be able to find optimums faster than genetic algorithms (Panda and Padhy 2007; Mouser and Dunn 2005; Hassan et al. 2005). Existing PSO methods, however, cannot be directly applied to solving agent-based models. With an agent-based model, all agents solve their own optimization problems under a dynamic economic environment since an agent's profit depends on the actions of other agents.

The objective of this essay is to adapt PSO to solve an agent-based model under a dynamic environment with non-cooperative agents. We also compare the proposed PSO algorithm to a genetic algorithm for finding equilibrium in the Cournot oligopsony market.

Theoretical Model of Oligopsony Market

The Cournot oligopsony market describes a situation where a few buyers compete in a market and each of them can influence the market price through a common price supply curve. In this situation, buyers must make strategic decisions, taking into account the decisions of their rivals.

Consider a homogeneous product market with M buyers and N sellers. The number of buyers is much less than the number of sellers ($M << N$). Assume that buyers process products that will be sold in the retail market and the marginal cost for processing is constant for all processors. The marginal value equals the selling price minus the marginal processing cost. To focus our research on the games between buyers and sellers in this market, we assume the final product price P and the marginal processing cost mc are constant. Thus the value of product before processing $R = P - mc$ is also constant.

Each firm uses processing ratio as its choice variable:

(1.1) $$x_i = q_i^d / (R \times N),$$

here x_i is the processing ratio, q_i^d is the processing quantity of the firm and it also defines the amount of procurement, R is the marginal revenue of product and also the supply level of sellers under the perfect competition price level, N is the total number of sellers. For example, if under perfect competition market all sellers will provide 10,000 products and the processing quantity of processor i is 3,000, its processing ratio x_i equals 0.3.

Then the total demand can be written as $D = \sum_{i=1}^{M} q_i^d$.

At the beginning of each processing period, buyers make procurement strategies x simultaneously. We assume all sellers are homogeneous and have supply function $q_j^s = p$, so the total supply is $S = Np$. The total demand of buyers will determine the market price together with the aggregate supply of sellers and the market price is $p = D/N$. In the simulation market, there are 4 buyers and 100 sellers, R equals \$100. According to theory, under perfect competition, the market price is \$100, the aggregate supply is 10,000 and the processing ratio is 25% for each firm; if the market reaches Cournot-Nash equilibrium, the market price equals \$80 and the processing ratio is 20% for each firm.

Particle Swarm Optimization Algorithm

This research adjusts PSO for a non-cooperative game by constructing multiple parallel markets and letting each agent have its own clones in every market. This means each agent has a separate "flock of birds" that does not share information with the flocks of other agents. The asynchronous best strategies of one agent in every parallel structure are called local best solutions and the best fit strategy among all parallel structures at the current simulation iteration is called the global best solution.

Each firm has its clones in every parallel market and these clones trade independently and simultaneously in all markets. We can look at this setting as firms separate the sellers into groups or a longer time into multiple periods and try different strategies within each group or period. This kind of marketing strategy can be observed in many real markets. For example in fed cattle markets, packing firms send many agents to purchase cattle from feeders and each of them visits feeders in a certain area. Agents bid differently but they will share information at the end of each period and adjust their strategies to increase profit. In real world markets, the dynamics of market prices are mostly path dependent which means the market prices only change a small value each time. So the adaptive feature of PSO is similar to actual learning behavior.

Since agents are continuously changing their strategies, the pervious local best solutions may not be the best for the current period. Thus, we adjust PSO by retesting the historical best locals of each agent under the current market environment and choose the best fit strategies as the current best locals. Every agent continuously uses its own PSO algorithm searching for better solutions in each parallel market guided by their own best local and global solutions.

PSO Algorithm Description

We set up K parallel markets and letting the M buyers each have their own clones in every market. Although having the same behavior rules, one agent and its K clones may take different market strategies since the initialized random values are different. In the simulation, buyers dynamically change their marketing strategies with the PSO algorithm but their sellers are price takers and simply sell their products to the current highest bidders.

The clone of firm i in the k^{th} parallel market has a quantity ratio value $x_{i,k} \in [0,1]$ as a strategy parameter, and each strategy parameter is randomly selected from a $U(0,1)$ distribution at the beginning of the simulation. Each clone has an evolutionary velocity, $v_{i,k} \in [-1,+1]$, which determines the change of its strategy. The changes of the clones' strategies are influenced by the location of the best solutions achieved by itself, $p^{l}_{i,k} \in [0,1]$ for the k^{th} clone, and by the whole population, $p^{g}_{i} \in [0,1]$. The superscripts l and g indicate local and global, the subscripts k and i indicate k^{th} parallel market and i^{th} firm respectively. Profit function $\pi_k(x_{i,k})$ is used to value the performance of each strategy $x_{i,k}$.

In every simulation step, the strategy of the i^{th} firm in the k^{th} parallel market is updated by the following equations:

$$x_{i,k,t+1} = x_{i,k,t} + v_{i,k,t}, \tag{1.2}$$

$$v_{i,k,t+1} = w \cdot v_{i,k,t} + c_1 u_1 (p^{l}_{i,k,t} - x_{i,k,t}) + c_2 u_2 (p^{g}_{i,k,t} - x_{i,k,t}), \tag{1.3}$$

where $x_{i,k,t}$ is the procurement ratio in period t, $v_{i,k,t}$ is the velocity vector, $u_j \in [0,1]$, $j = 1, 2$ are uniformly distributed random numbers, c_1 and c_2 are learning parameters and can be called self confidence factor and swarm confidence factor respectively, and w is an inertia weight factor.

The following equations indicate how to choose $p^{l}_{i,k,t}$ and $p^{g}_{i,t}$ among all parameters of firm i. In economic games, the payoff of one agent's strategy is also determined by the strategies of its rivals and the changing of its rivals' behaviors forms the dynamic economic environment of this agent. This may cause the agent's previous best local strategies not perform well in the current period. Thus we reevaluate an agent's best strategy by using its L previous best locals to trade versus other agents' current period strategies and compare their payoffs with that of its current strategy, and then choose the best among them as the best local of the current period. This procedure can be written as:

(1.4) $$p^{l}_{i,k,t} = \arg\max\left\{\pi_k(p^{l}_{i,k,t-1}),\ldots\ldots,\pi_k(p^{l}_{i,k,t-L}),\pi_k(x_{i,k,t})\middle| x_{i'\neq i,k,t}\right\},$$

where i' indicates firm i's rivals. The best global is selected from the best local parameters:

(1.5) $$p^{g}_{i,t} = \arg\max\left\{\pi_1(p^{l}_{i,1,t}),\pi_2(p^{l}_{i,2,t}),\ldots,\pi_K(p^{l}_{i,K,t})\right\},$$

where $k = 1,2,...,K$ and K is the total number of parallel markets.

Chatterjee and Siarry (2006) state that the inertia weight w in (1.3) is critical for the convergence behavior of PSO. A large inertia weight provides a larger exploration but a smaller one is needed to fine-tune the current search area. So it is worth making a compromise, e.g. start w with a higher initial weight at the beginning and then decrease it with iterations:

(1.6) $$w_t = \beta_0^w + \beta_1^w(t_{\max} - t)/t_{\max},$$

where both β_0^w and β_1^w are constants, $t_{\max}$ is the maximum number of iterations in one simulation round and t indicates the current iteration. The self confidence factor c_1 and swarm confidence factor c_2 in equation (1.3) are set as:

(1.7) $$c_{1,t} = c_{2,t} = \beta_0^{c_1} + \beta_1^{c_1}(t_{\max} - t)/t_{\max},$$

where both $\beta_0^{c_1}$ and $\beta_1^{c_1}$ are constants.

Summary of Simulation Procedure with PSO

In the Cournot oligopsony game, buyers select independently and simultaneously the quantity they produce. The total supplies along with the demand curve determine the retail price. There are K parallel markets. The M buyers act as independent agents and trade in each market at the same time. Each firm may have a different trading strategy in each parallel market. The steps are as follows:

(i) For the first L beginning iterations, randomly initialize strategy set $\mathbf{x}$ for all buyers in every parallel market. We choose the quantity ratio $x_{i,k,t} \in U[0,1]$ and the movement velocities $v_{i,k,t} = 0$ for $i = 1,...,M$, $k = 1,...,K$, and $t = 1,...,L$.

(ii) Buyers update their strategies with equations (1.2) and (1.3).

(iii) After the first L iterations, each buyer retest the past L best locals under current economic environment and compare their performance with that of the current strategy, the best among them is chosen as the new best local, as equation (1.4) shows.

(iv) Following equation (1.5), the best fit among all best locals is the best global.

(v) If the market does not reach equilibrium, go to step (ii).

Genetic Algorithm

GA is a general-purpose optimization method based loosely on Darwinian principles of biological evolution, reproduction and the survival of the fittest (Goldberg 1989). GA maintains a pool of candidate solutions called a population and repeatedly modifies them. At each step, the GA selects candidates from the current population to be parents and uses them to produce children for the next generation. Over successive generations, the population evolves toward an optimal solution. The GA is well suited to and has been extensively applied to solve complex design optimization problems because it can handle both discrete and continuous variables, as well as nonlinear objective and constraint functions. The recent GA study by Alkemade, Poute and Amman (2006) indicates that to avoid premature convergence of the evolutionary algorithm, each agent should have a large population of strategies from which agents can choose.

Genetic Algorithm Operators and Parameters

In this GA, a strategy of each firm can be represented with a chromosome which contains information about this strategy. The most used way of encoding is a binary string. We use *B* bit binary strings to encode strategies and the bits can be looked at as genes. Each firm has a population of *K* chromosomes*,* represents a collection of its strategies at time period *t*. The k^{th} strategy of firm i in period t can be stated with a string of length *B* as:

(1.8) $$\left\langle a_{i,k,t}^{B}, a_{i,k,t}^{B-1}, \ldots, a_{i,k,t}^{1} \right\rangle,$$

here $a_{i,k,t}^{b} \in \{0,1\}$ taken at the b^{th} position in the string, $b \in \{1,2,...,B\}$, and can be decoded into a decimal integer using

(1.9) $$d_{i,s,t} = \sum_{b=1}^{B} (a_{i,k,t}^{b} 2^{b-1}) .$$

The maximum value is $d_{\max} = \sum_{b=1}^{B} 2^{b-1}$. After choosing one active strategy $d_{i,k,t}$, the firm's procurement ratio can be calculated with $x_{i,k,t} = d_{i,k,t} / d_{\max}$. For example, if a string contains 4 bits, a binary code "0101" can be decoded to decimal value $d_i = 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$, and the maximum binary code of this string "1111" can be decoded to value $d_{\max} = 15$. Thus a firm using "0101" as its strategy means its procurement ratio is $x_{i,k,t} = d_{i,k,t} / d_{\max} = 5/15 = 30\%$. Since the larger the string length, the more accurate the procurement ratio, we use 15 bits as the string length in the simulation.

Buyers' decision rules are updated using four genetic operators, *elitism, reproduction, crossover,* and *mutation*. Elitism can very rapidly increase performance of GA, because it prevents losing the best found solution to date (De Jong, 1975). In our Cournot game, the profit difference between strategies could be very big. To avoid one high profit strategy dominating the next generation with profit proportional selection, ranking selection is used as the reproduction method.

Elitism copies a few of the best strategies from the current K strategies to the new population with an elitism rate ε. If $\varepsilon = 10\%$ and $K = 100$, this means the 10 best strategies are copied from the old population to the new one. The rest are chosen with linear ranking selection.

Reproduction chooses chromosomes as parents from the old strategy population. In this research, we use ranking selection methods. It ranks an agent's strategies in its population 1 to K from worst to best according to their profit (K = population size). If more than one strategy has the same profit, they are randomly ranked. The selection probabilities of the strategies x_k (k =1,…,K) are given by

(1.10) $$p(x_k) = \frac{1}{K}\left(r_{\min} + \frac{(r_{\max} - r_{\min})(rank(x_k) - 1)}{K - 1}\right),$$

where p is the probability of strategies being chosen as new ones, $r_{\max} + r_{\min} = 2$, and $1 \le r_{\max} \le 2$. We choose $r_{\max} = 1.1$ and $r_{\min} = 0.9$ and it is easy to see $\sum p(x_k)$ equals 1.

Crossover selects genes from two parent chromosomes and creates a new strategy. All new strategies selected with elitism and ranking selection methods are randomly matched as a group of parent chromosomes. For each pair of parents, the crossover is performed with a probability χ. The crossover randomly chooses an across point of the chromosome string, and bits before and after this point are exchanged for both chromosomes to generate new ones. Crossover can look like this (| is the crossover point):

Chromosome 1	11000 | 1011010100
Chromosome 2	10010 | 1101000001
New Chromosome 1	10010 | 1011010100
New Chromosome 2	11000 | 1101000001

Mutation takes place after a crossover is performed with a probability μ. This operator is to prevent falling into a local optimum. In the binary encoding method, mutation changes the bits of the new strategy from 1 to 0 or 0 to 1 with the mutation rate μ.

Like PSO, we can also use niching methods in GA, the function of parameters for three operators are

(1.11) $$\varepsilon_t = \beta_0^{\varepsilon} + \beta_1^{\varepsilon} \cdot t / t_{\max},$$

(1.12) $$\chi_t = \beta_0^{\chi} + \beta_1^{\chi} \cdot (t_{\max} - t) / t_{\max},$$

(1.13) $$\mu_t = \beta_0^{\mu} + \beta_1^{\mu} \cdot (t_{\max} - t) / t_{\max},$$

here ε_t, χ_t and μ_t indicate elitism, crossover and mutation rate, β s are constant, $t_{\max}$ is the number of maximum iterations of the one simulation round.

GA Simulation Procedure for Cournot Market

At the beginning of the simulation, buyers randomly generate strategies as new rules in the starting population. Every strategy in the population is randomly chosen to trade in the market. A new population is generated from the current one with following procedure. First, εS highest strategies are copied to the new population as elites. Then $(1-\varepsilon)S$ strategies are chosen with ranking selection methods from the whole population of the old generation, and are randomly matched and crossed over. Mutation operation is performed for the new strategies except the elites. Figure 1 gives the outline of the program.

Comparison of Algorithm Performance

In this research, we use the Cournot game with known Nash equilibrium to evaluate the performance of PSO and GA. We also define the algorithm convergence criterion in this section. We design 12 parameter settings for PSO and GA respectively under three categories, fixed and changing algorithm parameters, and different algorithm structures.

One simulation round contains multiple iterations and agents trade with each other repeatedly. Within these periods, agents use PSO or GA to update their strategies based on their rivals' strategies. Considering the randomness of the learning path, under each setting we run the Cournot game for 20 rounds with different random initialized strategies and the game is repeated for 400 iterations per round to see the learning behavior of agents and the equilibrium under each set.

Algorithm Convergence Criteria

Zero diversity in the population's strategy values signals the stopping point for GA and PSO. For every agent, if the variance of the strategies in the population is less than 0.01% and the variance of the mean value of the strategies for 10 generations is less than 0.01%, we say the algorithm reaches equilibrium. Considering the feature of mutation of GA, we delete 5% of the strategies that have the largest difference from the mean when calculating the mean and variance for GA.

Robustness Analysis

Robustness to small variations in the technical parameter settings (that have no clear economic meaning) is particularly important in agent-based models. It is important that results are valid for a wide range of parameter settings.

PSO and GA are nondeterministic and are not guaranteed to return to the same solution in each run. The speed and accuracy of the algorithms can vary depending on the chosen parameters. We test robustness of the conclusions by comparing the performance under alternative algorithm parameters, population size and retest times, which are listed in Table I-1.

Simulation Results

Hamm, Brorsen, and Hagan (2007) recommend using multiple sets of starting values when using genetic algorithms. Thus for each set of algorithm parameters, we run 20 times with different randomly generated initial strategies and then calculate the mean and standard deviation of the market price and players' procurement ratios at the 20 equilibrium points.

Robustness Analysis

We compare the performance of the PSO and GA with three categories: fixed algorithm parameters, changing algorithm parameters and different algorithm structure parameters. In each setting in the fixed algorithm category, we use constant values for the algorithm parameters, (w , c_1 , and c_2 for PSO, ε , χ ,and μ for GA). In each setting of the changing algorithm category, the algorithm parameters are changing with time as shown in equations (1.6), (1.7), and (1.11) to (1.13). Under our design of the market, theoretically the Nash equilibrium of market price is $80 and the procurement ratio for each buyer is 20%. These values are used to evaluate the performance of the algorithm.

Table I-2 gives the simulation results under fixed algorithm settings. For PSO, all settings give near Nash equilibrium solutions with low variances. For GA, all four settings give a near Nash equilibrium market price but buyers may have heterogeneous strategies at the market equilibrium.

Table I-3 presents the results of the changing algorithm parameter settings. PSO gives near Nash equilibrium results for all 4 settings with a small standard deviation. The settings of GA do not show much difference from the fixed algorithm parameter settings.

Table I-4 shows the results with different algorithm structures. For both PSO and GA, the large population size gives a better performance, but PSO is not as sensitive as GA to the algorithm structure changes. PSO settings with both large and small population size give near Nash equilibrium results for market price and buyers' strategies with small differences. GA settings with small population size show bigger differences in buyers' strategies.

For both PSO and GA with changing parameters, programs need less machine time and iterations to reach equilibrium than with fixed algorithm parameters. And GA generally uses around 15 to 80 fold more machine time than PSO. This is because GA needs coding and decoding of binary string bits and additional evaluation and calculation for the ranking and roulette selection.

From the above analysis, the overall performance of PSO is considerably faster and more precise than GA and less sensitive to the value of parameters.

Individual Runs

After analyzing the overall results, we also choose the best performance parameter set of PSO and GA out of all settings in both fixed and changing parameter category, draw the figure of one individual simulation run for each of them to illustrate how the buyers find the best response strategies during the dynamic environment and how the markets reach the equilibrium.

We choose the best performance parameter sets from fixed and changing categories for each algorithm, which are sets 2 and 7 in Table I-2 and 12 and 14 in Table I-3. Then we illustrate the evolution of market price level with figures under each setting, as Figure I-2 (a) and (b) show. In this example, both GA and PSO reach equilibrium at about the same time. The GA continues to mutate as the iterations proceed, but the mutations are quickly discarded.

When looking at the individual agent's marketing strategies, we found that with the PSO algorithm, agents in the markets tend to have the same strategy under market equilibrium, which is predicted in theory, while GA shows that individuals could have different strategies, as Figures I-2 and I-3 show. The difference can be explained by the learning methods adopted by the two algorithms. In GA, if the strategy population of one firm converges faster then others and others continue to adjust their strategies, the market results in equilibrium with heterogeneous strategies. Different from GA, besides considering the historical performance, global best in the parallel market is also taken into account in PSO, so one firm has little probability to take a larger market share than others. Then the equilibrium of PSO usually contains homogenous strategies of agents. Similarly, the quantity levels shown in Figures I-2 and I-3 show the much faster convergence of PSO relative to GA.

Summary and Conclusions

This paper adapts PSO to simulate agent-based models by allowing each agent to have its own parallel structures and learn from them. We also compare the proposed PSO algorithm to a genetic algorithm for finding equilibrium in the Cournot oligopsony market.

We find that with trial and error, artificial agents using PSO learning algorithm can learn to play the best response strategies as theory predicts. PSO needs few parameters and the simulation results are more robust to changing parameters than GA. The parameters of GA need to be carefully chosen to suit the specific simulation problem. It also requires parameter tuning for good performance and can sometimes be computationally expensive.

The comparison is undertaken under a relatively simple economic market design. The reader is cautioned that the generalizability of the results is not known. There are thousands of variations on genetic algorithms that have been suggested. The performance of both PSO and GA can depend on the parameters used. But, for the problem considered here, the adaptation of PSO was shown to work well in solving an agent-based model.

References

Alkemade, F., H. La Poutre, and H. Amman. 2006. "Robust Evolutionary Algorithm Design for Socio-Economic Simulation." *Computational Economics* 28:355-370.

Arifovic, J. 1994. "Genetic Algorithm Learning and the Cobweb-Model." *Journal of Economic Dynamics and Control* 18:3-28.

Arifovic, J. 1996. "The Behavior of the Exchange Rate in the Genetic Algorithm and Experimental Economies." *Journal of Political Economy* 104:510-41.

Arifovic, J., and M. Maschek. 2006. "Revisiting Individual Evolutionary Learning in the Cobweb Model – An Illustration of the Virtual Spite-Effect." *Computational Economics* 28:333–354

Axelrod, R. 1987. "The Evolution of Strategies in the Iterated Prisoner's Dilemma." *Genetic Algorithms and Simulated Annealing* pp. 32-41. London: Pitman.

Bullard, J., and J. Duffy. 1999. "Using Genetic Algorithms to Model the Evolution of Heterogeneous Beliefs." *Computational Economics* 13:41-60.

Chatterjee, A., and P. Siarry. "Nonlinear Inertia Weight Variation for Dynamic Adaptation in Particle Swarm Optimization." *Computers & Operations Research* 33:859–871.

Dawid, H. 1999. "On the Convergence of Genetic Learning in a Double Auction Market." *Journal of Economic Dynamics and Control* 23:1545-1567.

De Jong, K. A. 1975. "An Analysis of the Behavior of a Class of Genetic Adaptive Systems." Ph.D. Dissertation, University of Michigan, Ann Arbor, Mich.

Eberhart, R.C., and J. Kennedy. 1995. "A New Optimizer Using Particle Swarm Theory." *Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan*. pp. 39-43.

Erev, I., and A. Roth. 1998. "Predicting How People Play Games: Reinforcement Learning in Experimental Games with Unique Mixed Strategy Equilibria." *American Economic Review* 88:848–881.

Goldberg, D.E. 1989. “Genetic Algorithms in Search, Optimization, and Machine Learning.” Addison-Wesley, Massachusetts.

Hamm, L., B.W. Brorsen, and M.T. Hagan. 2007. “Comparison of Stochastic Global Optimization Methods to Estimate Neural Network Weights.” *Neural Processing Letters* 26: 145-158.

Hassan, R., B. Cohanim, O. De Weck, and G. Venter. 2005. “A Comparison of Particle Swarm Optimization and the Genetic Algorithm.” *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Austin, Texas, AIAA-2005-1897.

Kutschinski, E., T. Uthmann, and D.Polani. 2003. “Learning Competitive Pricing Strategies by Multi-Agent Reinforcement Learning.” *Journal of Economic Dynamics & Control* 27:2207-2218.

Panda, S. and N.P. Padhy. 2007. “Comparison of Particle Swarm Optimization and Genetic Algorithm for TCSC-Based Controller Design.” *International Journal of Computer Science and Engineering* 1:1-49.

Mouser, C.R., and S.A. Dunn. 2005. “Comparing Genetic Algorithms and Particle Swarm Optimisation for an Inverse Problem Exercise.” *Anizam Journal* 46:C89-C101.

Riechmann, T. 2001. “Genetic Algorithm Learning and Evolutionary Games.” *Journal of Economic Dynamics & Control* 25:1019-1037.

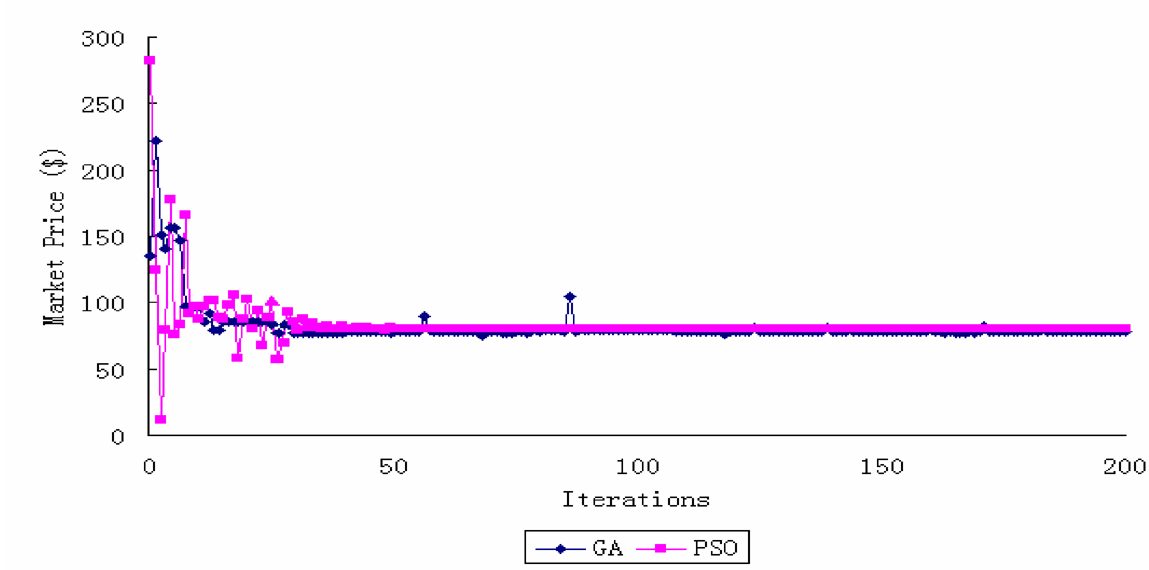
Vriend, J.N. 2000. “An Illustration of the Essential Difference between Individual and Social Learning, and Its Consequences for Computational Analyses.” *Journal of Economic Dynamics & Control* 24:1-19.

Step 1	Initialization {
	Every agent initializes the starting strategy population pool by randomly drawing S strategies;
	Assign monopoly profit as relative payoff.
	}

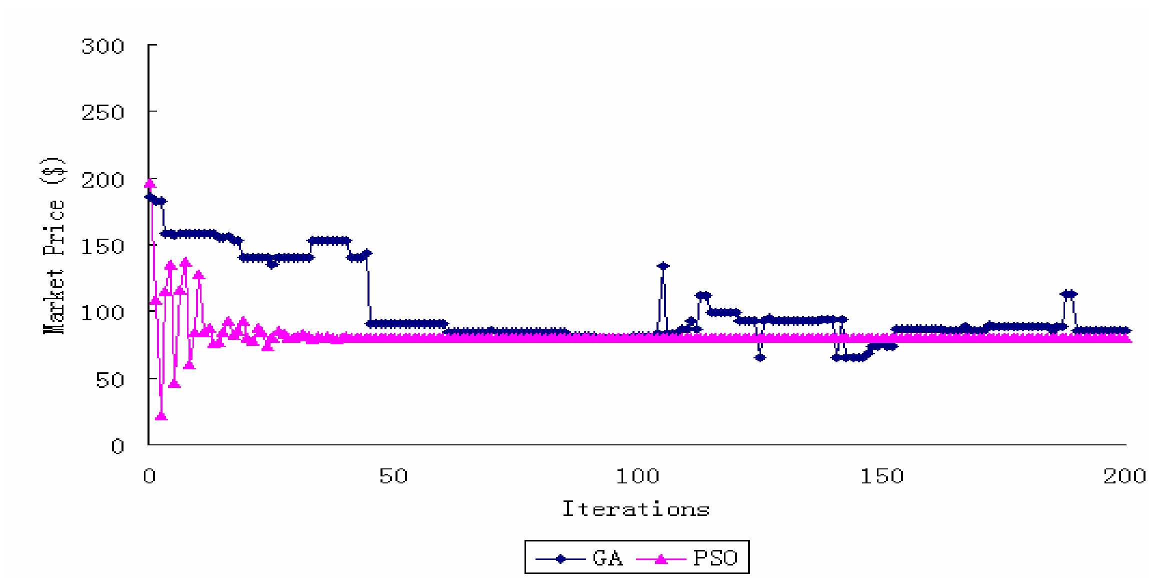
Step 2	For each generation do { For each agent {Randomly choose its active strategy from the population} Play the Cournot game; Calculate and store payoff for the current active strategy. } until all strategies are played.
Step 3	Update algorithm operators or parameters if needed;

Step 4	Generate new population from the old one{
	Choose εS strategies with highest profit as elites;
	Choose $(1-\varepsilon)S$ strategies with ranking selection methods;
	Randomly match all the selected ones as parents, apply single crossover to them with rate χ, until get $(1-\varepsilon)S$ number strategies, and apply mutation to them;
	These $(1-\varepsilon)S$ strategies combine with elites to form the new population.
	}
	
	If not converged, go back to step 2;
	End;

Figure I-1. Outline of the genetic algorithm (GA) for Cournot game



(a) With fixed algorithm parameters

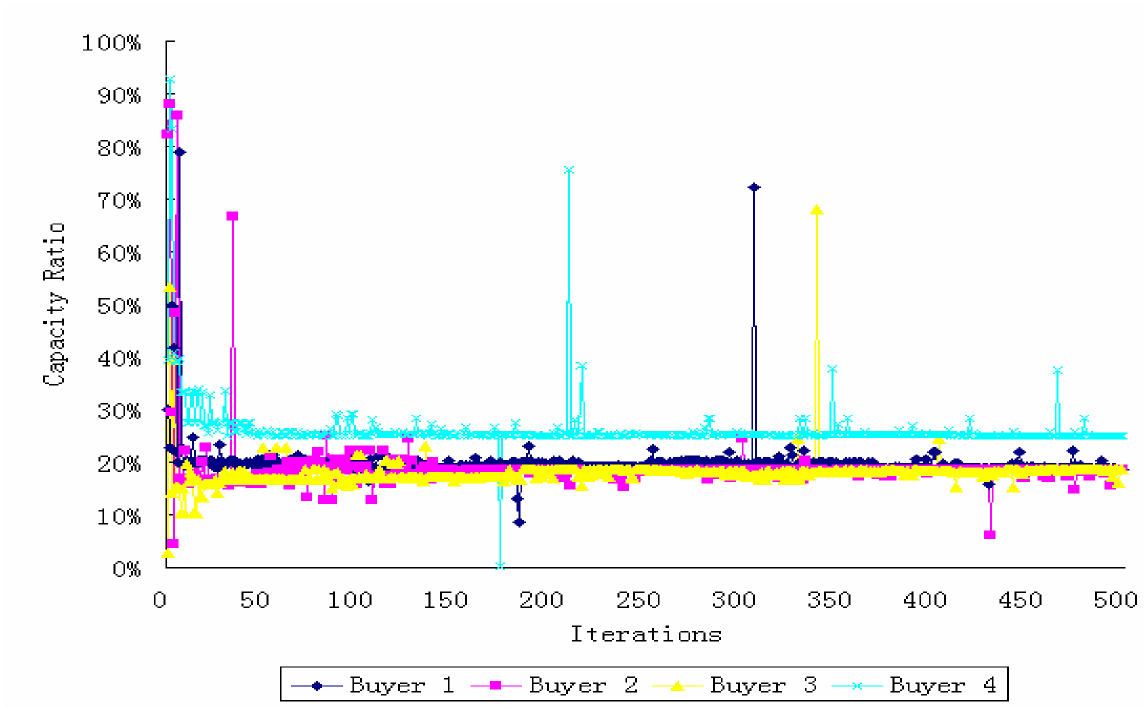


(b) With changing algorithm parameters

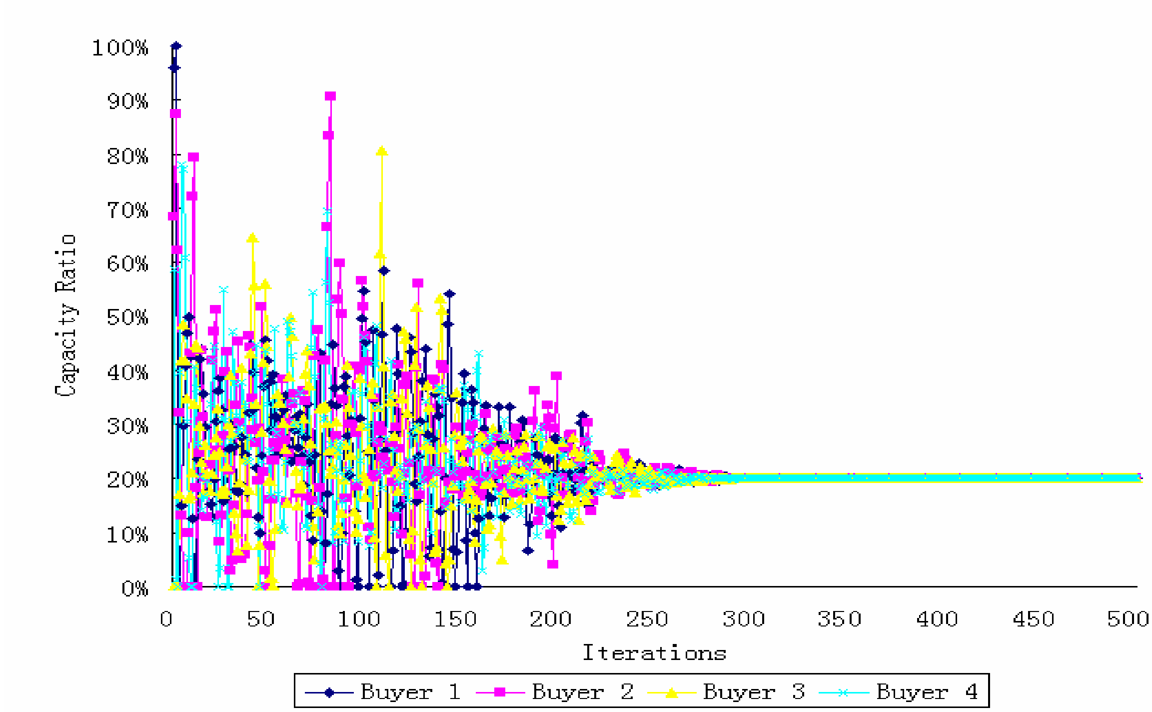
Figure I-2. Market price level

Note:

1. In (a), parameter set for PSO is $[w, c_1, c_2] = [\beta_0^w, \beta_0^{c_1}, \beta_0^{c_2}] = [0.4, 1, 1]$; parameter set for GA is $[\varepsilon, \chi, \mu] = [\beta_0^\varepsilon, \beta_0^\chi, \beta_0^\mu] = [10\%, 76\%, 0.33\%]$ and slopes in equations (1.11) to (1.13) are zeros.
2. In (b), parameter set for PSO is $[\beta_1^w, \beta_1^{c_1}] = [0.5, 1]$; parameter set for GA is $[\beta_1^\varepsilon, \beta_1^\chi, \beta_1^\mu] = [10\%.76\%, 0.33\%]$ and intercepts in equations (1.11) to (1.13) are zeros, $[\beta_0^\varepsilon, \beta_0^\chi, \beta_0^\mu] = [0, 0, 0]$.



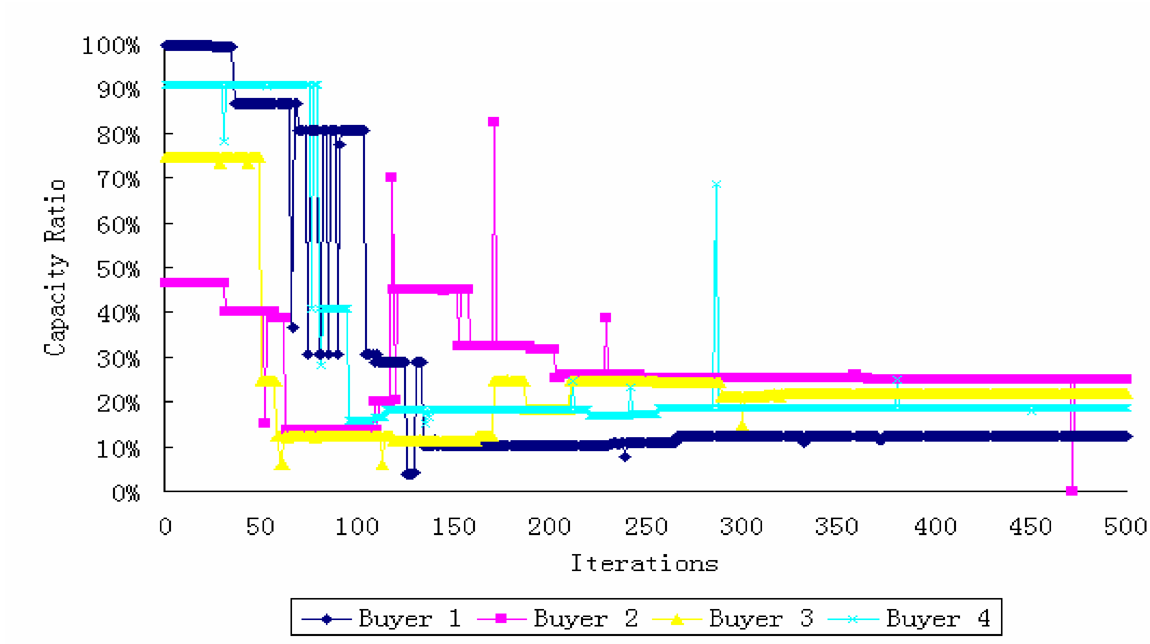
(a) GA



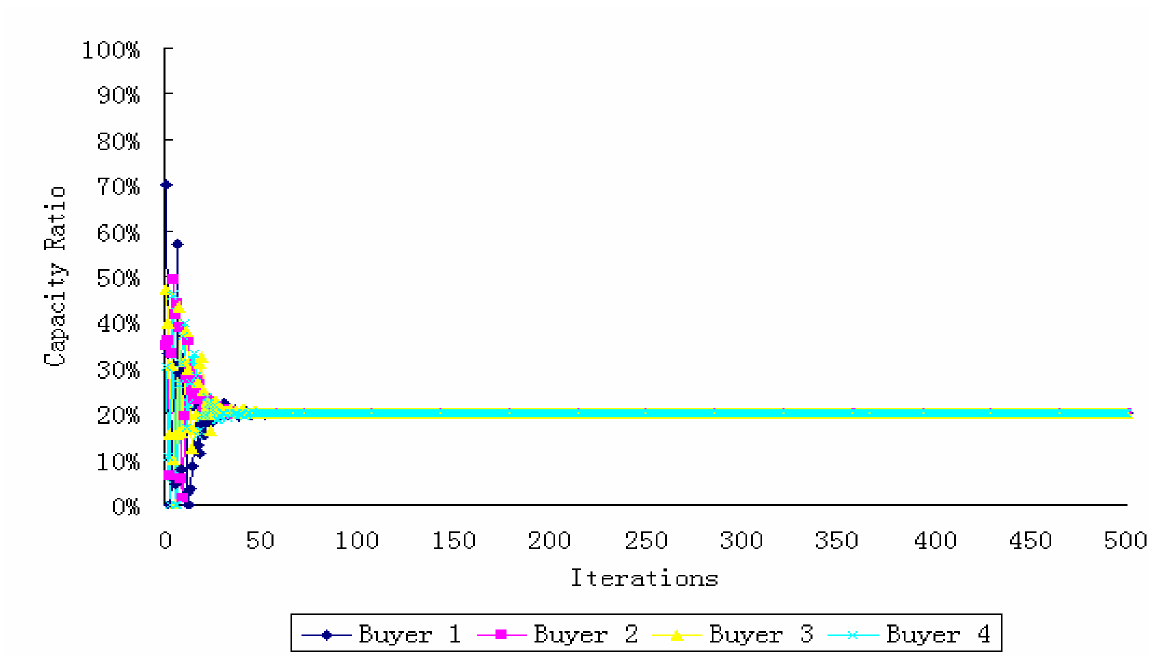
(b) PSO

Figure I-3. Quantity level with fixed algorithm parameters

Note: Parameter set for PSO is $[w, c_1, c_2] = [\beta_0^w, \beta_0^{c_1}, \beta_0^{c_2}] = [0.4, 1, 1]$; parameter set for GA is $[\varepsilon, \chi, \mu] = [\beta_0^\varepsilon, \beta_0^\chi, \beta_0^\mu] = [10\%, 76\%, 0.33\%]$ and slopes in equations (1.11) to (1.13) are zeros.



(a) GA



(b) PSO

Figure I-4. Quantity level with changing algorithm parameters

Note: Parameter set for PSO is [$\beta_1^w, \beta_1^{c_1}$] = [0.5, 1]; parameter set for GA is [$\beta_1^\varepsilon, \beta_1^\chi, \beta_1^\mu$] = [10%, 76%, 0.33%] and intercepts in equations (1.11) to (1.13) are zeros, [$\beta_0^\varepsilon, \beta_0^\chi, \beta_0^\mu$] = [0, 0, 0].

Table I-1. Parameters for PSO and GA in the Cournot Oligopsony Simulations

PSO parameters	**GA** parameters
Number of parallel markets: K	Strategy population size: K
Number of retest local best parameters: L	Loop per iteration: L
Inertia weight: $w_t = \beta_0^w + \beta_1^w \left(t_{\max} - t\right) / t_{\max}$	String bit: B
Local confidence factor: $c_{1,t} = \beta_0^{c_1} + \beta_1^{c_1} \left(t_{\max} - t\right) / t_{\max}$	Elitism rate: $\varepsilon_t = \beta_0^{\varepsilon} + \beta_1^{\varepsilon} \cdot t / t_{\max}$
Global confidence factor: $c_{2,t} = \beta_0^{c_2} + \beta_1^{c_2} \left(t_{\max} - t\right) / t_{\max}$	Crossover rate: $\chi_t = \beta_0^{\chi} + \beta_1^{\chi} \cdot \left(t_{\max} - t\right) / t_{\max}$
	Mutation rate: $\mu_t = \beta_0^{\mu} + \beta_1^{\mu} \left(t_{\max} - t\right) / t_{\max}$
	Ranking selection parameter: $r_{\max} = 1.1$, $r_{\min} = 0.9$.

Table I-2. PSO and GA Simulation Results with Fixed Algorithm Parameters

Set	Parameters			Statistic	Market Price($)	Capacity Ratio				Machine Time	Equilibrium Iteration
						Buyer 1	Buyer 2	Buyer 3	Buyer 4		
PSO	w	c_1	c_2								
1	0.4	1.5	1.5	Mean	81.82	21.32%	21.54%	20.30%	19.86%	294	N/A
				SD	1.16	0.49%	0.53%	0.43%	0.53%	32	
2	**0.4**	**1**	**1**	**Mean**	**80.00**	**20.00%**	**20.01%**	**19.99%**	**20.00%**	**230**	**66**
				SD	**0.00**	**0.01%**	**0.01%**	**0.00%**	**0.00%**	**27**	**6**
3	0.4	0.5	0.5	Mean	80.06	20.21%	21.32%	19.51%	19.10%	228	286
				SD	0.46	0.76%	0.93%	0.67%	1.10%	22	149
4	0.1	1	1	Mean	80.01	20.00%	20.09%	19.98%	19.95%	280	121
				SD	0.06	0.14%	0.18%	0.06%	0.08%	116	101
GA	ε	χ	μ								
5	10.00%	80.00%	1.00%	Mean	83.38	19.53%	25.53%	19.33%	18.98%	4,780	N/A
				SD	2.23	3.75%	3.35%	3.91%	3.93%	719	
6	0.00%	80.00%	0.33%	Mean	79.58	12.32%	20.77%	25.33%	21.13%	5,401	N/A
				SD	1.65	3.28%	2.68%	4.57%	2.93%	2,081	
7	10.00%	76.00%	0.33%	Mean	79.58	18.70%	21.92%	20.23%	18.70%	5,539	398
				SD	1.76	3.27%	3.03%	2.60%	3.40%	1,051	9
8	30.00%	56.00%	0.33%	Mean	78.17	12.50%	25.01%	18.75%	21.91%	4,979	252
				SD	2.50	7.83%	4.25%	4.42%	3.47%	1,146	104

Note: For PSO, the parallel market size for PSO is 20. For GA, the population size is 40, the bit length is 15.

Table I-3. PSO and GA Simulation Results with Changing Algorithm Parameters

Set	Parameters			Statistic	Market Price($)	Capacity Ratio				Machine Time	Equilibrium Iteration
						Buyer 1	Buyer 2	Buyer 3	Buyer 4		
PSO	β_1^w	$\beta_1^{c_1}$									
9	0.98	2.5		Mean	80.01	19.99%	20.00%	20.00%	20.04%	290	150
				SD	0.02	0.01%	0.02%	0.03%	0.02%	66	7
10	0.98	0.5		Mean	80.00	19.99%	20.00%	20.02%	20.00%	255	88
				SD	0.01	0.01%	0.02%	0.02%	0.02%	20	9
11	**0.5**	**1**		**Mean**	**80.00**	**20.00%**	**20.00%**	**20.00%**	**20.00%**	**324**	**59**
				SD	**0.00**	**0.02%**	**0.01%**	**0.03%**	**0.01%**	**159**	**14**
12	0.2	1		Mean	80.02	20.32%	19.99%	19.85%	19.87%	241	80
				SD	0.10	0.24%	0.27%	0.28%	0.24%	30	34
GA	β_1^ε	β_1^χ	β_1^μ								
13	10.00%	86.00%	0.33%	Mean	81.29	18.95%	18.55%	25.02%	18.76%	3,854	170
				SD	1.33	3.12%	3.12%	3.38%	3.79%	305	22
14	**20.00%**	**66.00%**	**1.00%**	**Mean**	**80.18**	**20.01%**	**18.75%**	**19.53%**	**21.88%**	**4,037**	**188**
				SD	**0.83**	**2.18%**	**2.28%**	**2.43%**	**2.56%**	**356**	**11**
15	40.00%	76.00%	1.00%	Mean	79.95	19.25%	18.55%	17.06%	25.08%	4,650	96
				SD	1.37	2.34%	3.75%	3.97%	2.90%	998	8
16	40.00%	66.00%	0.33%	Mean	80.07	20.33%	20.32%	19.93%	19.48%	4,348	154
				SD	0.62	1.87%	2.26%	2.65%	1.12%	595	20

Note:

1. For PSO, the parallel market size for PSO is 20. For GA, the population size is 40 and the bit length is 15.
2. For PSO, the intercepts β_0^w and $\beta_0^{c_1}$ in equations (1.6) and (1.7) are chosen as constant value 0.5 and 1 respectively.
3. For GA, the intercepts β_0^ε , β_0^χ and β_0^μ in equations (1.11), (1.12) and (1.13) are zero.

Table I-4. PSO and GA Simulations Results under Different Algorithm Structure

Set	*P*	*L*	Statistic	Market Price($)	Capacity Ratio				Machine Time	Equilibrium Iteration
					Buyer 1	Buyer 2	Buyer 3	Buyer 4		
PSO										
17	20	10	Mean	79.99	20.00%	19.99%	20.03%	19.98%	434	60
			SD	0.02	0.01%	0.04%	0.01%	0.01%	39	11
18	10	3	Mean	79.99	19.97%	19.97%	20.04%	20.02%	122	76
			SD	0.20	0.28%	0.38%	0.98%	0.23%	11	26
19	3	3	Mean	79.77	20.24%	20.22%	19.10%	20.22%	45	80
			SD	2.31	4.69%	5.04%	4.88%	1.93%	15	45
GA										
20	100	100	Mean	79.70	20.30%	20.32%	18.74%	20.31%	35,805	172
			SD	0.76	1.91%	1.40%	1.14%	1.86%	2,732	18
21	40	40	Mean	78.93	21.09%	20.31%	25.00%	12.49%	4,131	150
			SD	1.24	2.90%	2.84%	3.43%	2.35%	568	26
22	20	40	Mean	78.01	21.32%	12.49%	21.97%	22.20%	1,838	121
			SD	1.73	5.13%	3.67%	3.62%	4.21%	186	23

Note: *P* indicates parallel markets number and population size for PSO and GA respectively, *L* indicates number of retest local best of PSO and loop number per generation of GA respectively.

CHAPTER II

COLLUSION AND COMPETITION OF OLIGOPSONY FIRMS WITH QUANTITY-PRICE STRATEGIC DECISIONS: AN AGENT-BASED ARTIFICIAL MARKET

Introduction

The classical Bertrand model states that for price setting firms, the Nash equilibrium is the perfect competition level and the solution does not depend on the number of firms. Edgeworth and his followers show that a capacity constraint can bind the price away from the competitive level (Levitan and Shubik 1972; Tasnadi 1999). Kreps and Scheinkman (1983) propose a theoretical model and find that if firms decide quantity first then compete with price these firms perform like Cournot competition.

In contrast to theoretical work, experiments with human subjects show prices deviating from perfect competition or even colluding to the monopoly level in duopoly and triopoly markets with or without a capacity constraints (Dufwenberg and Gneezy 2000; Brandts and Guillen 2007). Brandts and Guillen design experiments by letting firms choose price and quantity simultaneously. For both duopoly and triopoly markets, Brandts and Guillen find that either competitors learn to collude at the monopoly level or only one firm survives, which results in a monopoly market. But since they assume the demand curve is “boxed-shaped” in that consumers will purchase products up to the maximum quantity if price is less than a fixed value, their results cannot differentiate monopoly from Cournot. Suetens and Potters (2007) also find significantly more tacit collusion in Bertrand than in Cournot markets after reexamining experimental results of four previous studies (Fouraker and Siegel 1963; Huck, Normann, and Oechssler 2000; Davis 2002; Altavilla, Luini, and Sbriglia 2006). Huck, Normann and Oechssler (2000) find that experimental markets with more than three competitors are more likely to be competitive than oligopolistic.

The Nash equilibrium of the Bertrand model assumes agents are rational. The concept of bounded rationality revises this assumption to account for the fact that perfectly rational decisions are often not feasible in practice due to the finite computational resources available for making them (Simon 1991). The departure of experimental results from the theoretical model implies that a specific learning algorithm needs to be employed to describe how agents make economic decisions. Lucas (1986) suggests that comparing the results of adaptive learning algorithms and experiments with human subjects may illustrate how people learn in a specific problem.

To help explain the gap between the Bertrand theoretical predictions and experimental results, this research studies behavior of quantity-price competition firms with an agent-based computational model. Past research with agent-based models in economics uses genetic algorithms (GA) (Ariforic 1994 and 1996; Axelrod 1987; Vriend 2000) and reinforcement leaning algorithm (RL) (Erev and Roth 1998; Kutschinski, Uthmann and Polani 2003). With GA, researchers have to be very careful to choose parameters and methods for each problem or it may cause premature convergence. The large population size required also makes GA slow to find equilibrium. RL is a sub-area of machine learning and the environment is typically formulated as a finite-state Markov decision process in which an agent increases the probability of choosing successful strategies under the possible strategy spaces of its rivals. When the possible strategy space is large or continuous, the computational cost increases exponentially.

To avoid the limitation of the above two algorithms, this study adopts the particle swarm optimization (PSO) algorithm to model the learning behavior of agents. PSO is a stochastic optimization technique developed by Eberhart and Kennedy (1995) and can be used for an economic optimization problem. Equilibrium is found for markets with one to four firms. Each firm operates in a set of parallel markets and each firm uses PSO to solve its own optimization problem.

Description of Oligopsony Model

A processing industry is used as an example. In a product supply chain, processing firms purchase input from many relatively small sellers and sell processed goods to a big retail market. Buyers have to decide both production capacity and procurement price strategy simultaneously in advance. Once their procurement prices are announced, sellers will choose to sell products to the current highest bidder. Buyers need to arrange other inputs before processing and the capacity is determined in advance and unchangeable in one production period, so the buyer will stop purchasing activity at the capacity point even if it can get more. Each seller has an upward sloping supply curve. We assume the demand curve is perfectly elastic so we can focus on the buyers' trading behavior.

Consider a homogeneous product market with M buyers and N sellers. The number of buyers is much less than the number of sellers ($M << N$). Assume that buyers process products that will be sold in the retail market and the marginal cost for processing is the same for all processors. The marginal value equals the selling price minus the marginal processing cost. To focus on the games between buyers and sellers in this market, the final product price P and the marginal processing cost mc are constant and as a result the before processing value, $R = P - mc$, is also constant. There is a capacity cost and the cost is a constant value c^C for one additional unit of capacity.

Assume all sellers are homogeneous and have the linear supply function $q_j^s = b$. This means that the sellers sell their product to the current highest biding buyer with bid b and provide quantity q_j^s. If buyers are perfectly competitive, they will all bid $b = R - c^C$ for each product and get zero profit. Therefore $R - c^C$ is also the supply quantity of each seller under buyers' perfect competition condition.

At the beginning of each processing period, buyers make combination strategies $\mathbf{x} = [x^b, x^C]'$ simultaneously. The bids and prices are discrete and use cents as the minimum unit. The highest bidder gets the supply first up to its capacity. Then the next highest bidder makes the procurement and so on. If more than one buyer bids the highest price, they split the supply quantity until their capacity. For comparison convenience, we use price ratio and capacity ratio to indicate the buyers' strategies. The price ratio of each buyer is

$$x^b = b/(R - c^C), \tag{2.1}$$

where x^b is defined as the bid price ratio, b is bid price, and R is the buyer's marginal revenue. In some cases, buyers spend capacity cost to build a building, hire people, or arrange other materials in advance to meet the processing requirement besides *mc*. The marginal capacity cost is c^C. For example, since the revenue *R* for one product is \$100, if the capacity cost c^C for one product is \$10, then the value of the product before processing is $R - c^C = \$90$. Thus, when a buyer plans to use a bid price ratio of 50% as its pricing strategy, it will bid \$45 for the product in the market. Besides price strategy, buyers have to make a capacity choice at the same time. The processor's capacity ratio is

$$x^C = Q^C / [(R - c^C) \times N], \tag{2.2}$$

where x^C is the capacity ratio, Q^C is the processing capacity of the buyer, and N is the total number of sellers. Buyers choose capacity ratio to plan processing quantity. For example, if the capacity cost if \$10, with total sellers number *N* = 400 and *R* = \$100, the sellers will provide 36,000 under buyers' perfectly competitive condition. Thus, if one buyer plans to process 3,600 products, its processing strategy is to use 10% as its capacity ratio.

Simulation Design with Particle Swarm Optimization Algorithm

The idea of PSO came from watching the way flocks of birds, fish or other animals adapt to avoid predators and find food by sharing information. The difference here is that each buyer has a separate "flock of birds" that does not share information with the flocks of the other agents.

We set up *K* parallel markets and letting the agents each have their own clones in every market. Although having the same behavior rules, one agent and its *K* clones may take different market strategies since the initialized random values are different. In the simulation, buyers dynamically change their marketing strategies with the PSO algorithm but sellers are price takers and simply sell their products to the current highest bidders.

Considering agents continuously changing their strategies, the local best solutions may not be the best for the current period. Thus, we adjust the PSO by retesting the past best locals of each clone under current market environment and choose the best fit as the current best local among these best locals and the current strategy.

Particle Swarm Optimization Algorithm

First we describe the generic particle swarm optimization algorithm. Table II-1 shows the pseudo code of the PSO algorithm. Every clone has a strategy parameter set $\mathbf{x}$ which is randomly initialized at the beginning of the simulation. Each strategy in the set has an adjustment velocity, $v_{i,k,t}^{\Gamma} \in [-1,+1]$, which determines the change of the choice variable, the superscript Γ indicates bid price or capacity strategy in the decision set. The velocity change of a strategy parameter for a clone is a function of the local best solutions achieved in its own market, $p_{i,k,t}^{\Gamma,l} \in [0,1]$, and its global best solution among all the parallel markets, $p_{i,t}^{\Gamma,g} \in [0,1]$. The superscripts l and g indicate local and global, the subscripts k and i indicate the k^{th} parallel market and i^{th} agent respectively. The profit function $\pi_k(\mathbf{x}_{i,k,t})$ is used to value the performance of the choice variable set $\mathbf{x}_{i,k,t}$.

In every simulation step, each new choice variable of the i^{th} agent in the k^{th} parallel market can be updated by the following equation:

(2.3) $$x_{i,k,t+1}^{\Gamma} = x_{i,k,t}^{\Gamma} + v_{i,k,t}^{\Gamma},$$

and the velocity is modeled as:

(2.4) $$v_{i,k,t+1}^{\Gamma} = wv_{i,k,t+1}^{\Gamma} + c_1u_1(p_{i,k,t}^{\Gamma,l} - x_{i,k,t}^{\Gamma}) + c_2u_2(p_{i,t}^{\Gamma,g} - x_{i,k,t}^{\Gamma}),$$

where $v_{i,k,t}^{\Gamma}$ is the velocity, $u_j \in [0,1]$, $j = 1,2$ are uniformly distributed random numbers, c_1 and c_2 are learning parameters and are called self confidence factor and swarm confidence factor respectively, and w is an inertia weight factor.

The following equations describe how to choose an agent's local best and global best. As stated in the introduction, we adapt the PSO by retesting the best locals. In our research, the new best local is chosen from the best locals of the previous L periods and the strategy $\mathbf{x}_{i,k,t}$ of the current period:

(2.5) $$\mathbf{p}_{i,k,t}^{l} = \arg\max\left\{\pi_k(\mathbf{p}_{i,k,t-1}^{l}),......,\pi_k(\mathbf{p}_{i,k,t-L}^{l}),\pi_k(\mathbf{x}_{i,k,t})\middle|\mathbf{x}_{i'\neq i,k,t}\right\},$$

where i' indicates opponents, π is profit. The different past best locals of each agent in L periods will be reevaluated under the current t period economic environment by holding other agents' strategies in this period unchanged. The profits will be compared with that of that of the current strategy. The one with the highest profit is the new local best. Just like in real economic markets, the market information will be revealed to participants. The best global parameter is selected from the best locals:

(2.6) $$\mathbf{p}_{i,t}^{g} = \arg\max\left\{\pi_1(\mathbf{p}_{i,1,t}^{l}),\pi_2(\mathbf{p}_{i,2,t}^{l}),\ldots,\pi_K(\mathbf{p}_{i,K,t}^{l})\right\}.$$

Chatterjee and Siarry (2006) state that the inertia weight w in (2.3) is critical for the PSO's convergence behavior. A large inertia weight provides larger exploration than a smaller one. So it is worth making a compromise and letting w start with a higher value at the beginning and then decreasing w as the optimization proceeds:

(2.7) $$w_t = \beta_0^w + \beta_1^w\left(t_{\max} - t\right)/t_{\max},$$

where both β_0^w and β_1^w are constants, $t_{\max}$ is the maximum number of iterations and t is the current iteration. Similarly, we set c_1 and c_2 in equation (2.4) as:

(2.8) $$c_{1,t} = c_{2,t} = \beta_0^{c_1} + \beta_1^{c_1}\left(t_{\max} - t\right)/t_{\max},$$

where both $\beta_0^{c_1}$ and $\beta_1^{c_1}$ are constants.

This research also studies an oligopsony market where agents make combinatorial decisions on both capacity and price, so buyers have a combinatorial strategy set instead of one choice variable. Each clone of a buyer chooses a capacity-price ratio set $\mathbf{x} = [x^b, x^C]'$ as a combinatorial strategy in each parallel market, $x^\Gamma \in [0,1]$, $\Gamma = b, C$.

Equilibrium Criterion

Zero diversity in the strategies of all parallel markets for every agent can be used to signal the stopping point for PSO. Diversity diminishes with time which causes the same strategy to dominate among all parallel markets. In our simulation, for all agents, if the variance of the strategies in the population is less than 0.01% and the mean value of the strategies for 10 iterations is less than 0.01%, we say the algorithm is converged.

Summary of Simulation Procedure

We test three oligopsony markets (duopsony, triopsony and 4-buyer market) as well as a monopsony market. For each market structure, two capacity cost scenarios (zero, $30) are considered. Buyers choose bid price or quantity and bid for product simultaneously, and improve the combinatorial strategy set with PSO at the end of each period.

We design the artificial markets with PSO by setting up K = 20 parallel markets and buyers and their clones trade in all markets simultaneously and independently. The retest iteration number L is chosen as 10. The number N of sellers is 400 among all experiment settings. The simulation steps are described as follows.

(i) For the first L beginning iterations, randomly initialize strategy set $\mathbf{x}$ for all buyers in every parallel market. We choose the quantity and price ratio $x_{i,k,t}^{\Gamma} \in U[0,1]$ and the movement velocities $v_{i,k,t} = 0$ for $i = 1,...,M$, $\Gamma = 1,2$, $k = 1,...,K$, and $t = 1,...,L$.

(ii) Buyers update their capacity ratio and price ratio respectively with equations (2.3) and (2.4).

(iii) Within each parallel market, supply first goes to the current highest bidder up to the buyers' capacity. If more than one buyer bids the same price, then a sharing rule is assumed. Then the remaining supply goes to the second highest bidder up to its capacity and so on.

(iv) After the first L iterations, each buyer retest the past L best locals under current economic environment and compare their performance with that of the current strategy, the best among them is selected as the new best local, as equation (2.5) shows.

(v) Following equation (2.6), the best fit among all best locals is the best global.

(vi) If the market does not reach equilibrium, go to step (ii).

Simulation Settings

We simulate markets with and without a capacity cost. Under each capacity cost scenario we simulate markets with different numbers of buyers, which are monopsony, duopsony, triopsony, and a 4-buyer market and there are 8 alternative settings in total. The market and algorithm parameters used in the simulations are listed in Table II-1.

One simulation round contains multiple iterations in which agents trade with each other repeatedly. Within each round, agents play the game repeatedly and learn to find the best response strategy set until the market reaches equilibrium or meets the maximum 2000 iterations constraint. Considering the randomness of the learning path, each setting is run 100 rounds with different randomly initialized starting strategies for all agents. Once equilibrium is reached, the average value of the last 20 iterations for the market price and the agent's strategies are used as the market equilibrium values. The mean and standard deviation of the market equilibrium values of the 100 rounds are used to characterize market equilibrium. Theoretically, the price of monopsony and perfect competition levels are $(R-c^C)/2$ and $R-c^C$ respectively. The monopsony and perfect competition price are \$50 and \$100 when there is no capacity cost; and are \$35 and \$70 with the \$30 capacity cost.

Simulation Results

The mean and standard deviation of the equilibrium strategy parameters for each scenario are shown in Tables II-2 and II-3. Actual quantity ratio is defined as $q/[(R-c^C)\times N]$, here q is how many products it actual purchased with its price-quantity combinatorial strategy set in the market.

Table II-2 shows that without capacity cost, market prices are at the monopsony level always for markets with1 or 2 buyers, mostly for 3 buyers, and almost never for 4 buyers. From monopsony to triopsony, the more buyers in the market, the higher the market price is but only by a small amount. The market average price of $53.29 for the triopsony market is only a little higher than $50.07 for duopsony market. Buyers fight to nearly perfect competition level when the market contains 4 buyers with market price $99.98. Without capacity cost, buyers tend to maintain a capacity much higher than the actual market aggregate supply from monopsony to triopsony, and near to the actual market aggregate supply for the 4-buyer market. The actual quantity variances for one buyer are around 17% in triopsony market and much higher than around 5% in duopsony and around 3% in 4-buyer markets.

Table II-3 shows that when there is a capacity cost, the duopsony buyers still collude to nearly the monopsony level but the triopsony buyers compete to near the perfectly competitive level. The average market price of $36.63 in the duopsony market is slightly higher than the monopsony price of $34.99 and the market price of $69.41 in the 4-buyer market is a little higher than the triopsony market price of $68.88. With capacity cost, buyers make capacity plans more carefully. Especially for the monopsony market, the buyer plans its capacity to exactly match the quantity it actually gets in the market. For markets containing more than one buyer, excess capacity still exists but is smaller for duopsony and 4-buyer markets than the triopsony market.

Table II-4 presents the market price in equilibria under different experimental settings. Under market equilibrium, if the actual quantity ratio that a buyer gets is less than 5%, we say it is inactive in the market and the other active players share the supplies. Then we calculate the percentage of each category out of the equilibrium of 100 simulation rounds. The results in Table II-4 show that capacity cost causes the markets to have more varied equilibria than without capacity cost and the profit decreases considerably because of the excess capacity of processors. The duopsony and 4-buyer markets continue to collude and compete, but the competition in the triopsony market becomes more severe and the players compete to a higher price level or even the perfect competition price. Without capacity cost, the duopsony buyers always collude and the buyers in the 4-buyer market always compete to the perfectly competitive level and the equilibrium for the two markets are stable. The triopsony buyers collude to the monopsony price level with mostly 2 players active in the market, with a small chance of having only one player left. When there is a capacity cost, the multiple equilibria phenomena exist for all three markets. For the duopsony market, though buyers still mostly collude, a few times they compete to a higher market price level and sometimes there is only one buyer active in the market. The competition in the 4-buyer market with capacity cost is less severe than without capacity cost. In the 4-buyer market, there is a small chance that one or two players do not get any products and the remaining buyers share the supply. The capacity cost in the triopsony market causes the players in it to compete to a higher price level or even the perfectly competitive level instead of collude to the monopsony level as they usually do without capacity cost.

Figure II-1 shows the frequency of equilibrium market prices out of the 100 simulation runs for each market setting. According to the market design, with or without a $30 capacity cost, the monopsony market price levels are $50 and $35, and perfect competition price levels are $100 and $70 respectively. The results show that for the duopsony market without capacity cost, the market price is mostly exactly the monopsony level or a little bit higher. With capacity cost, the equilibrium price of the duopsony market mostly locates slightly higher than the monopsony level of $35 and scatters larger than without cost. For the triopsony market, market equilibrium prices are tend to be a few dollars higher than the monopsony level without a capacity cost but tend to the perfect competition level with capacity cost.

Figure II-2 shows the frequency of the buyers' actual capacity ratios in each market setting. We put the actual capacity ratio of all buyers at equilibrium for all 100 simulation runs together, separate them into different ranges and draw their distribution. For example, for the results of 100 simulation runs of the triopsony market, there are 300 equilibrium actual capacity ratios total for all three players. If there are 30 ratios in the range 22.5%-27.5%, we say the frequency of this range is 10% among all possible equilibrium actual capacity ratios.

Figure II-2 (a), (e) and (f) show that for the duopsony market without capacity cost and the two 4-buyer market settings, there is mainly one equilibrium for each market and buyers tend to split the supply. For the duopsony market with a capacity cost, as Figure II-2(b) shows, about 8% of the time one buyer does not purchase any product and the remaining buyer dominates, but most of the time the buyers split the market share. For all three markets, the actual capacity ranges scatter larger with capacity cost than without cost.

Figure II-3 shows individual runs to illustrate the pricing behavior evolution of buyers with no capacity cost setting. This figure shows that monopsony and 4 buyers markets reach equilibrium faster than duopsony and triopsony markets. In this example, all buyers have the same pricing strategy. When the number of buyers is less than 4, they collude to monopsony and otherwise they compete to the perfectly competitive level.

This phenomenon shows that when there are more than 3 players in the price competition market, they have little chance to collude and will compete to perfect competition level.

Conclusions

This research studied the procurement behavior of oligopsony buyers with capacity-price combination decisions. When buyers can learn from their past performance and make strategic decisions with some randomness, buyers collude to nearly the monopsony level for duopsony market and compete to nearly the perfect competitive level when the market contains four buyers. For triopsony, the equilibrium price level increases from monopsony to the near perfectly competitive level when a capacity cost is added. Huck, Normann, and Oechssler (2004) obtain similar results with quantity competition experiments with human subjects. They also find that the number of buyers affects the competitive behavior and two buyers tend to collude and four buyers tend to be competitive. The results also show that buyers tend to have excess capacity. This phenomenon can be observed in the beef packing industry, where the processing rate is only around 70% of the buyers' total capacity.

The results show that computer-based adaptive algorithms can mimic how firms adapt their behavior under strategic price-quantity competition markets and present a potential explanation of the phenomena observed in experiments with human subjects which differ from the predictions of or cannot be explained by existing theoretical models with fully rational agents. This suggests that people use heuristic learning rules, which give different outcomes than the profit maximization rules assumed in most past theoretical work.

References

Altavilla, C., L. Luini, and P. Sbriglia. 2006. "Social Learning in Market Games." *Journal of Economic Behavior and Organization* 61: 632-652.

Arifovic, J. 1994. "Genetic Algorithm Learning and the Cobweb-Model." *Journal of Economic Dynamics and Control* 18:3-28.

Arifovic, J. 1996. "The Behavior of the Exchange Rate in the Genetic Algorithm and Experimental Economies." *Journal of Political Economy* 104:510-41.

Arifovic, J., and M. Maschek. 2006. "Revisiting Individual Evolutionary Learning in the Cobweb Model – An Illustration of the Virtual Spite-Effect." *Computational Economics* 28:333–354

Axelrod, R. 1987. "The Evolution of Strategies in the Iterated Prisoner's Dilemma." In *Genetic Algorithms and Simulated Annealing*, Pitman, London, pp. 32-41.

Brandts, J., and Guillen, P., 2007. "Collusion and Fights in An Experiment with Price-Setting Firms and Advance Production." *Journal of Industrial Economics* 55:453-474.

Chatterjee, A., and P. Siarry. "Nonlinear Inertia Weight Variation for Dynamic Adaptation in Particle Swarm Optimization." *Computers & Operations Research* 33:859–871.

Davis, D. D. 2002. "Strategic Interactions, Market Information and Predicting the Effects of Mergers in Differentiated Product Markets." *International Journal of Industrial Organization* 20:1277–1312.

Dufwenberg, M., and U. Gneezy. 2000. "Price Competition and Market Concentration: An Experimental Study." *International Journal of Industrial Organization* 18:7–22.

Eberhart, R.C., and J. Kennedy. 1995. "A New Optimizer Using Particle Swarm Theory." *Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan*. pp. 39-43.

Erev, I., and A. Roth. 1998. “Predicting How People Play Games: Reinforcement Learning in Experimental Games with Unique Mixed Strategy Equilibria.” *American Economic Review* 88:848–881.

Fouraker, L.E., and S. Siegel. 1963. *Bargaining Behavior*. McGraw-Hill, London.

Huck, S., H. Normann, and J. Oechssler. 2004. “Two Are Few and Four Are Many: Number Effects in Experimental Oligopolies.” *Journal of Economic Behavior & Organization* 53:435-446.

Kreps, D. M., J.A. Scheinkman. 1983. “Quantity Recommitment and Bertrand Competition Yield Cournot Outcomes.” *The Bell Journal of Economics* 14:326–337.

Kutschinski, E., T. Uthmann., and D. Polani. 2003. “Learning Competitive Pricing Strategies by Multi-agent Reinforcement Learning.” *Journal of Economic Dynamics & Control* 27:2207-2218.

Levitan, R., and M. Shubik. 1972. “Price Duopoly and Capacity Constraints.” *International Economic Review* 13:111–122.

Lucas, R.E., Jr., 1986. “Adaptive Behavior and Economic Theory.” *Journal of Business* 59:401-426.

Simon, H.A. 1991. “Bounded Rationality and Organizational Learning.” *Organization Science* 2:125-134.

Suetens, S., and J. Potters. 2007. “Bertrand Colludes More than Cournot.” *Experimental Economics* 10:71-77.

Tasnadi, A., 1999. “A Two-Stage Bertrand-Edgeworth Game.” *Economics Letters* 65:353-358.

Vriend, J.N. 2000. “An Illustration of the Essential Difference between Individual and Social Learning, and Its Consequences for Computational Analyses.” *Journal of Economic Dynamics & Control* 24:1-19.

Table II-1. PSO Parameters in the Artificial Market Simulation

Parameter	Symbol	Value
Market Parameters		
Number of firms	M	1 for monopsony market; 2 for duopsony market; 3 for triopsony market; 4 for 4-packer market
Number of sellers	N	400
Product value	R	\$100
Particle Swarm Optimization (PSO) Algorithm Parameters		
Intercept of inertia weight in equation (2.7) of PSO	β_0^w	1.5
Slope of inertia weight in equation (2.7) of PSO	β_1^w	0.5
Self and global confidence factors of PSO	$c_1 = c_2$	1
Number of parallel market	K	20
Maximum iteration of one simulation round	$t_{\max}$	2,000
Number of simulation round		100

Table II-2. Simulation Results of Price-Quantity Strategic Buyers without Capacity Cost

Market Structure	Buyer	Statistic	Market Price($)	Price Ratio	Capacity Ratio	Actual Quantity Ratio	Profit($)
Monopsony	Buyer 1	Mean	50.00	50.00%	83.31%	50.00%	249,990
		SD	0.00	0.00%	14.20%	0.02%	89
Duopsony	Buyer 1	Mean	50.07	50.06%	84.71%	25.53%	127,442
		SD	0.07	0.07%	9.96%	5.53%	27,550
	Buyer 2	Mean		50.06%	84.40%	24.54%	122,523
		SD		0.07%	9.41%	5.52%	27,543
Triopsony	Buyer 1	Mean	53.59	53.55%	85.49%	16.36%	75,762
		SD	1.24	1.24%	14.30%	17.17%	79,333
	Buyer 2	Mean		53.57%	87.47%	20.45%	94,954
		SD		1.24%	12.77%	17.01%	79,138
	Buyer 3	Mean		53.56%	84.78%	16.78%	77,843
		SD		1.24%	13.38%	15.29%	70,998
Four-Buyer	Buyer 1	Mean	99.98	99.98%	96.31%	25.42%	39
		SD	0.00	0.00%	3.70%	3.05%	10
	Buyer 2	Mean		99.98%	95.62%	25.16%	40
		SD		0.00%	5.39%	3.12%	10
	Buyer 3	Mean		99.98%	95.82%	24.43%	39
		SD		0.00%	4.76%	3.14%	10
	Buyer 4	Mean		99.98%	96.24%	24.92%	37
		SD		0.00%	4.19%	3.21%	10

Note:

1. Marginal capacity cost is zero, which is $c^C = 0$.
2. Price ratio is the bidding price of a buyer relative to $R - c^C$ = \$100 here; for example, 50% price ratio means this buyer will bid with \$50.
3. Capacity ratio is the processing quantity plan of a buyer relative to $(R - c^C) \times N$ = 10,000; for example, buyer 1's capacity ratio is 83.30%, and actual capacity ratio is 49.98% means it plans to purchase 8,330 products but actually get 4,998 in the market.

Table II-3. Simulation Results of Price-Quantity Strategic Buyers with Capacity Cost

Market Structure	Buyer	Statistic	Market Price($)	Price Ratio	Capacity Ratio	Actual Quantity Ratio	Profit($)
Monopsony	Buyer 1	Mean	34.99	49.99%	49.99%	49.99%	122,483
		SD	0.02	0.03%	0.03%	0.03%	17
Duopsony	Buyer 1	Mean	36.63	52.30%	52.05%	24.78%	565
		SD	0.21	0.30%	0.82%	13.00%	56,911
	Buyer 2	Mean		52.31%	51.88%	27.53%	13,172
		SD		0.29%	2.64%	12.98%	57,366
Triopsony	Buyer 1	Mean	68.88	98.36%	85.99%	32.16%	-110,629
		SD	0.95	1.34%	23.55%	29.35%	68,434
	Buyer 2	Mean		98.38%	84.14%	32.18%	-106,707
		SD		1.35%	23.75%	24.67%	62,054
	Buyer 3	Mean		98.37%	86.89%	33.79%	-108,781
		SD		1.35%	21.14%	26.82%	60,707
Four-Buyer	Buyer 1	Mean	69.41	97.64%	55.28%	23.79%	-65,040
		SD	0.69	3.46%	9.34%	8.90%	14,433
	Buyer 2	Mean		97.55%	56.39%	25.37%	-64,166
		SD		4.13%	9.84%	8.67%	16,560
	Buyer 3	Mean		97.40%	54.31%	23.75%	-63,050
		SD		5.08%	10.13%	8.65%	15,863
	Buyer 4	Mean		97.41%	56.30%	24.21%	-66,371
		SD		4.61%	9.26%	9.03%	16,560

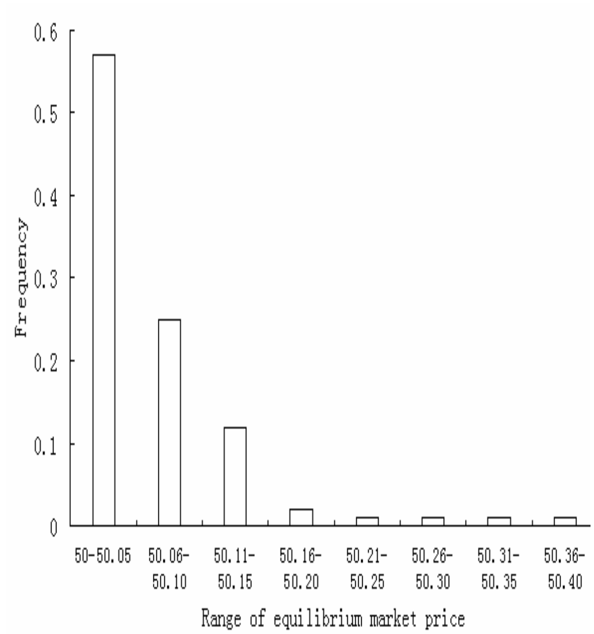
Note:

1. Marginal capacity cost is \$30, which is c^C = \$30.
2. Price ratio is the bidding price of a buyer relative to $R - c^C$ = \$70 here; for example, 50% price ratio means this buyer will bid with \$35.
3. Capacity ratio is the processing quantity plan of a buyer relative to $(R - c^C) \times N =$ 7,000; for example, buyer 1's capacity ratio is 83.30%, and actual capacity ratio is 49.98% means it plans to purchase 8,330 products but actually get 4,998 in the market.

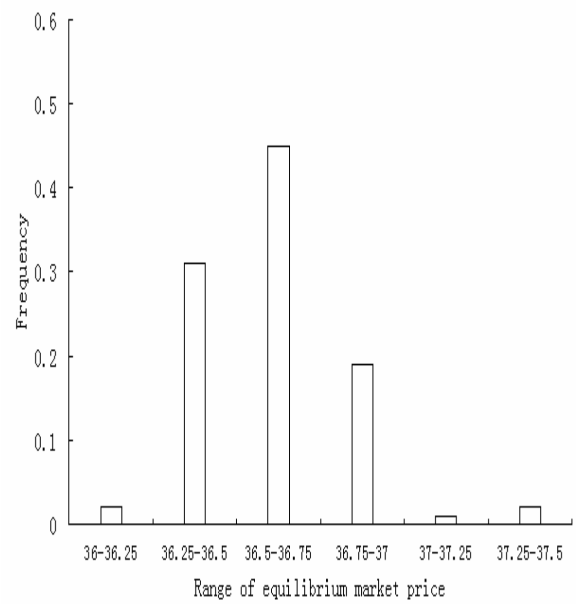
Table II-4. Equilibria under Different Market Settings

		No Capacity Cost			$30 Capacity Cost		
Market Structure	Numbers of Active Buyers	Monopsony Price Level	Perfect Competition Price Level	Others	Monopsony Price Level	Perfect Competition Price Level	Others
Duopsony	1				8.5%		
	2	100%			88.5%		3%
Triopsony	1	5%				9%	7%
	2	95%				29%	13%
	3					28%	14%
4-Buyer	1						
	2					1%	
	3					4%	
	4		100%			75%	20%

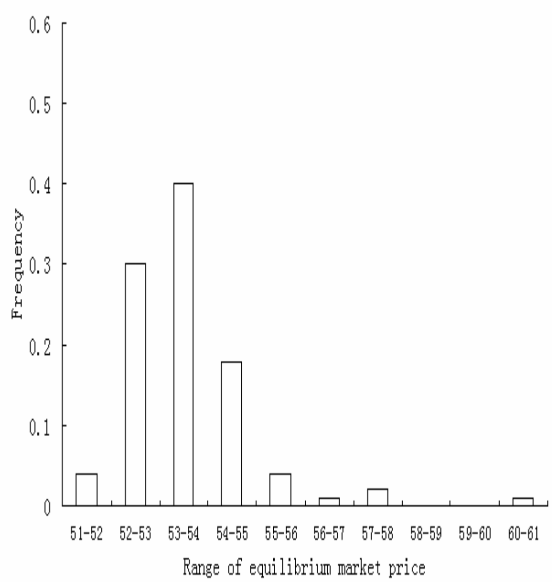
Note:
1. The percentage in this table presents the frequency of an equilibrium among 100 simulation rounds under each setting.
2. Under zero capacity cost setting, market price between $98.5 and $100 are looked as perfect competition level.
3. Under $30 capacity cost setting, market price between $68.5 and $70 are looked at as perfect competition level.



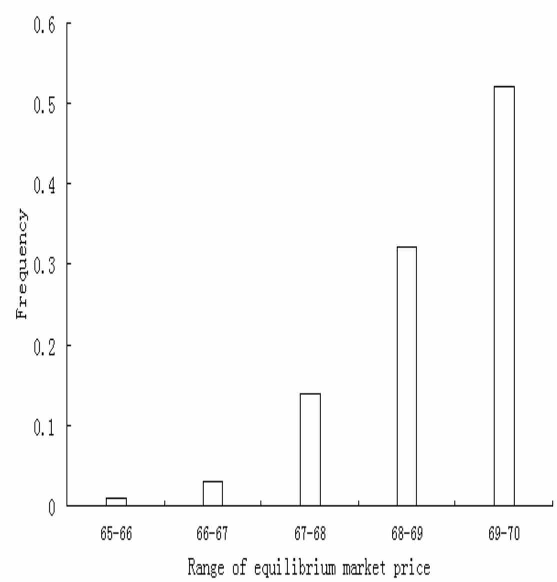
a. Duopsony market without capacity cost



b. Duopsony market with $30 capacity cost

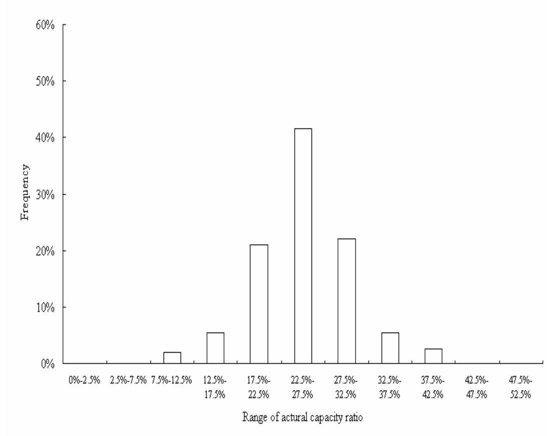


c. Triopsony market without capacity cost

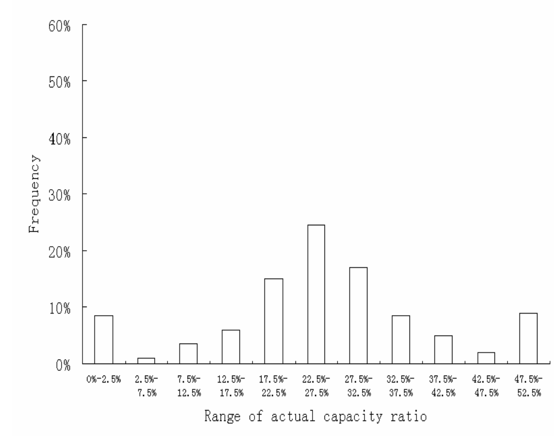


d. Triopsony market with $30 capacity cost

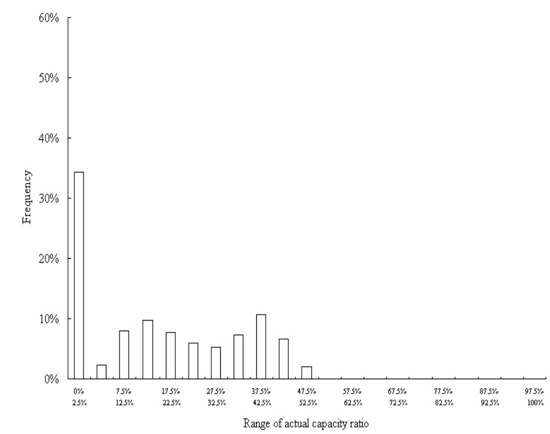
Figure II-1. Frequency of equilibrium market price



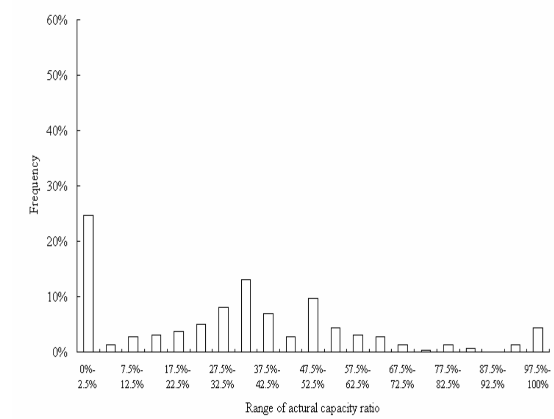
a. Duopsony market without capacity cost



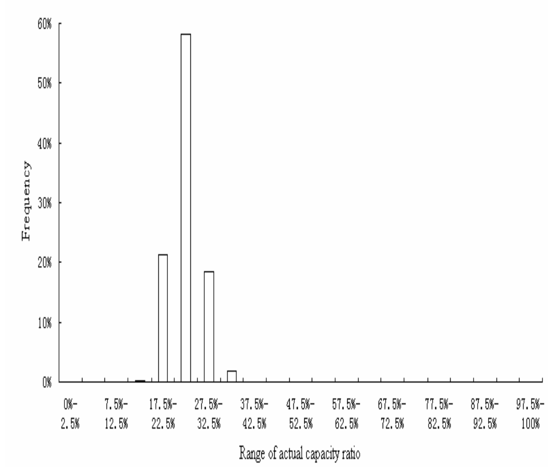
b. Duopsony market with capacity cost



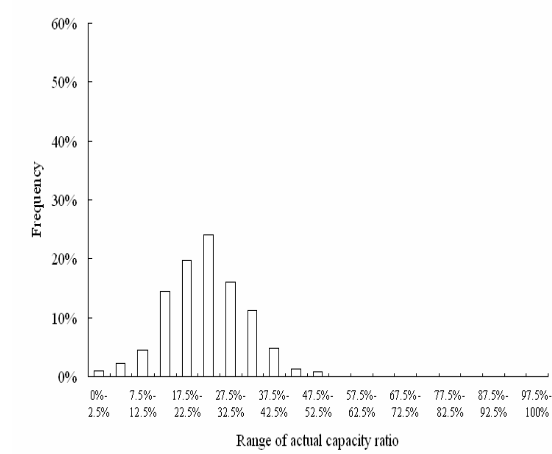
c. Triopsony market without capacity cost



d. Triopsony market with capacity cost

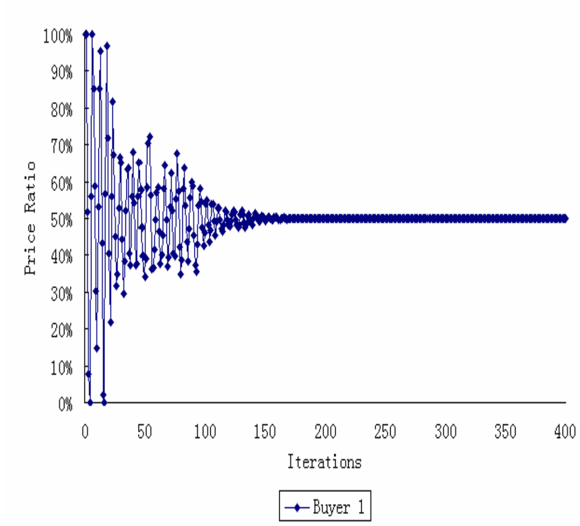


e. Four-buyer market without capacity

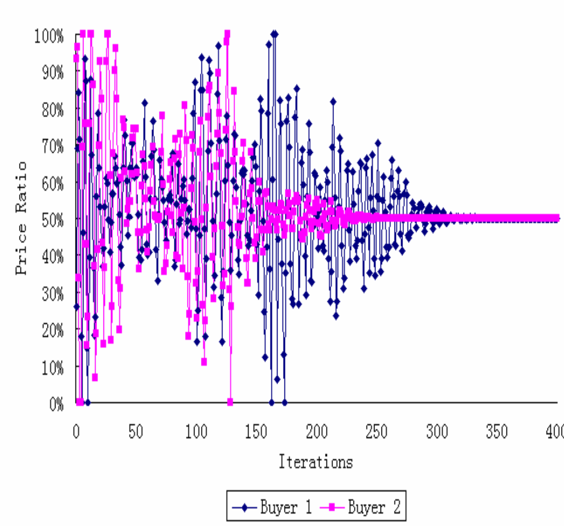


f. Four-buyer market with capacity cost

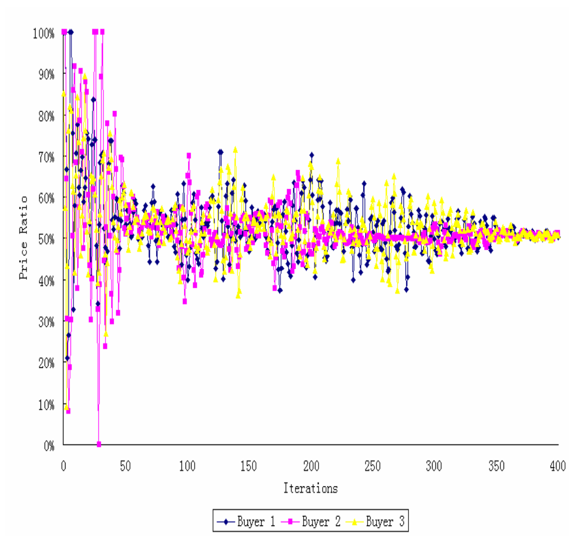
Figure II-2. Frequency of buyers' actual capacity ratio



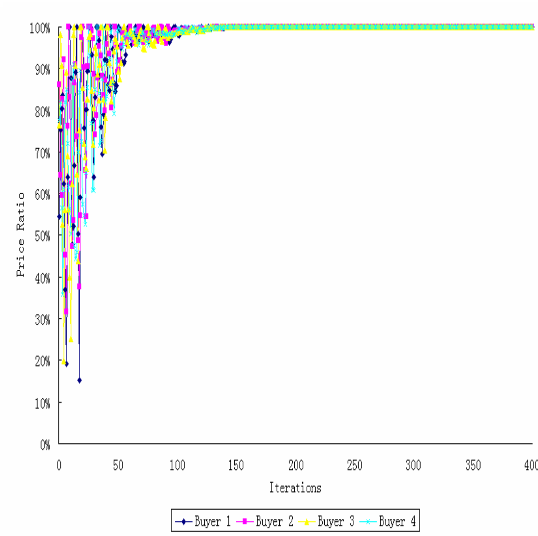
(a) Monopsony market



(b) Duopsony market



(c) Triopsony market



(d) 4-buyer market

Figure II-3.Buyers' pricing behavior without capacity cost (c^C = 0)

CHAPTER III

THE LONG RUN AND SHORT RUN IMPACT OF CAPTIVE SUPPLIES ON SPOT MARKET PRICE: AN AGENT-BASED ARTIFICIAL MARKET

Introduction

In the beef packing industry, vertical integration through captive supplies between packers and feeders has increased significantly during the past decades. Captive supplies include marketing agreements, packer owned cattle, and forward contracts. Most packers procure cattle both through exclusive captive supply contracts and from the spot market. According to a recent GIPSA Livestock and Meat Marketing Study (USDA 2007), 38.3% of cattle were purchased with captive supplies, from which marketing agreements take the largest share of 28.8%, with 4.5% forward contracted, and the rest packer owned. The price of captive supply cattle is typically linked to the subsequent spot market price in one way or another. In addition to the increased vertical integration, the U.S. beef processing industry also experienced horizontal integration with the four-firm concentration ratio reaching 80% in 2002 (Ward).

The increased use of captive supplies by oligopsony packing firms has led to concern about negative impacts of captive supplies on the cash market. Xia and Sexton (2004) construct a theoretical duopsony market where packers purchase cattle both with exclusive captive contracts and in the cash market, and the price of captive supplies is linked to the spot market price. They show that packers can use captive supplies to reduce competition and depress the cash market price to the monopsony level if 50% of the cattle are contracted. In contrast to the large price depression predicted by Xia and Sexton's theoretical model, previous empirical studies have found that captive supplies have only a small negative or insignificant effect. Ward, Koontz and Schroeder (1998) find small negative relationships between price and the percentage of cattle delivered with forward contracts and marketing agreements. Parcell, Schroeder and Dhuyvetter (1997) find that a 1% increase in captive supply shipments is associated with a $0.02/cwt and $0.03/cwt reduction in basis in Colorado and Texas. USDA (2007) gives similar results as the previous empirical studies and shows that a 10% increase in capacity utilization through captive supplies is associated with a $0.04 per pound of carcass weight decrease in the cash market.

One possible explanation of the difference between Xia and Sexton's static model and the previous empirical results is that price depression from captive supplies is a short run effect. In the long run, if packers reduce the price they pay for cattle, contracted feeders will reduce the number of cattle they produce. While the Xia and Sexton result is mathematically correct, its assumptions may not match what happens in actual cattle markets. In this article, we demonstrate that the extra market power provided by captive supplies is a short run phenomenon.

In this research, we use the agent-based computational economics (ACE) method to study the fed cattle market by conducting experiments with programmed agents. Agent-based computational economics (ACE) simulates games between interactive agents (Tesfatsion 2001) and adopts concepts and methods from game theory, cognitive science and computer science. ACE models are increasingly used to study economic phenomena and are especially suitable to simulate economic games in which agents interact with each other assuming bounded rationality and simple adaptive learning rules.

ACE has been used to study the behavior of agents in the cobweb model, the exchange rate problem, prisoner's dilemma, etc. (Ariforic 1996; Axelrod 1987; Erev and Roth 1998; Riechmann 2001; Vriend 2000), but to our knowledge, has not been used in agricultural economics. ACE can be used to study problems with simple behavioral assumptions that are too difficult to analyze with mathematical methods. ACE is more economical and time efficient compared to experiments with human subjects and it is more controllable.

This research uses a particle swarm optimization algorithm to model the learning behavior of agents in the artificial fed cattle market. PSO is a stochastic optimization technique developed by Eberhart and Kennedy (1995). The idea of PSO came from observing how flocks of birds, fish, or other animals adapt to avoid predators or to find food by sharing information. In our game, packers do not cooperate with each other and only learn from their own experience. Thus, we adjust PSO by constructing multiple parallel markets and letting each agent have its own clones in every market. Agents trade in every market simultaneously and independently, but they learn from their experience with their own clones. This means each packer has a separate "flock of birds" that does not share information with the flocks of the other agents.

In this research, we develop an artificial fed cattle market using an agent-based model and use it to determine the impacts of captive supplies under different short run and long run contract assumptions. We verify that price depression impact of captive supplies exists in the short run, but find that it disappears in the long run.

The Oligopsony Market with Captive Supplies

Consider a homogeneous product market with M packers and N feeders. The number of packers is much less than the number of feeders ($M \ll N$). Packers procure from feeders and sell processed goods to the retail market. To focus our research on the game between packers and feeders in this market, we assume that the final processed boxed beef price, the processing rate, and the marginal cost are constant, so the fed cattle value to packers is also constant. This result means the marginal revenue for each animal is constant, and we define the marginal revenues as R.

Packers first make contracts with chosen feeders and then compete for the remaining cattle in the spot market. We follow Xia and Sexton's assumption that packers use quantity as their competition strategy. The market prices are determined by the packers' total demand in the spot market and the aggregate supply from the non-contracted feeders. In this section, we construct three scenarios by first fixing both the number of contracts and the quantity per contract. Next, we allow supply response by feeders. Finally, we allow supply response and let packers choose the number of captive supply contracts. The agent-based models under the three assumptions are developed in the next section. The simulation result with the agent-based model is compared with the theoretical result.

Fixed Number of Contracts and Fixed Quantity per Contract

We first present a theoretical model that extends the Xia and Sexton (2004) results. Xia and Sexton only consider the duopsony case, but we generalize their results to the oligopsony case of M packers.

Assume M processing packers and N feeders in the fed cattle market. Packers purchase cattle from feeders with both exclusive contracts and in the spot market. The price of contracted cattle is linked to the spot market price. Packers choose quantities rather than price and so this is a Cournot game.

Assume packers make exclusive contracts with n_i^c chosen feeders, and the quantity of each contract q_i^c is fixed, where c indicates contract market. In each period, the contracted feeders deliver cattle to packers and packers compete with each other for cattle from the non-contracted feeders. The spot price is determined by the market clearing price from the spot market aggregate demand and supply, and the contracted cattle are also valued with this price. Feeders always accept the contracts. We use S to indicate the total number of feeders with contracts, $S = \sum_{i=1}^{M} n_i^c$ and $S < N$.

At the beginning of each processing period, packers make procurement strategies simultaneously and then purchase cattle in the spot market. The choice variable of the procurement strategy is the procurement ratio:

(3.1) $$x_{i,t}^d = q_{i,t}^d /(R \times N),$$

where $x_{i,t}^d$ is the procurement ratio, N is the total number of feeders, and the superscript d indicates packer's demand in the spot market. Packer i 's processing quantity q_i^d is also the amount of its procurement. R is the marginal revenue of one packer and also the supply level of feeders under the perfect competition price level. For example, if under perfect competition, all feeders will provide 10,000 cattle and the processing quantity of packer i is 3,000, its procurement ratio x_i equals 0.3.

The total demand in the spot market can be written as $Q_t^d = \sum_{i=1}^{M} q_{i,t}^d$. We assume all feeders are homogeneous and have a linear supply function $q_{j,t}^s = p_t$, so the total supply in the spot market is $Q_t^s = (N-S)p_t$, since the contracted S feeders deliver all their products with contracts. The market clearing condition is where the spot market aggregate demand equals supply, which is $Q_t^s = Q_t^d$. Thus we obtain the equilibrium spot market price:

(3.2) $$p_t = Q_t^d \big/ (N-S).$$

Packer i's total profit, which is determined by the quantity it purchases both with captive contracts and in the spot market, is $\pi_{i,t} = (R - p_t)(q_{i,t}^d + n_i q^c)$, $i = 1,...,M$. Because the quantity per contract is fixed, the contract quantity $n_i q^c$ is constant for each processing period. Thus in every period, packers only need to decide how many cattle to buy through the spot market to maximize their profit. In addition, since we know that packers' procurement decisions will also affect the spot market price, we substitute equation (3.2) into the packers' profit function and solve its first order conditions with respect to $q_{i,t}^d$, holding $n_i q^c$ fixed to get the following packers' reaction functions:

(3.3) $$q_{i,t}^d = R(N-S)/2 - \sum_{i' \neq i} q_{i',t}^d / 2 - n_i^c q^c / 2, \text{ for all } i = 1,...,M.$$

Simultaneously solving these reaction functions of M packers, we obtain the spot demand quantities for each packer:

(3.4) $q_{i,t}^{d} = R(N-S)/(M+1) + (S-n_i^c)q^c/(M+1) - n_i^c q^c M/(M+1)$, for $i = 1,...,M$.

Add the above individual spot demands together and substitute the aggregate spot market demand $Q_t^d = \sum_{i=1}^{M} q_{i,t}^d$ into equation (3.2), and the spot market clearing price is

(3.5) $$p_t = MR/(M+1) - Sq^c/[(M+1)(N-S)].$$

From this result, we can see that without captive supplies, which means $S = 0$, the equilibrium price is the Cournot oligopsony level. With captive supplies, the price is lower than without them.

Now we assume the contracted feeder does not have a supply response and quantity q^c is agreed to be fixed to a value. We assume that the fixed quantity per contract will be based on the long run equilibrium price. Thus, packers and contracted feeders fix the quantity of a captive contract to Ep . Substitute $q^c = Ep$ to equation (3.5), which results in:

(3.6) $$Ep = M(N-S)R/[(M+1)N - MS].$$

If the oligopsony model is restricted to be a duopsony model by setting $M = 2$, this spot market price becomes $Ep = 2(N-S)R/(3N-2S)$, which is same as equation (5') in Xia and Sexton (2004). In addition, when the total $(M-1)N/M$ number of feeders sign captive contracts and agree to produce at the market price level, the spot market price reaches the monopsony level $R/2$. For example, when there are $M = 4$ packers in the market, they need to make exclusive contracts with $3N/4$ feeders to depress the spot market to the monopsony level. In Xia and Sexton's duopsony model, packers only need to contract with $S = N/2$ feeders to depress the spot market price to the monopsony level. These results illustrate that the larger the number of packers, the larger number of aggregate exclusive contracts that are needed to depress the spot market price the same amount.

From the above results, we can see that the spot market price could be depressed to the monopsony level, when both the number of contracts and the quantity per contract are fixed.

Fixed Number of Contracts and Flexible Quantity per Contract

Now relax the previous model by allowing a supply response from contracted feeders. Other assumptions are the same as with the previous model. *M* packers and *N* feeders in the market, and the total contracted feeder number is *S*. The spot market price is the same as equation (3.2).

We assume that the contracts are made one period ahead and that contracted feeders will produce the quantity based on the expected spot market price of the delivery period. Thus, the supply equation of the contracted feeder is adjusted as $q_{j,t}^{s} = Ep_t$. Substitute this contract quantity into packers' total profit function, there are $\pi_{i,t} = (R - p_t)(q_{i,t}^{d} + n_i q_t^{c}) = (R - p_t)(q_{i,t}^{d} + n_i Ep_t)$, $i = 1,...,M$. When the market reaches equilibrium, the spot market prices between different time periods will be the same, which means $Ep_t = p_t$. Substitute this condition and equation (3.2) into the profit function, and take the first order condition with respect to the packers' procurement quantity $q_{i,t}^{d}$, and the result is the packers' reaction functions:

(3.7) $$q_{i,t}^{d} = (N - S)R/2 - (N - S + 2n_i^{c})\sum_{i' \neq i} q_{i',t}^{d} \Big/ [2(N - S + n_i^{c})], \text{ for all } i = 1,...,M .$$

Simultaneously solving these reaction functions of *M* packers for the aggregate demand

$Q^d = \sum q_{i,t}^d$ in the spot market and then substitute the result in the market clearing equation (3.2), we get the spot market clearing price in equilibrium as

(3.8) $$Ep = R[(N-S)M+S]/[(M+1)(N-S)+2S].$$

From this result, we can see that without captive supplies, which means $S = 0$, the equilibrium price is still the Cournot level. If we restrict the oligopsony model to be a duopsony model by setting the number of packers $M = 2$, this spot market price becomes $Ep = R(2N-S)/(3N-S)$, which is higher than that in the previous model. For example, when $S = N/2$, the spot market price is $3R/5$, which is higher than the monopsony level $R/2$ but lower than the duopsony level $2R/3$.

From the results above, we can see that with a fixed number of contracts and with supply response, the spot market price level is higher than without supply response. But, in this model captive supplies still reduce market prices.

Flexible Contracts and Flexible Quantity per Contract

Now assume that in the long run feeders who sign captive supply contracts have a supply response and packers can adjust their captive supply contract numbers and the procurement quantity in the spot market.

First, packers decide how many feeders to make exclusive contracts with. The contract ratio x^c is used as a packer's captive supply choice variable:

(3.9) $$x_{i,t}^c = n_{i,t}^c / N,$$

where $x_{i,t}^c$ is the contract ratio of packer i, which indicates the percent of feeders out of the total number of feeders with whom this packer contracts in time t. Then feeders decide how many cattle they will produce based on their expectation of the market price. We can reasonably assume that feeders expect the spot market price of the next period will be the same as the current one. Thus, with a linear supply function that has an intercept of zero and a slope of one, feeders will deliver $q_t^c = p_{t-1}$ to their contracted packers. At last, packers decide how many cattle to process with strategy x^q and compete with other packers in the spot market from the remaining feeders. Thus the packers' profit function changes to:

(3.10) $$\pi_{i,t} = (R - p_t)(q_{i,t}^d + n_{i,t}^c q_t^c) = (R - p_t)(q_{i,t}^d + n_{i,t}^c p_{t-1}), \text{ for all } i = 1,...,M .$$

The maximization of the above functions involves variables in multiple time periods and the current period contains two choice variables for each packer. Solving such a dynamic model with mathematical analysis would be difficult. Therefore, we use an agent-based model to simulate this market. In the following section, we introduce the market design of the agent-based model for an artificial oligopsony market with captive contracts.

Agent Based Artificial Fed Cattle Market with PSO Algorithm

An agent-based model is a computer simulation artificial market which contains multiple programmed strategic agents interacting in an economic market system. These agents have simple behavioral rules and can learn to use better strategies based on their past experiences.

In this article, we use programmed intelligent agents acting as *N* feeders and *M* packers in the fed cattle market. Feeders are price takers, and packers compete for cattle both with captive supply contracts and in the spot market. The transactions between packers and feeders occur in a captive contract market and in a cash market. We set up three simulation procedures: a) fixed number of contracts and fixed quantity per contract; b) fixed number of contracts and flexible quantity per contract, and c) flexible number of contract and flexible quantity per contract. Figure III-1 illustrates how packers and feeders dynamically make their transactions under these market designs.

In the simulation, we assume that packers choose quantities and that market participants discover the interception point of the current aggregate demand and supply curve and use it as the market clearing price. Thus, if no captive supply is present, the simulation results should be exactly what the Cournot theory predicts. Since packers cannot form enforceable agreements, if any market power is exercised which makes the spot market price lower than the Cournot result, it must be done through captive supply. The following paragraphs describe the market mechanisms of each market and the behaviors of agents in them. Market participants have their own marketing strategies and can improve their performance by learning based on their past performance.

Figures III-1(a) and (b) show the time lines with short run and long run periods. In the short run, we assume that both the captive contract and the quantity per contract are fixed. Under this assumption, we simulate the behavior of packers to show how they adjust their spot market procurement quantity to find the best response level. This process means that during the short run simulation, packers only have one choice variable, the procurement ratio in the spot market. Different from the short run model, Figure III-1(b) shows that in the long run, packers can select the number of contract feeders as well as the procurement ratio in the spot market. Also, contracted feeders have a supply response with respect to the spot market price.

The learning behaviors of packers are modeled with a particle swarm optimization algorithm. By playing the game repeatedly, packers can learn from their own experiences and adopt the best strategy for themselves.

Particle Swarm Optimization Algorithm

To model the adaptive learning of agents, evolutionary (genetic algorithm) and machine learning (reinforcement learning) algorithms are increasingly applied to ACE (Ariforic 1994; Axelrod 1987; Erev and Roth 1998). With GA, researchers have to be very careful to choose parameters and methods for each problem or the result may be premature convergence. The large population size also makes finding equilibrium with GA time consuming. Reinforcement learning is a sub-area of machine learning, and the environment is typically formulated as a finite state Markov decision process in which an agent increases the probability of choosing successful strategies under the possible strategy spaces of its rivals. When the possible strategy space is large or continuous, the computational cost increases exponentially. To avoid the problems of GA and RL, we use a particle swarm optimization algorithm to model the learning behavior of agents. As we state in the introduction section, agents have their own parallel clones and share information between them. This kind of marketing strategy can be observed in many real markets. For example in fed cattle market, packing firms send many agents to purchase cattle from feeders and each of them visits feeders in a certain area. Agents bid differently but they will share information with their colleges at the end of each period and adjust their strategies to increase profit. This sharing of information does not occur in GA, and this may explain why PSO has been found to lead to faster convergence.

In the market simulated here, agents face a changing economic environment since all agents continuously update their market strategies. We set up *K* parallel markets, and agents each have their own clones in every market. For example, with 20 parallel markets, agents each have 20 clones as the population from which they can learn. Although having the same behavioral rules, the *K* clones of one agent may take a different strategy in each market since the initialized random values are different. In the simulation, packers dynamically change their marketing strategies with the PSO algorithm but feeders are price takers and simply sell their products at the market price.

Suppose the k^{th} clone of packer i chooses $x_{i,k}^{\Gamma}$ as one of its two strategy parameters, $x_{i,k}^{\Gamma} \in U[0,1]$, and each strategy parameter is randomly initialized at the beginning of the simulation, here Γ indicates a strategy in the decision set of a clone. Each clone has a velocity, $v_{i,k}^{\Gamma} \in U[0,1]$, which determines the change of the strategy value. The changes of choice variables are influenced by the value of the best solutions achieved by the k^{th} clone itself, $p_{i,k}^{l,\Gamma} \in U[0,1]$, and by the best solution among the whole population, $p_i^{g,\Gamma} \in U[0,1]$. The superscripts *l* and *g* indicate local and global, the subscripts *k* and i indicate k^{th} parallel market and i^{th} packer respectively. Profit function $\pi_k(\mathbf{x}_{i,k})$ is used to evaluate the performance of each decision set $\mathbf{x}_{i,k} = [x_{i,k}^{q}, x_{i,k}^{c}]'$.

In every simulation step, each strategy of the k^{th} clone of packer i is selected using equations:

(3.11) $$x_{i,k,t+1}^{\Gamma} = x_{i,k,t}^{\Gamma} + v_{i,k,t}^{\Gamma} \text{ and}$$

(3.12) $$v_{i,k,t+1}^{\Gamma} = w \cdot v_{i,k,t}^{\Gamma} + c_1 u_1 (p_{i,k,t}^{\Gamma,l} - x_{i,k,t}^{\Gamma}) + c_2 u_2 (p_{i,k,t}^{\Gamma,g} - x_{i,k,t}^{\Gamma}),$$

where $x^{\Gamma}_{i,k,t}$ indicates the strategy, $v^{\Gamma}_{i,k,t}$ is the velocity vector, $u_{\zeta} \in U[0,1]$, and $\zeta = 1,2$ are uniformly distributed random numbers, c_1 and c_2 are learning parameters and can be called self confidence factor and swarm confidence factor respectively, w is an inertia weight factor, $p^{\Gamma,l}_{i,k}$ and $p^{\Gamma,g}_{i}$ are local best and global best, and the subscript of Γ is d or c to indicate strategy x as procurement ratio or contract ratio. The calculated value of $x^{\Gamma}_{i,k,t+1}$ or $v^{\Gamma}_{i,k,t+1}$ is restricted to be one or zero when it overflows the range.

The following equations indicate how to choose $p^{\Gamma,l}_{i,k}$ and $p^{\Gamma,g}_{i}$ among all parameters of agent i. Under a dynamic environment where agents' best response strategy depends on how others behave, the fitness value of the previous local best may not be the same when it is used in the current economic environment. Then the local best needs to be retested. The best locals of the previous L iterations are retested under the current market environment. The current best local is choosen from the best performance past parameters $p^{\Gamma,l}_{i,k,t'}$ and the current strategy:

(3.13) $$p^{\Gamma,l}_{i,k,t} = \arg\max\left\{\pi_k(\mathbf{p}^{l}_{i,k,t-1}),\ldots,\pi_k(\mathbf{p}^{l}_{i,k,t-L}),\pi_k(\mathbf{x}^{\Gamma}_{i,k,t})\middle|\mathbf{x}^{\Gamma}_{i'\neq i,k,t}\right\},$$

where $k = 1,2,...,K$ and i' indicates packer i's rivals. And the best global parameter is selected from the best local parameters:

(3.14) $$p^{\Gamma,g}_{i,t} = \arg\max\left\{\pi_1(\mathbf{p}^{\Gamma,l}_{i,1,t}),\ldots,\pi_K(\mathbf{p}^{\Gamma,l}_{i,K,t})\right\},$$

where K is the total number of parallel markets.

The inertia weight w in (3.12) is critical in affecting the speed of convergence (Chatterjee and Siarry 2006). A large inertia weight provides a larger exploration but a slow convergence, while a smaller inertia weight is needed to fine-tune the current search area. It is worth making a compromise, such as starting with a higher value at the beginning and then decreasing w with iterations:

$$w_t = \beta_0^w + \beta_1^w \left(t_{\max} - t\right) / t_{\max} , \tag{3.15}$$

where $t_{\max}$ is the maximum number of iterations and t is the current iteration. Self confidence and global confidence factors c_1 and c_2 in equation (3.12) can be set as constant and are usually between 0.5 and 2.5. Here we choose 1 for both of them.

Simulation Procedure with PSO

There are M packers and N feeders. Each packer and feeder has K clones in the K parallel markets. Each clone of a packer may have a different trading strategy in each parallel market. The steps in the simulation are:

(i) In each market, randomly initialize $x_{i,k,t}^{\Gamma}$ and $v_{i,k,t}^{\Gamma}$ for all i. We choose the quantity ratio $x_{i,k,t}^{\Gamma} \in U[0,1]$ and $v_{i,k,t}^{\Gamma} = 0$ for all $i = 1,..., M$, $k = 1,..., K$, and $t = 1,..., L$.

(ii) Select the best locals for each clone with equation (3.13).

(iii) Select best global for each packer with equation (3.14).

(iv) While the market is not converged, each packer continuously uses the function (3.11) and (3.12) to select new strategies.

Equilibrium Criterion

Typically, zero diversity in the population's strategies among all markets signals the stopping point for a PSO. As the population evolves, diversity diminishes and each agent uses the same strategy in each parallel market. For every agent, if the variance of the strategies in the population is less than 0.01% and the variance of the mean value of the strategies for 10 generations is less than 0.01%, we say the algorithm reaches equilibrium.

Simulation Results

We design two scenarios under the short run assumption by simulating with and without a supply response from the feeders with fixed captive supply contracts, and one scenario under the long run assumption by simulating with flexible captive supply contracts. In each scenario, we determine the market equilibrium of a duopsony market and an oligopsony market containing 4 packers. Thus, in this research we have 12 simulation settings.

The parameters used in the three scenarios are shown in Table III-1. The market parameters and PSO parameters are the same for all scenarios and the packer number *M* is 2 in the duopsony markets and 4 in the oligopsony markets. We have 400 feeders in each market. The cattle value before processing is \$100 for packers. For the PSO algorithm, we choose 1.5 and 0.5 for the intercept β_0^w and slope β_1^w of inertia weight w in equation (3.15), and let the maximum iteration $t_{\max}$ be 500. Both c_1 and c_2 in equation (3.12) are chosen as constant value 1 with 20 parallel markets. For the two short run scenarios, since we assume that the captive supply contracts last for infinite periods with or without contract quantity fixed, we also set the values of these two variables as parameters, and the chosen values are shown in Table III-1.

A simulation round contains multiple iterations that agents repeatedly play the game until the market reaches equilibrium. We run 100 rounds for each of the 12 experimental settings with different random starting values. Within each simulation round, we let agents trade until the equilibrium is determined by the convergence criterion or a maximum of 500 iterations is reached. At the beginning, packers randomly choose their strategies and learn to use better strategies based on their experience.

In the short run simulation, the captive contracts are fixed, and packers interact in the market to find the optimal procurement strategies with or without quantity per contract fixed. We simulate the market by letting packers contract with 50% and 75% of feeders in the duopsony market and the four packer market respectively. Since packers are homogeneous, we can reasonably assume that packers will split the contracts equally, and each of them will contract with 25% of the total feeders in the duopsony market and 18.75% in the 4 packer market. In the short run setting without contract supply response, we set the quantity per contract as 50. According to our theoretical derivation, if packers contract with $(M\text{-}1)N/M$ feeders and the contract quantities are fixed at the monopsony level $R/2$, packers can depress the spot market price to the monopsony level. So we use this setting to test if packers in the artificial market can learn to find the optimal procurement strategies to benefit from the monopsony price in the spot market. Thus, in the short run simulation, packers have one choice variable - the procurement ratio; but in the long run, they have two choice variables - the contract ratio and the procurement ratio.

The mean and standard deviation of the market price and packers' strategies at equilibrium from 100 rounds are calculated. The simulation results are shown in Table III-3. The results show that in the short run, packers can depress the spot market price to the monopsony level of \$50 for both the duopsony market and the oligopsony market; but in the long run, packers compete to obtain the Cournot results, and the spot market price is \$66.7 in the duopsony market and around \$80 in the 4-packer market. In addition, when packers can choose both the number of captive supply contracts and the procurement quantity in the spot market, packers most often use the spot market to purchase cattle. This result means that in the long run, packers cannot use captive contracts to depress the spot market price, and packers behave like they do not need captive supplies as an alternative procurement method.

Besides the statistical analysis of the market equilibrium, Figures III-2 to III-5 show the dynamics of the spot market price and the packers' strategies in an individual run under each experimental setting. Figure III-2 shows the market prices for the duopsony and 4-packer models under the long run assumption and the short-run assumption with a fixed contract without contract supply response. From Figure III-2, we can see that if packers make long term contracts with feeders and the quantity of contracts are fixed to a carefully chosen value, they can depress the spot market price to the monopsony level of \$50 even without collusion. However, without long term contracts where packers adjust strategies on both captive supply and spot market procurement, the spot market price goes to the Cournot solution.

Figure III-3(a) shows the simulation results of the duopsony market under fixed contracts without contract supply response. The figure shows that at equilibrium, each packer uses a procurement ratio of 12.5% as its optimal strategy, which yields 5,000 spot market procurement quantities according to equation (3.1) since R and N equal \$100 and 400. Thus, the total demand in the spot market is 10,000. Substituting this quantity and the number of uncontracted feeders of 200 into equation (3.2), we see that the market price is $50. This result is consistent with our simulation results in Figure III-1 and the theoretical results of Xia and Sexton (2004).

Following the method above, we simulate the four-packer market by letting packers make contracts with 75% of the total feeders. The contract quantity is also fixed at 50. The simulation results in Figure III-3(b) show that the market reaches equilibrium when each packer uses a spot procurement ratio around 3.125% as its strategy. Substitute these values into equation (3.5), and we can also get the market price as $50. These results are consistent with our simulated results in Figure III-2. The results confirm that when the market contains more packers, the packers need to contract with more feeders than the duopsony market to depress the spot price to the monopsony price level.

Both Figures III-4 and III-5 show the simulation results for markets without fixed contracts. Under this situation, packers adjusted the number of feeders they contract with as well as the spot procurement strategies. Also, contracted feeders are allowed to have a supply response. In long run markets, packers show oligopsony behavior. The spot market plays a crucial role, and packers tend to purchase all demand in it instead of the captive supply market, and the contract ratio goes to zero. Packers use 33.33% and 20% spot market procurement ratios in the two markets respectively which is consistent with the results of the traditional Cournot model without captive supply markets.

There is no price depressing effect of captive supplies, but that is because packers choose to have no captive supplies.

Conclusions

An agent-based model is used to study the impact of captive supplies under fixed or flexible contracts. With a fixed number of contracts with or without supply response, analytical solutions are available. But for the long run scenario with flexible contracts and flexible quantity per contract, the solution cannot be found with mathematical analysis and an agent-based simulation method is used. The agent-based model has been used in economics but is new to agricultural economists. This model provides an alternative method to study the complex problems which are difficult to solve with mathematical analysis and less costly than experiments with human subjects.

Our simulation results indicate that captive supplies can depress the spot market price in the short run if the contracts are fixed. This result is consistent with Xia and Sexton's model. But this is a short run effect. In the long run when the packers can adjust the number of contracts and feeders have a supply response for contract quantity, the price depression phenomena of captive supplies disappears since packers do not contract any cattle.

References

Arifovic, J. 1994. "Genetic Algorithm Learning and the Cobweb-Model." *Journal of Economic Dynamics and Control* 18:3-28.

Arifovic, J. 1996. "The Behavior of the Exchange Rate in the Genetic Algorithm and Experimental Economies." *Journal of Political Economy* 104:510-41.

Axelrod, R. 1987. "The Evolution of Strategies in the Iterated Prisoner's Dilemma." In *Genetic Algorithms and Simulated Annealing* pp. 32-41. London: Pitman.

Chatterjee, A. and P. Siarry. 2006. "Nonlinear Inertia Weight Variation for Dynamic Adaptation in Particle Swarm Optimization" *Computers & Operations Research* 33: 859-87.

Eberhart, R.C., and J. Kennedy. 1995. "A New Optimizer Using Particle Swarm Theory." *Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan*. pp. 39-43.

Erev, I., and A. Roth. 1998. "Predicting How People Play Games: Reinforcement Learning in Experimental Games with Unique Mixed Strategy Equilibria." *American Economic Review* 88:848–881.

Parcell, J.L., T.C. Schroeder, and K.C. Dhuyvetter. 1997. "The Effect of Captive Supply on Live Cattle Basis." *Journal of Agricultural and Resource Economics* 22:394-.

Riechmann, T. 2001. "Genetic Algorithm Learning and Evolutionary Games." *Journal of Economic Dynamics & Control* 25**:** 1019-1037.

Tesfatsion, L. 2001. "Introduction to the Special Issue on Agent-Based Computational Economics." *Journal of Economic Dynamics and Control* 25:281-293.

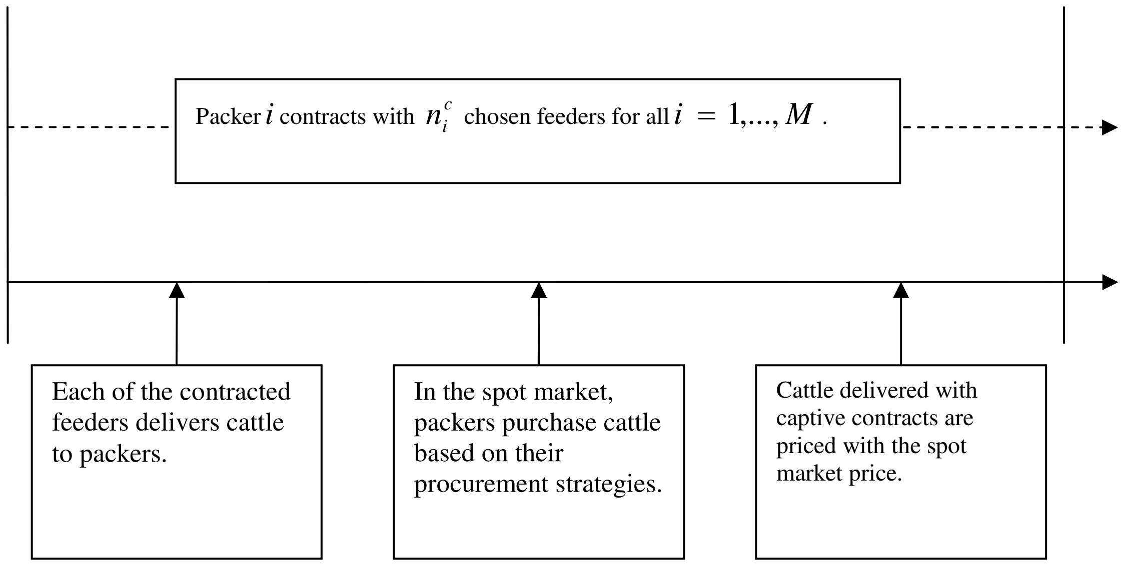
U.S. Department of Agriculture. 2007. "Fed Cattle and Beef Industries Final Report." *GIPSA Livestock and Meat Marketing Study*. Grain Inspection, Packers and Stockyards Administration. Washington, DC, RTI-0209230. http://www.usda.gov/gipsa/pubs/captive_supply/captive.htm.

Ward, C.E. 2002. "A Review of Causes for and Consequences of Economic Concentration in the U.S. Meatpacking Industry." *Current Agriculture, Food & Resource Issues* 3:1-28.

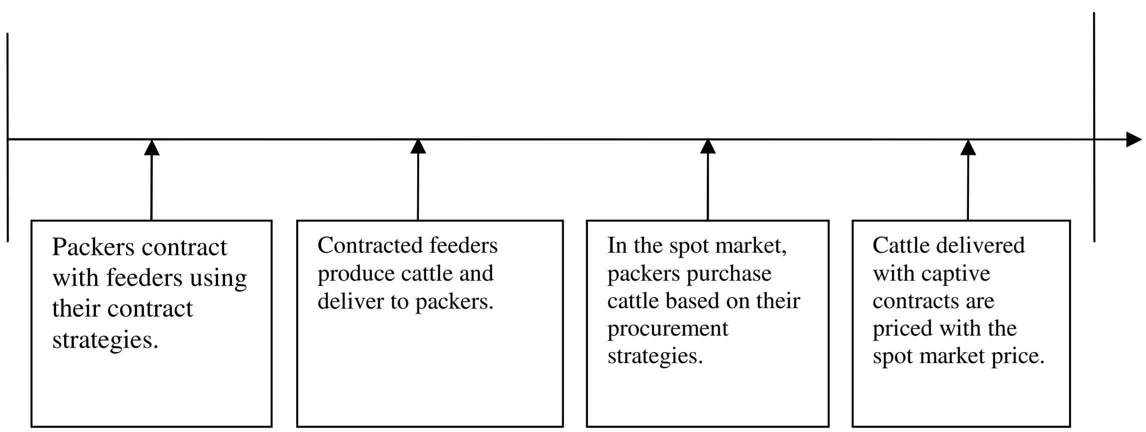
Ward, C.E., S.R. Koontz, and T.C. Schroeder. 1998. “Impacts from Captive Supplies on Fed Cattle Transaction Prices.” *Journal of Agricultural and Resource Economics* 23:494-514.

Xia, T., and R.J. Sexton. 2004. “The Competitive Implications of Top-of-The-Market and Related Contract-Pricing Clauses.” *American Journal of Agricultural Economics* 86:124-138.

Vriend, J.N. 2000. “An Illustration of the Essential Difference between Individual and Social Learning, and Its Consequences for Computational Analyses.” *Journal of Economic Dynamics & Control* 24:1-19.



(a). Fixed number of contracts and with or without captive supply response



(b). Long run model with flexible number of contracts and captive supply response

Figure III-1. The timeline of the model

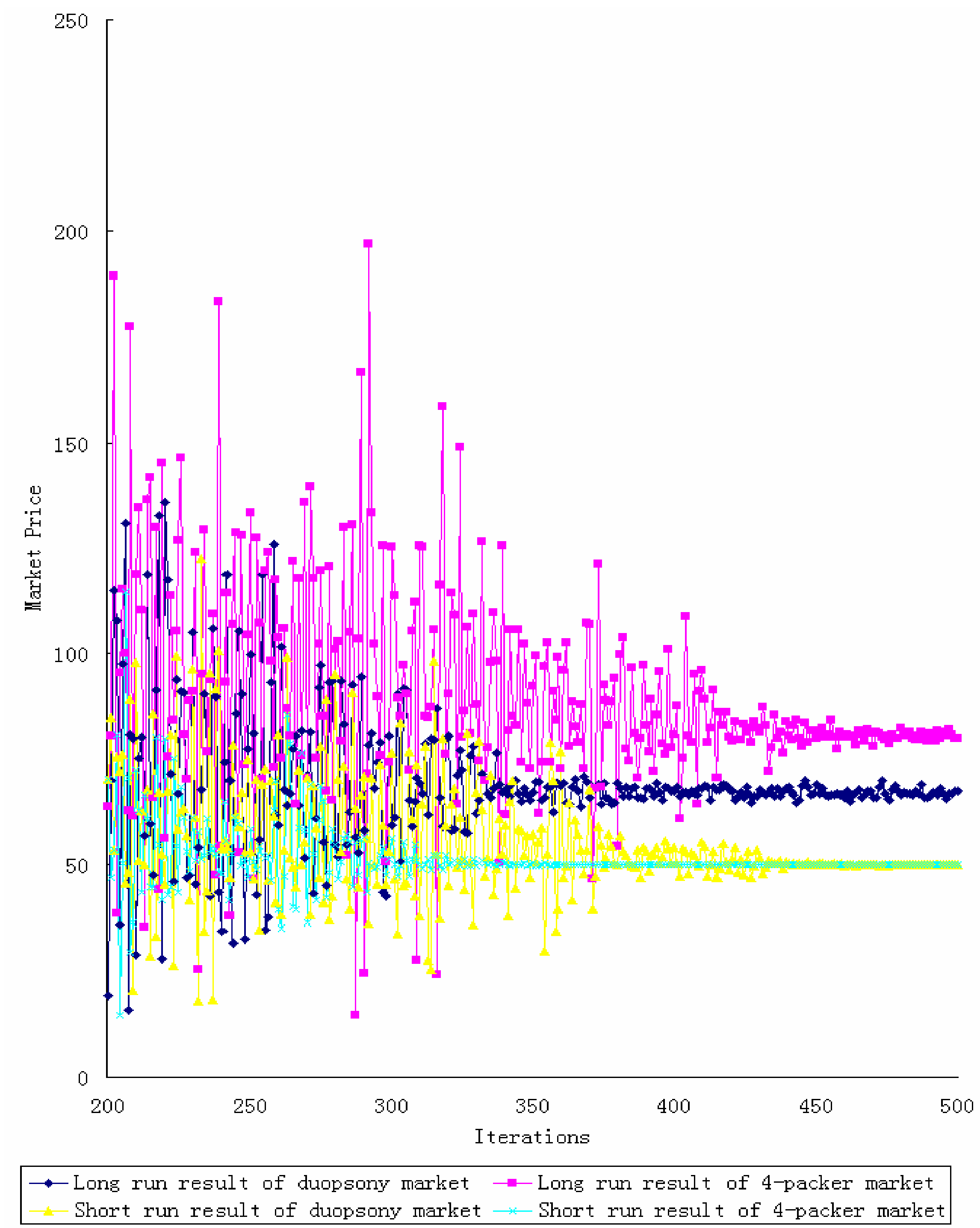
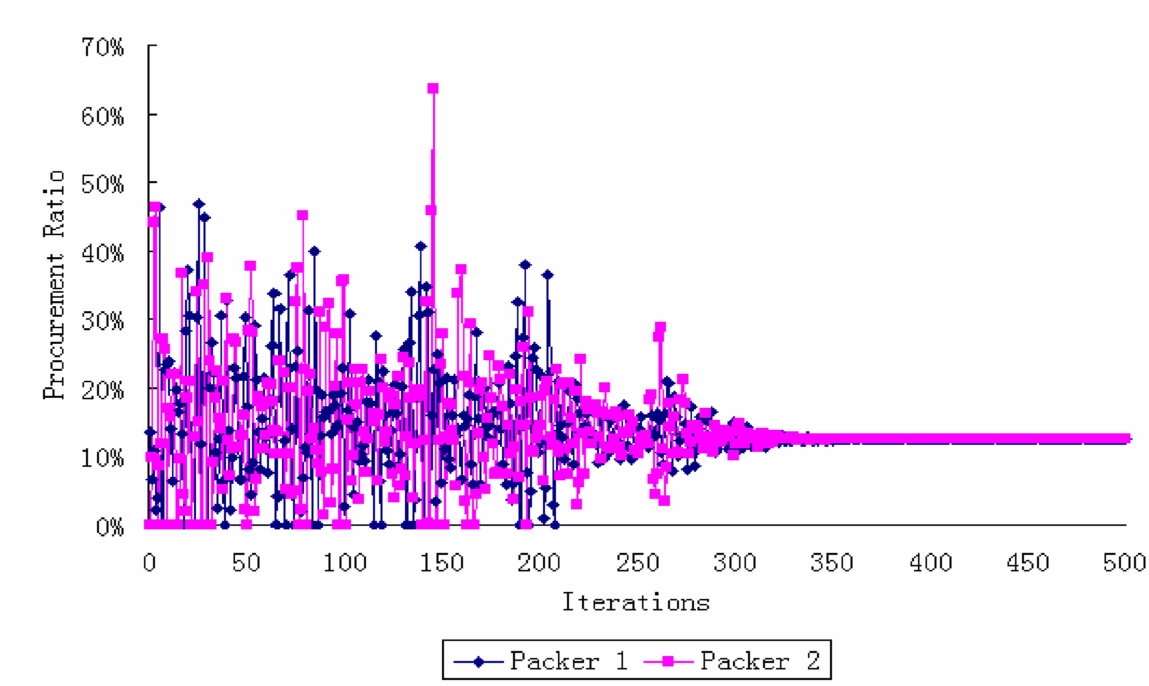
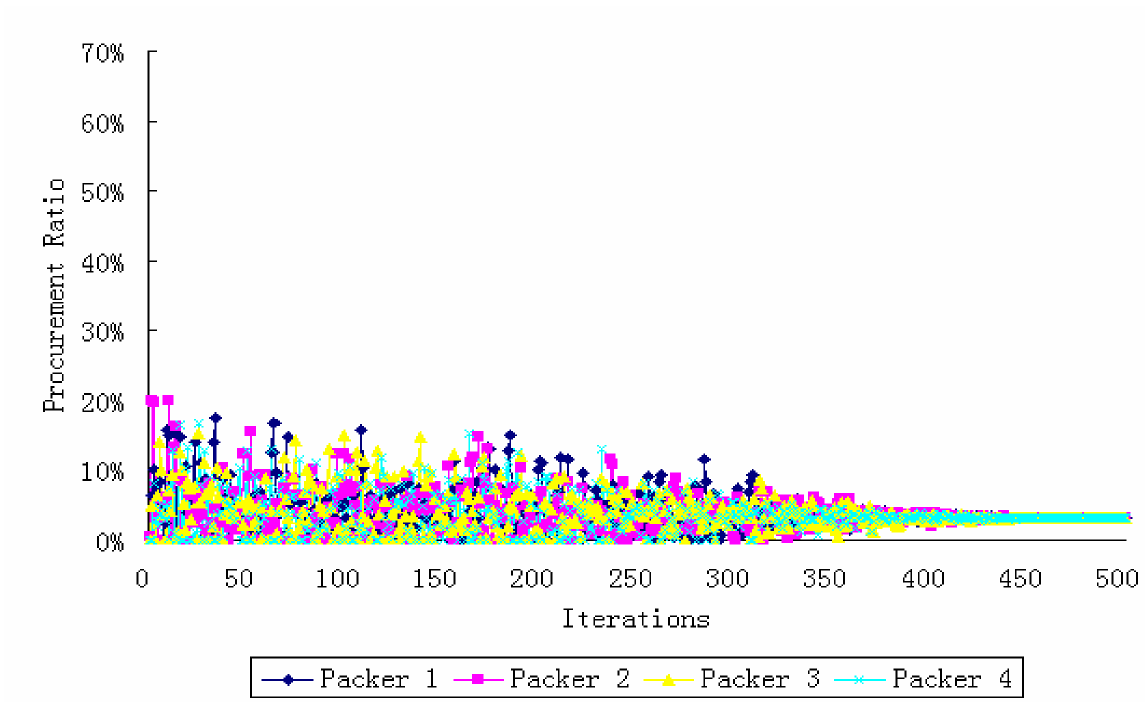


Figure III-2. Spot prices of duopsony and four-packer markets

Note: For the two short run settings, both the number of contracts and quantity per contract are fixed.

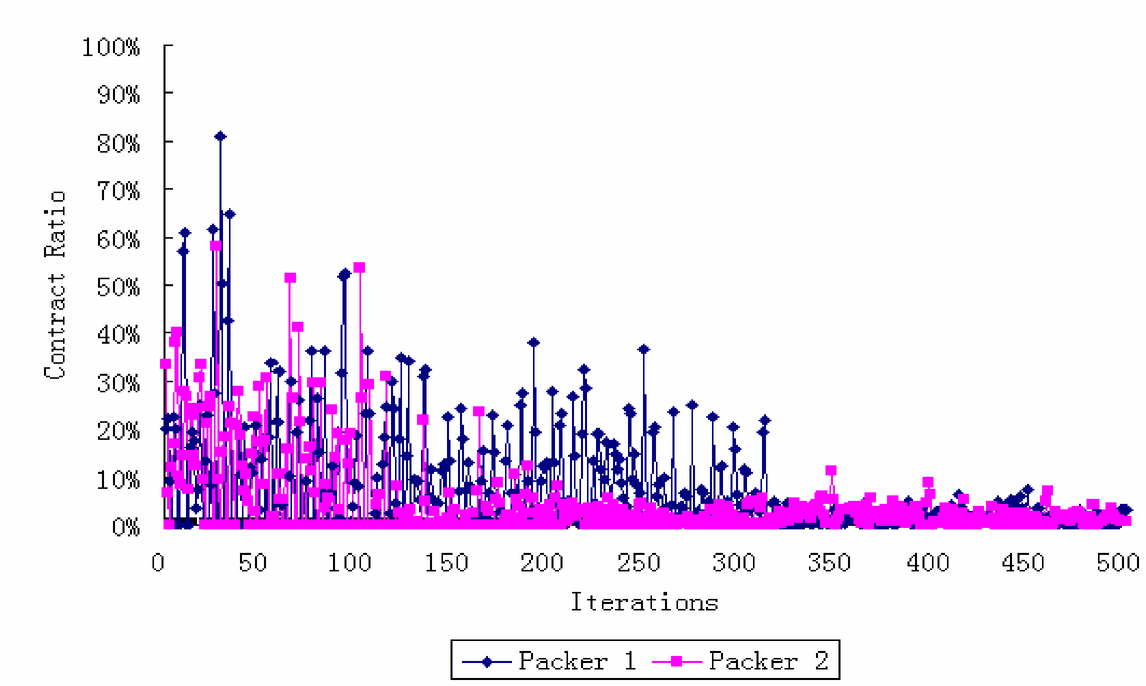


(a). Duopsony Market

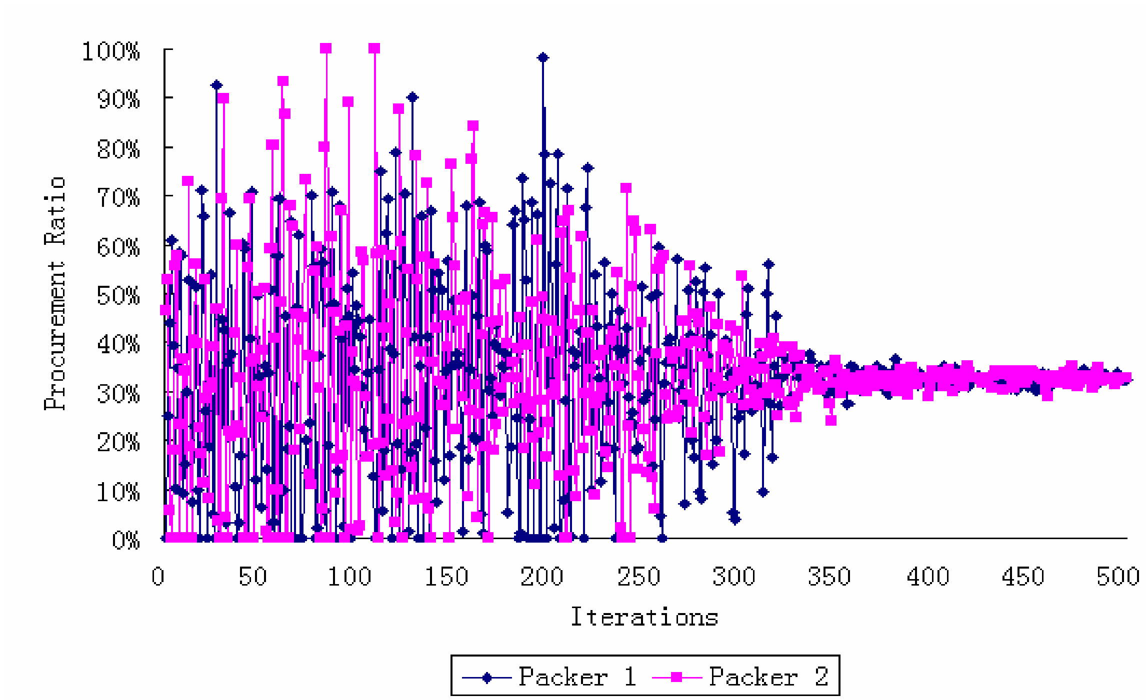


(b). Four-Packer Market

Figure III-3. Packers' short run procurement ratio in the spot market without contracts supply response

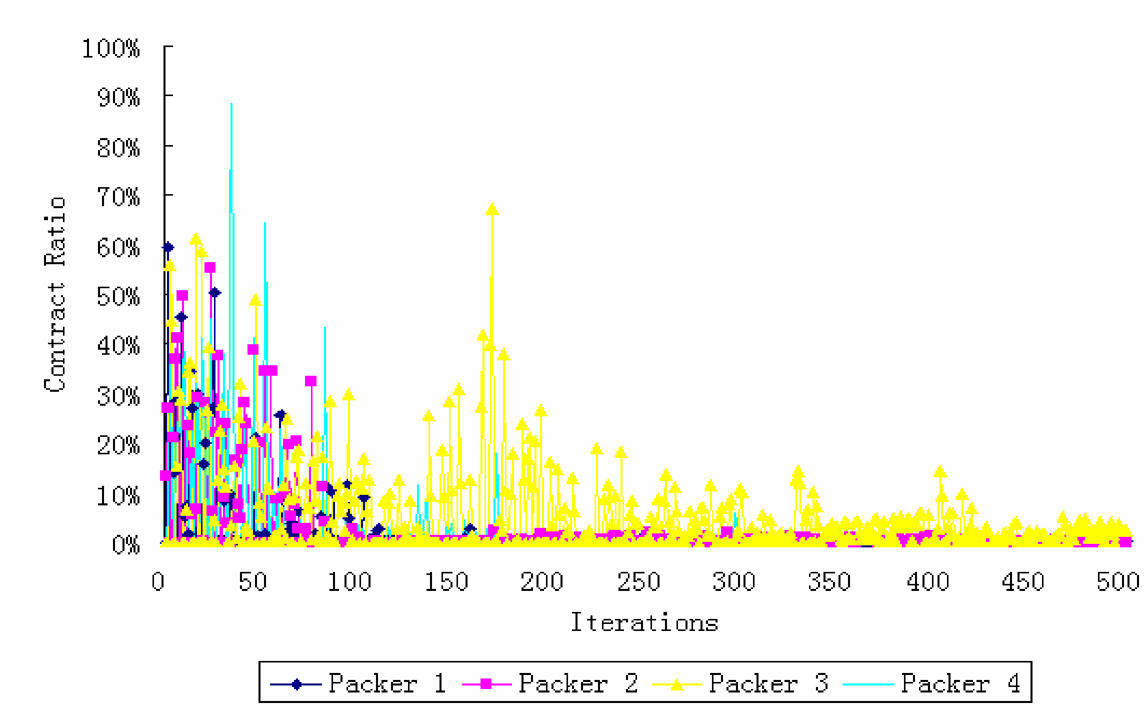


(a). Contract Ratio

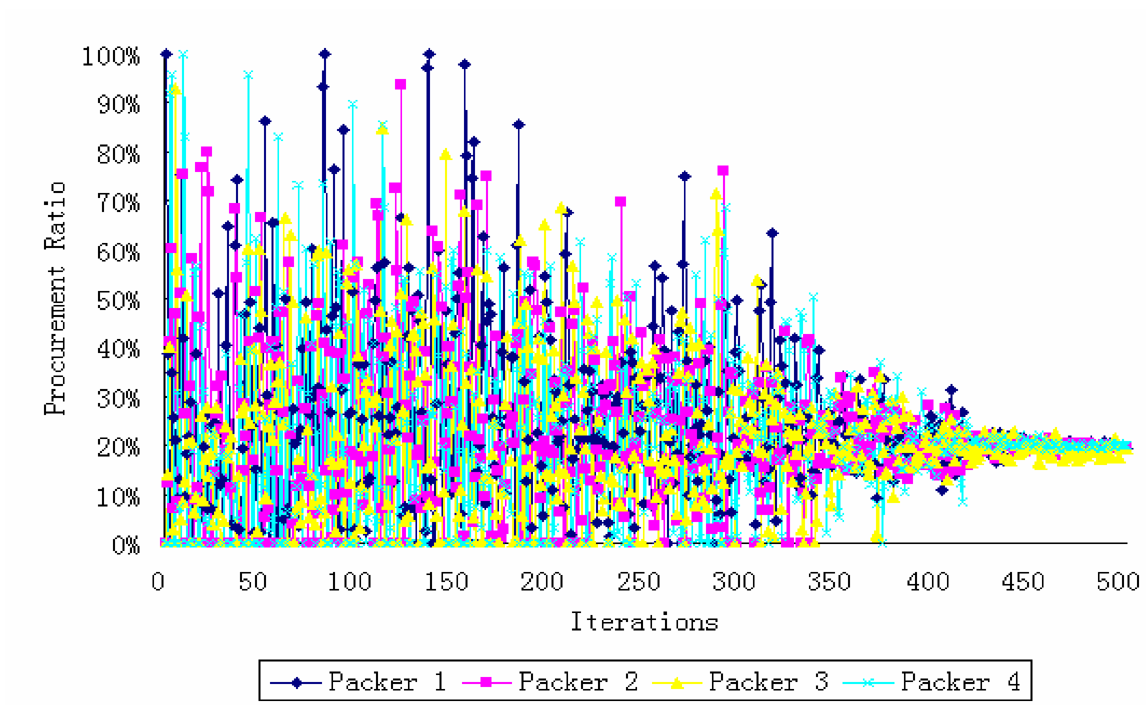


(b). Spot Procurement Ratio

Figure III-4. Packers' strategies in duopsony market under long run assumption



(a). Contract Ratio



(b). Spot Procurement Ratio

Figure III-5. Packers' strategies in four-packer market under long run assumption

Table III-1. Parameter Setting in Artificial Market Simulation Design

Parameter	Symbol	Value
Market Parameters		
Number of Packers	M	2 for duopsony market; 4 for four-packer market
Number of Feeders	N	400
Cattle Value Before Processing	R	$100
Particle Swarm Optimization (PSO) Algorithm Parameters		
Intercept of inertia weight in equation (3.15)	β_0^w	1.5
Slope of inertia weight in equation (3.15)	β_1^w	0.5
Self and global confidence factors of PSO	$c_1 = c_2$	1
Number of parallel markets	K	20
Maximum iteration of one simulation round	$t_{\max}$	500
Parameters for Model with Fixed Contracts		
Number of contracted feeders for each packer	n^c	$N/4$ for duopsony market; $3N/16$ for four-packer market
Quantity per captive supply contract	q^c	50

Table III-2. Short Run and Long Run Simulation Results of Market Prices and Packers' Strategies under Duospsony Market and Four-Packer Oligopsony Market Settings

Market Structure	Packer	Statistic	Short Run						Long Run			
			Without Contract Supply Response			With Contract Supply Response						
			Market Price	Procurement Ratio	Profit	Market Price	Procurement Ratio	Profit	Market Price	Contract Ratio	Procurement Ratio	Profit
Duopsony		Mean	50.00			60.00			66.76			
		SD	0.00			0.00			0.54			
	Packer 1	Mean		12.50%	500,000		15.00%	480,000		1.55%	32.45%	444,000
		SD		0.00%	0		0.00%	0		1.52%	0.90%	7,539
	Packer 2	Mean		12.50%	500,000		15.00%	480,000		1.89%	32.14%	443,000
		SD		0.00%	0		0.00%	0		1.15%	0.76%	4,702
Four-Packer		Mean	50.00			66.41			80.20			
		SD	0.01			1.20			0.41			
	Packer 1	Mean		3.13%	250,000		4.12%	218,000		0.83%	19.48%	159,500
		SD		0.00%	0		0.34%	9,515		0.79%	0.42%	2,236
	Packer 2	Mean		3.13%	250,000		4.15%	219,000		1.22%	19.07%	158,500
		SD		0.00%	0		0.36%	7,182		0.93%	0.83%	4,894
	Packer 3	Mean		3.13%	250,000		4.20%	220,500		1.16%	19.32%	160,000
		SD		0.00%	0		0.25%	8,256		0.97%	0.58%	3,244
	Packer 4	Mean		3.13%	250,000		4.14%	219,500		1.16%	19.16%	159,000
		SD		0.00%	0		0.25%	8,870		1.05%	0.99%	3,078

Note:

1. In the short run duopsony market, each packer uses a fixed captive contract ratio of 25% , which means it contracts with 100 feeders in every iteration period;
2. In the short run four-packer market, each packer uses a fixed captive contract ratio of 18.75% , which means it contracts with 75 feeders in every iteration period;
3. Contract quantities are fixed at 50 for short run markets without contract supply response.

VITA

Tong Zhang

Candidate for the Degree of

Doctor of Philosophy

Dissertation: AGENT-BASED ARTIFICIAL MARKETS

Major Field: Agricultural Economics

Education:

Received Bachelor of Science degree in physics from Jilin University, Changchun, P.R. China in July of 1995; obtained Master of Science degree in Mechanical Engineering from Beijing Institute of Technology University, Beijing, P.R. China in April of 2001; obtained Master of Science degree in Agricultural Economics from Oklahoma State University in Stillwater, Oklahoma in August of 2005; completed the requirements for the Doctor of Philosophy degree with a major in Agricultural Economics at Oklahoma State University in July 2008.

Experience:

Research assistant for the Department of Agricultural Economics at Oklahoma State University from 2004 to present.

Fellowships and Awards:

Spillman Scholarship at Oklahoma State University, 2006.

Name: Tong Zhang | Date of Degree: July, 2008

Institution: Oklahoma State University | Location: Stillwater, Oklahoma

Title of Study: AGENT-BASED ARTIFICIAL MARKETS

Pages in Study: 79 | Candidate for the Degree of Doctor of Philosophy

Major Field: Agricultural Economics

Scope and Method of Study: This study is composed of three essays. The first essay adapts the particle swarm optimization algorithm (PSO) to find the equilibrium in an agent-based artificial market. The simulated agents follow simple behavioral rules. The PSO algorithm is compared to a genetic algorithm (GA) under a Cournot game. The second essay uses the agent-based model with PSO algorithm to find equilibrium in a price-quantity competition game. In this game, agents purchase products from sellers and sell processed goods to the retail market using both bid price and capacity as their strategies. Simulations consider different numbers of buyers and are performed with and without a capacity cost. The third essay uses an agent-based model to estimate the impact of captive supplies on the spot market price under long run and short run assumptions in fed cattle markets. Packers purchase cattle from feeders both under captive supply contracts and in the spot market. Captive contracts are assumed fixed in the short run and flexible in the long run. Packers have one choice variable, procurement quantity in the spot market, in the short run; and have an additional choice variable, number of captive contracts in the long run.

Findings and Conclusions: The first essay successfully adapts the PSO algorithm to solve dynamic economic games. PSO gives faster convergence and more precise answers than the genetic GA methods used by some previous economic studies. The agent-based model is new to agricultural economics and suitable to study complex economic problems that are hard to solve with mathematical methods. The simulation results of the second essay show that the agent-based model can explain the collusion and competition phenomena observed in previous experimental studies with human subjects which cannot be explained by theories. Under price-quantity competition, prices with one or two firms are at the monopsony level and with four firms prices are always at the perfectly competitive level; but the triopsony market changes from mostly monopsony to perfect competition when capacity cost increases from zero to a higher level. The third essay shows that the price depressing effect of captive supplies found in previous theoretical work is a short run effect and in the long run this phenomenon disappears.

ADVISER'S APPROVAL: Dr. B. Wade Brorsen