Genetic Operators and Market Selection: An Agent-Based Modeling Approach ⋆

Bin-Tzong Chie [a,*], Shu-Heng Chen [b], Ching-Ting Yao [c]

[a] *Department of Industrial Economics, Tamkang University, Taipei County 251, Taiwan.*

[b] *AI-ECON Research Center, Department of Economics, National Chengchi University, Taipei 116, Taiwan,*

[c] *International Tax Services, Ernst&Young, Taipei 110, Taiwan*

Abstract

In this study, the simple genetic programming has been modified by the automatically defined terminals. An agent-based two-firm competition market is build to test firms' survivabilities. Through the simulation, we discover the significance of genetic operator rates in product designs by observing its impact on consumer satisfaction and firms' competitiveness.

JEL classification: D20; D43; D83

Key words: Innovation; Agent-based modeling; Automatically defined terminals; Genetic operator

1 Motivation and Introduction

Extended from our previous studies Chen and Chie (2004, 2006, 2007), which have emphasized *modularity*, we replicate an environment to simulate the evolutionary and innovation process of commodities with *genetic programming* (GP). Meanwhile, the role of *genetic operators* (e.g., recombination and mutation) shall not be overlooked under the GP application. This can be evidenced

⋆ Research Support from NSC grant No. NSC97-2410-H-032-062 is gratefully acknowledged.

* Corresponding author at Department of Industrial Economics, Tamkang University, Taipei County 251, Taiwan. Tel: +886 2 2621-5656 ext 2975

Email addresses: chie@aiecon.org (Bin-Tzong Chie), chchen@nccu.edu.tw (Shu-Heng Chen), Janice.Yao@tw.ey.com (Ching-Ting Yao).

from two extreme examples. One is that the market selects commodities without recombination and mutation; therefore, little improvement can be made. Another is when recombination and mutation occur too often in production so that consumers face different commodities each time choice being made. Commodities with positive elements are replaced with completely new creation. In the latter, knowledge of invention is hard to accumulate and preserve. To remedy this failure, a model with more subtlety is established to simulate the market competition in a two-firm environment. Through large scale simulation, the role of genetic operator in production competencies will also be exploited.

In Chen and Chie (2007), simple GP is modified with modularity approach, which is automatically defined terminals (ADTs). Borrowing the idea of Koza Koza (1994), ADTs have been found to have an affinity for knowledge accumulation. The characteristics of primitive and advanced commodity designs can be captured by ADTs. For instance, it is possible to depict the processing of wheat to flour, and from flour to dough by introducing ADTs. The model also shared the some similarities with Koza's automatically defined functions (ADFs), including hierarchical mechanism, simplicity, and encapsulation.

According to Goldberg (2002), if we combine crossover and selection, continuous innovation will result. In like manner, the combination of mutation and selection is the essence of improvement. However, little discussion has been focused on the impacts brought by different crossover and mutation rates, especially in the area of agent-based economics modeling of innovation. This paper will try to discuss these two genetic operators based on simulation results. Two observations of economic experiments have been carried out. The first one is the contribution of two competitive firms to consumer satisfaction under two different selection pressures. The selection pressure stems from consumer's search intensity, for instance, in a two-firm competition market, consumer should visit both of the firms under high search ability and only visit one of the two firms under low search ability; The second one is the importance of genetic operators in competition among firms. We follow the agent-based model of Chen and Chie (2007) to allow a firm who designs new products with higher crossover rate or mutation rate to compete with the other firm who has lower crossover rate or mutation rate. In a sense, this is equivalent to replicating two different organizational cultures regarding the competition between open-minded and prudent culture.

The rest of the paper is structured as follows. The agent-based model is introduced in Section 2. Section 3 and Section 4 present two experiments with findings, followed by the concluding remarks made in Section 5.

2 The Model

The model is calibrated under a modular economy. In Chen and Chie (2004), we considered an economy of profit-maximizing firms (producers) and surplus-maximizing consumers. Fig. 1 demonstrates the interaction between these two sorts of agents. Producers supply commodities to the market and earn profit/incur loss as a result. Intuitively, each producer is motivated to engage in product innovation and dominate the market through winning more customer satisfaction. This is defined as the "innovation" process, as shown in the green box. Consumers, on the other side, allocate budget in a way to maximize their respective level of satisfaction. *Consumer Surplus* is measured by the difference of maximum willingness-to-pay and the actual price paid for a product. The maximum willingness-to-pay is assessed by given preference, which will be detailed later.

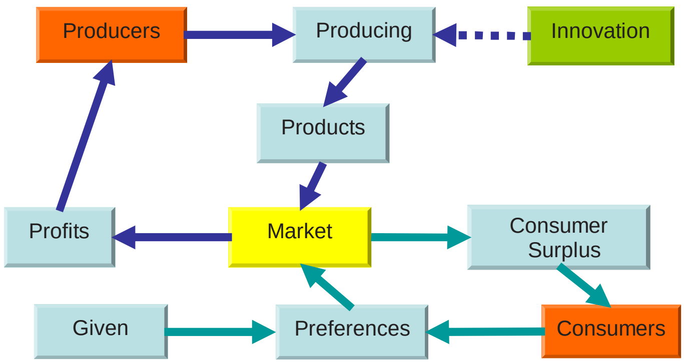

Fig. 1. The agent-based modular economy.

In the following, we will introduce the parameters of *agent engineering* and *environment design*. We first observe the attributes and behaviors of producers and consumers. For example, producers have the capability of *production*, *innovation*, *marketing* and *resource allocation*, which are altogether subject to individual attributes such as *capital limits*, *knowledge*, and *cultural factors*. With a closer investigation, we find that genetic operators play an important role in innovation behavior. Unlike static behavior (e.g., adaptive inventory adjustment), a producer not only perceives market demand through maintaining a variety of product lines, but also evolve and improve the commodities as time goes by. The evolving behaviors are captured by *genetic parameters*. In the meantime, consumers' budget, preference, utility function, and willing-to-pay determine the market demand. The next subsection will discuss the details of parameters.

Producer Parameters

(1) **Working Capital** (k) is a firm's endowment used for production.
(2) **Inventory Adjustment Rate** (λ) adjusts inventory by excess demand (ED) or excess supply of a specific product. The adjustment process can be defined by $q^m_{t+1} = \lambda ED + q^m_t$, where $\lambda \in [0, 1]$ and q^m represents quantity of product m.
(3) **Mark-up Rate** (η) is a component of producer's asking price (ask_m), which is calculated as $ask_m = (1 + \eta)\bar{C}_m$, where $\bar{C}_m$ is the average production cost of product m.
(4) **R&D Rate** ($\gamma_{R\&D}$) is the proportion of working capital spent on R&D. The R&D working capital ($k_{R\&D}$) is determined by the following formula: $k_{R\&D} = \gamma_{R\&D} \times (k - k_{\text{inv}})$, where k_{inv} is the capital invested in inventory adjustment.
(5) **R&D Ceiling** ($\overline{R\&D}$) defines the maximum usage of R&D resource. Due to the limited market size, R&D expenditure should not increase without boundary.
(6) **Cost per Node** (c) is the unit production cost of using a terminal or function node. It is assumed that the costs of terminal and function nodes are the same.

Consumer Parameters

(1) **Consumer Income** (I) is a consumer's endowment exogenously given at the beginning of each period, which usually has a crucial impact on market demand.
(2) **Depth of Preference** (d_p) defines the total tree depth of a consumer's preference. The deeper the preference, the more sophisticated a consumer may be.
(3) **Depth of Common Preference** (d_c) is the depth of identical preference tree structure shared among all consumers.
(4) **Base of Preference to Utility** (z) is a component of the utility function. It is normally greater than 2 to ensure the *synergy effect.* [1]
(5) **Price to Utility Ratio** (v) determines a consumer's subjective valuation of a particular product, which is also known as the *willingness-to-pay* (i.e. bid). It is calculated as $bid = v \times U_m$, where U_m is the consumer's utility level for product m.
(6) **Search Intensity** (r_s) is the percentage of total producers that will be selected. In other words, the selection intensity determines how many producers will be visited by a consumer, which is calculated as $r_s \times n_p$.

[1] The synergy effect of the consumer preference is defined in Section 2.3.

Genetic Parameters

(1) **Initialization of Tree**. According Koza (1992), there are three methods to initialize a tree population. They are *growth method*, *full method*, and *ramped half-and-half method* (a mixed of the growth and full method). In this paper, we use full method to set consumer common preference, and the ramped half-and-half for heterogeneous preference and commodity.
(2) **Number of Primitives** (ρ) is the size of terminal set and function set.
(3) **Initial Tree Depth** (d_{ini}) restricts the depth of the first generation commodity tree. [2]
(4) **Maximum Tree Depth** (d_{max}) describes the maximum commodity tree depth allowed in this simulation. In order to satisfy consumer preference, it is normally higher than d_p and is limited to computer capacity (e.g., memory size).
(5) **Tournament Size Ratio** (r_{ts}) determines the tournament population (POP^j_{ts}) size used to run genetic operators. It is defined as $POP^j_{\text{ts}} = r_{\text{ts}} \times POP^j$, where POP^j represents the whole commodity population of producer j.
(6) **Crossover Rate** (p_c) is the probability of recombining the most two profitable commodity trees in the tournament population.
(7) **Mutation Rate** (p_m) is the chance to alter offspring's terminal or function nodes or sub-trees after crossover.
(8) **Automatically Defined Terminal** (ADT) is a prototype which can be used to produce higher level of commodities. [3]

2.2 Environment Design

This subsection introduces the underlying *learning cycle* and *market rules* for firms and consumers. As mentioned earlier, a firm may maximize profit by producing various products, hereby learning consumer preference. As shown in Fig. 2, the GP driven innovation process constitutes a series of new products launched every generation. In this paper, the model runs 5,000 generations, each calibrating with five trading days. After each learning cycle, a firm determines what to produce (including new products developed via product innovation), how many to produce, as well as the price to charge for the next generation. Such decision is based on the feedbacks (e.g. sales and profit statistics) collected from the previous generation.

The trading between buyers (consumers) and sellers (producers) is also regu-

[2] Due to the lack of market (demand) information, this setting captures the initial stage of producers' behavior, which is to produce less complex and more diverse commodities.
[3] See Chen and Chie (2007) for details.

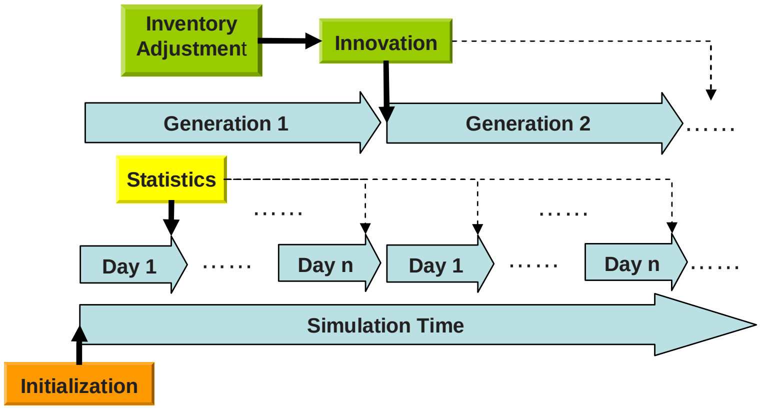

Fig. 2. Market days and learning cycles.

lated by some market rules. First, a buyer will research all available products in the marker before making a purchasing decision. The pricing strategy established by the sellers is a *take-it-or-leave-it* offer based on respective cost and mark-up rate. A buyer's strategy, on the other hand, is to maximize consumer surplus which can be defined as:

$$\max(bid_m^i - ask_m^j),\ \forall j, m \tag{1}$$

and a transaction can be described as

$$buy_m^{ij} = \begin{cases} 1 & \text{if } bid_m^i \geq ask_m^j \text{ and } q_m^j > 0 \\ 0 & \text{otherwise.} \end{cases}$$

where buy_m^{ij} is a binary indicator. When it equals one, a transaction will be carried out, and the price of product m will be $P_m^j = ask_m^j$; otherwise, no transaction will take place. If, however, product m was sold out (i.e., $q_m^j = 0$), the consumer will select the second best choice and the rest may be deduced analogously until $bid_m^i < ask_m^j$. The shortage of product m will then be recorded in the log as *exceed demand* by producer j. In other words, each consumer will at most purchase one product in each trading day. In addition, to avoid negative shopping experience, a consumer's failing to purchase a commodity from a seller today may lower the probability of his or her meeting the same seller tomorrow, which can be described as

$$\text{Prob}_{ijk} = \frac{1 + \sum_{i=1}^{k} buy_m^{ijk}}{k + n_p}. \tag{2}$$

where Prob_{ijk} is the probability that consumer i has met producer j for k times, and n_p is the number of producers. The search intensity (r_s) of a consumer also plays an important role in the trading process. The search space for a consumer is denoted by $\text{round}(r_s \times n_p)$, where round() represents *round-off*

Table 1
Parameter Settings

Parameter	Tyep (Variable)	Range	Default Value
	Producer		
Number of Producers	Integer (n_p)	$[1, \infty)$	1
Initial Working Capital	Integer (K_0)	$[1, \infty)$	500
Working Capital per Gen.	Integer (K)	$[1, \infty)$	500
Inventory Adjustment Rate	Real (λ)	$[0, 1]$	80%
Mark-up Rate	Real (η)	$[0, \infty)$	100%
R&D Rate	Real ($\gamma_{\text{R\&D}}$)	$[0, 1]$	1%
R&D Ceiling	Real ($\overline{\text{R\&D}}$)	$[0, \infty)$	500
Cost per Node	Real (c)	$[0, \infty)$	1.0
	Genetic Operator		
Number of Primitives	Integer (ρ)	$[1, \infty)$	5
Initial Tree Depth	Integer (d_{ini})	$[1, \infty]$	5
Maximum Tree Depth	Integer ($d_{\max}$)	$[1, \infty]$	11
Tournament Size Ratio	Real (r_{ts})	$[0, 1]$	10%
Crossover Rate	Real (p_c)	$[0, 1]$	90%
Mutation Rate	Real (p_m)	$[0, 1]$	80%
Automatically Define Function	Boolean (ADT)	T, F	T
	Consumer		
Number of Consumers	Integer (n_c)	$[1, \infty)$	100
Consumer Income per Gen.	Integer (I)	$[1, \infty)$	10000
Depth of Consumer Preference	Integer (d_p)	$[1, \infty]$	6
Depth of Common Preference	Integer (d_c)	$[1, d_p]$	5
Base of Preference to Utility	Integer (z)	$[2, 10]$	4
Price to Utility Ratio	Real (ν)	$[0, \infty)$	5.0
Search Intensity	Real (r_s)	$[0, 1]$	100%
	Time Schedule		
Trading Days per Gen.	Integer (Day)	$[1, \infty)$	5
Number of Generations (Gen.)	Integer (Gen)	$[1, \infty)$	5000

function.[4] When $r_s < 100\%$, only portion of the producers may be included in a buyer's shopping list. For example, if there are two producers and a hundred consumers, and `consumer 1` has completed eight shopping experiences, which are zero bad and five good transactions with `producer 1` and two bad and one good transactions with `producer 2`, the consumer's updated probabilities for the two producers are $[\text{Prob}_{115}, \text{Prob}_{123}] = [0.86, 0.4]$. After re-scale the probability $[\text{Prob}_{115}, \text{Prob}_{123}] = [0.68, 0.32]$, and then when $r_s = 0.5$, the consumer will have more tendency to choose `producer 1` next time. In other words, in the case of $r_s = 100\%$, the value of Prob_{ijk} will not affect the search space of a consumer, because all the producers will be visited by each consumer. Table 1 summarizes the parameters applied in this model.

[4] The function rounds off $r_s \times n_p$ to the nearest integer.

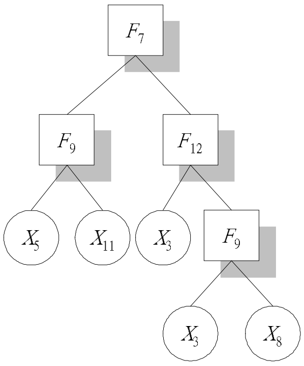

Fig. 3. An illustration of a product.

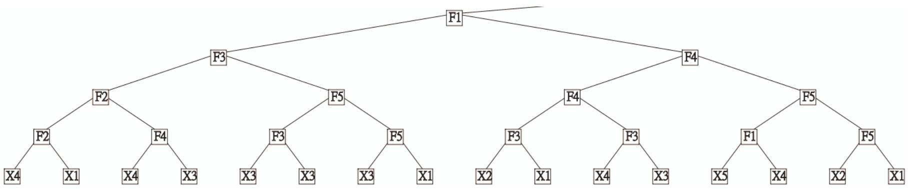

Fig. 4. An illustration of the common preference of a consumer.

2.3 Commodity and Preference

The commodity in the model is represented as a *parse tree* which is shown in Fig. 3. Each parse tree corresponds to a LIST program.[5] The terminal leaves correspond to the raw inputs (materials) $X1$, $X2$, ..., whereas the root and all intermediate nodes represent the processors, $F1$, $F2$, ..., applied to these raw materials in a bottom-up order, as the usual behavior of a LIST program. The whole parse tree can, therefore, be interpreted as a production process associated with the commodity. The unit cost of the commodity is a function of the number of processors and the number of raw inputs, i.e., it is a positive function of the node complexity of the commodity. In a simpler way, we assume that the unit cost is a linear function of the node complexity.[6]

The preference of the consumers is also represented by a GP tree. To make the preference tree well-behaved in economics, three assumptions have been made in Chen and Chie (2004), namely, the *monotone, synergy*, and *consistency condition.* In our simulation, there are 100 consumers in the market. Each consumer has a preference tree with a depth of six. Viewed from the topmost level (the root level), the preference tree is composed of two modules. The one on the left, having a depth of five as shown in the Fig. 4, is identical

[5] In the case of Fig. 3, the product can be represented as a LIST program form, $(F_7(F_9X_5X_{11})(F_{12}X_3(F_9X_3X_8)))$.

[6] The cost function can be defined by $C(N)$, where N is the used number of functional and terminal nodes. The positive function is defined by $\frac{dC}{dN} > 0$, and the linear function is defined by $\frac{dC}{dN} = c$, where c is a constant.

among all consumers, whereas the one on the right, having a depth of five or less, is heterogeneous, and is randomly generated by the ramped half-and-half method, an initialization method frequently used in GP. In this way, consumers' preferences have a common part as well as an idiosyncratic part. The example of an idiosyncratic part is shown in Table 2.

The utility of consuming a commodity is based on measuring the degree of similarity between commodity and preference. Chen and Chie (2006) has developed a module-matching algorithm to perform the task of matching each commodity module (subtree) with each preference module in a descending order relating to the depth of the tree. Under this mechanism, the biggest module will be processed first; if it is successfully matched, the process will stop, otherwise, it will proceed to process the second biggest commodity module until the commodity modules are exhausted. To satisfy the synergy condition and hence the idea of added-value, Chen and Chie (2006) assumes a power utility function for the preference tree as

$$U(S_{d,j}) = \begin{cases} z^{d-1}, & \text{if } j \text{ is matched} \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

$$U = \sum_j U(S_{d,j}). \tag{4}$$

Each of the modular preference ($S_{d,j}$) is sorted by the depths (d), where j is the index of the subtree. The raw utility $U(.)$ is generated by Equation (3) with base z, where $z \geq 2$. As a result, the utility U is exponentially increasing when higher levels of modular preferences are satisfied. To demonstrate the utility calculation, we use the product in Fig. 3 and the idiosyncratic preference in Table 2 as an example. The first biggest commodity module is the commodity itself, which is $(F_7(F_9X_5X_{11})(F_{12}X_3(F_9X_3X_8)))$. However, it fail to match any one of preference module in Table 2. Then we try the next biggest commodity module, which is $(F_{12}X_3(F_9X_3X_8))$. It still match none of the preference module. The process stops when the next commodity module $(F_9X_3X_8)$ and $(F_9X_5X_{11})$ match with $S_{2,1}$ and $S_{2,2}$ respectively. It should also be noted that there is another commodity element X_3 matches preference at the first level of depth. As a result, as Equation (3) shows, the commodity total gets 5 units of utility from the idiosyncratic preference.

Table 2
An example of idiosyncratic modular preference ($z = 2$)

Depth (d)	Subtrees or terminals	$z^{(d-1)}$
1	X_3, X_5, X_8, X_{11}	1
2	$S_{2,1} = (F_9X_3X_8)$ $S_{2,2} = (F_9X_5X_{11})$	2
3	$S_{3,1} = (F_5X_3(F_9X_5X_{11}))$	4
4	$S_{4,1} = (F_2(F_9X_3X_8)(F_5X_3(F_9X_5X_{11})))$	8
5	$S_5 = (F_6X_3(F_2(F_9X_3X_8)(F_5X_3(F_9X_5X_{11}))))$	16

3 Market Selection and Consumer Satisfaction

Table 3
Parameters of Search Intensity

Parameter	Type (Variable)	Range	Value
	High Search Intensity		
Number of Firms	Integer (n_p)	$[1, \infty)$	2
Search Intensity	Real (r_s)	[0, 1]	100%
	Low Search Intensity		
Number of Firms	Integer (n_p)	$[1, \infty)$	2
Search Intensity	Real (r_s)	[0, 1]	50%

Note: The rest of the parameter settings are the same as Table 1.

This section explores the influence of two kinds of consumer search abilities, presented in Table 3. With low selection pressure, each consumer meets only one of the two firms. On the contrary, it is found that high selection pressure may foster the meeting of both firms on every trading day. With limited market demand, the higher the search intensity, the higher the market selection pressure.[7] For purpose of further analysis, statistics about consumer satisfaction in each generation are reported. Consumer satisfaction is normalized by dividing the consumer surplus received from consumption with the maximum potential surplus of the consumer, multiplied by 1000. In other words, the normalized value will lie in [0, 1000]. Averaging the consumer satisfaction over all 100 consumers, we then derive the aggregate consumer satisfaction, which also lies in the same interval. The result is shown in Fig. 5, which is the maximum average value of all the experimental runs. Since the variation of high search intensity case is much larger than the low search intensity one, we collected 100 runs for the former and 50 runs for the latter to achieve statistical validity. As illustrated in Fig. 5, consumer satisfaction rises with his or her own search ability. However, the survivability of these two firms is rather different. Both firms can survive under low selection pressure, but the situation changes dramatically under high selection pressure. Under the severe selection pressure scenario, usually one firm survives. Nonetheless, after examining the all experimental results, the chance of the success in either case do not have significant difference. In the next section, we will further investigate the parameters of genetic operator from a competitive advantage point of view.

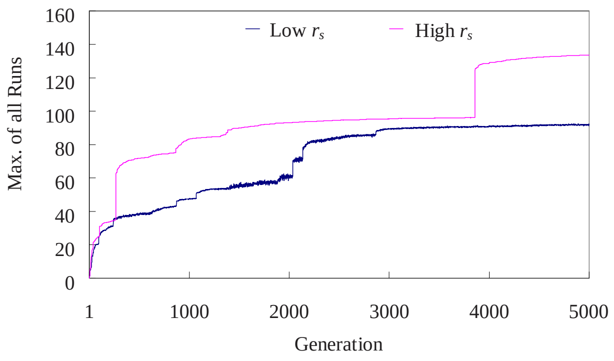

Fig. 5. Time series of consumer search intensity and consumer satisfaction.

Table 4
Parameters of Genetic Operator Rates

Parameter	Type (Variable)	Range	Value
	Crossover Rate Competition		
Number of Firms	Integer (n_p)	$[1, \infty)$	2
Crossover Rate	Real (p_c)	[0, 1]	(1) 45% (2) 90%
	Mutation Rate Competition		
Number or Firms	Integer (n_p)	$[1, \infty)$	2
Mutation Rate	Real (p_m)	[0, 1]	(1) 40% (2) 80%

Note: (1), (2) are the parameters for low-rate firm and high-rate firm respectively.

4 Genetic Operator and Competitiveness

As shown in the previous section, compared to the low selection pressure environment, high selection pressure not only brings more consumer satisfaction but increases the overall market competition as well. To understand how genetic operators may influence a firm's competitiveness and hence the level of market competition, we have further investigated the effects of crossover rates and mutation rates in a high selection pressure setting of two firms. We have considered the following two scenarios:

- Scenario 1: High crossover rate (90%) versus low crossover rate (45%)
- Scenario 2: High mutation rate (80%) versus low mutation rate (40%)

Table 4 has summarized the relevant parameters. [8] With simulation, the survivability (which is positively correlated to competitiveness) of the two firms under different settings can be unveiled through monitoring respective market share at different time horizon. Market share is calculated as the total sales of each firm divided by the total sales of the market. [9]

[7] Note that our consumers are allowed to buy at most one commodity each trading day.
[8] The rest of the parameter settings are the same as Table 1.
[9] As a firm may produce more than one product, its total sales are normally generated from a bundle of products. Similarly, the total sales of market is measured

In this study, we have simulated 100 runs for each scenario. Results of the two scenarios can be found in Fig. 6. Means of the 100 runs are presented on the left, whereas the medians are on the right. The shaded area represents the market share of the low-rate firm as time goes by; the complement is the market share of the high-rate firm. As can be seen, the low-rate firm seldom owns more than half of the market share and hence is dominated in a market of high competition. This coincides with our earlier finding that when the selection pressure is high, usually one of the two firms dominates the market. Despite the rates under investigation, it is also found that the behaviors of the two kinds of time-series are rather different. While the means are relatively smooth below the midline, the medians imply that competition may cause more extremity overtime.

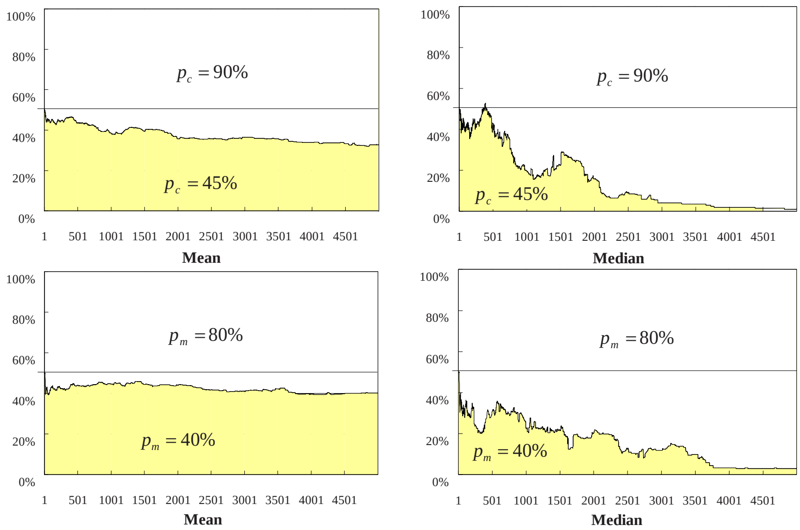

Fig. 6. Genetic operator rates and competitiveness.

In addition to the above findings, we would like to interpret the impacts of genetic operators from an economic point of view. Crossover describes the activity of exchanging information, whereas mutation is the process of deviation from former position. Higher crossover rates raise the probability of information-exchanging and encourage recombining innovative ideas. Although crossover and mutation sometimes may bring destructive results, they have been found useful in discovering and preserving good ideas. This has also been suggested by Goldberg (2002) in his book "The Design of Innovation."

With genetic operators, it is possible to simulate the impacts of various organizational cultures or attitudes. For example, a progressive (e.g. high-rate) firm may supply more innovative commodities to the market and face more uncertainty in selling those products. Meanwhile, a conservative (e.g. low-rate) firm may adhere to former designs, failing to catch up with market trend. As

based on all the available products in the market at a particular point of generation.

a result, it is likely that a low-rate firm would be driven out of market due to loss of consumer satisfaction.

5 Concluding Remarks

Consumers and firms behavior can be studied and the patterns can be extracted. In this paper, we studied the significance of genetic operators in product innovation and market selection. Two observations have been found in this study. First, selection pressure is positively correlated with consumer satisfaction. The more the pressure experienced in a firm's production, the higher the satisfaction level of a consumer can achieve. Selection pressure is defined in terms of the number of choices consumers have. If no choice being granted, products with low quality or little invention may still be selected. However, the more the choices, the better a consumer becomes enhanced of his or her own satisfaction.

In addition, parameters of genetic operator are crucial in determining the competencies of producers. In our agent-based two-firm model, two producers with different crossover and mutation rates compete with each other in innovation. It is found that product innovation takes not only modularity but also proper settings of genetic operator. Various parameters of genetic operator may represent different organizational cultures. This mechanism provides an opportunity to observe the competition between an open-minded culture and a more conservative one.

References

Chen, S.-H., Chie, B.-T. (2004) Agent-Based Economic Modeling of the Evolution of Technology: The Relevance of Functional Modularity and Genetic Programming. International Journal of Modern Physics B, **18**, 17-19, 2376–2386.

Chen, S.-H., Chie, B.-T. (2006) A Functional Modularity Approach to Agent-based Modeling of the Evolution of Technology. In: A. Namatame, T. Kaizouji, and Y. Aruka. (eds.): The Complex Networks of Economic Interactions: Essays in Agent-Based Economics and Econophysics, Lecture Notes in Economics and Mathematical Systems Vol. 567, Springer-Verlag, 165–178.

Chen, S.-H., Chie, B.-T. (2007) Modularity, Product Innovation, and Consumer Satisfaction: An Agent-Based Approach, In: 8th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'07), Birmingham, UK, December 16-19.

Goldberg, D. E. (2002) The Design of Innovation: Lessons from and for Competent Genetic Algorithms. Kluwer Academic Publishers, Boston, MA.

Koza, J.R. (1992) Genetic Programming: On the Programming of Computers by the Means of Natural Selection. The MIT Press, Cambridge.

Koza, J.R. (1994) Genetic Programming II–Automatic Discovery of Reusable Programs, The MIT Press, Cambridge.

A Appendix

Pseudo Code:

```
Initialize Market, Consumers, and Firms
Ask Firms [ Initialize Commodities
           Pricing Commodities
           Waiting for Consumers ]
Do While day mod 5 <> 0 [
Ask Consumers [ Decide Shopping List
               Visit Firms on Shopping List
               Buying Commodities
               Update Experiences of Visited Firms
               day ++ ] ]
Do While generation < 5000
[
  generation ++
  Ask Firms [ Acquire Statistics
             Adjust Inventory
             Innovation
             Pricing Commodities
             Waiting for Consumers ]
  Do While day mod 5 <> 0 [
  Ask Consumers [ Decide Shopping List
                 Visit Firms on Shopping List
                 Buying Commodities
                 Update Experiences of Visited Firms
                 day ++ ] ]
]
```