Truckin' Project

A Successful Experiment with Genetic Algorithms

Yi Ling Ni

A Thesis

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

January 2004

Your file Votre référence
ISBN: 0-612-91095-4
Our file Notre référence
ISBN: 0-612-91095-4

ABSTRACT

Truckin' Project

A Successful Experiment with Genetic Algorithms

Yi Ling Ni

Genetic algorithms use natural selection and reproduction to search for good solutions to a problem from among a great number of possible solutions. Since they were introduced by John Holland in the early seventies, they have been used in a variety of areas to solve complex problems because genetic algorithms are more robust, more efficient and more flexible than conventional artificial intelligence techniques.

Truckin' project uses genetic algorithms to simulate a country where trucks and retailers compete to make profit. The behavior of each truck or retailer is decided by a genome. This thesis does three things on the basis of old Truckin' project. It first enhances trucks' and retailers' performance by adding a new gene to each of them. And then it allows some successful retailers to open new branches to see if other retailers are forced to go bankrupt in the competition. Finally, it analyzes trucks' profit composition and evaluates each gene's performance in the simulation.

The simulation result proves the successful use of genetic algorithms in the Truckin' project.

Acknowledgements

This thesis wouldn't be without the support of my supervisor, Dr. Peter Grogono. Thanks for his unwearied guidance and indispensable patience.

Table of Contents

List of Figures

List of Tables

1 Introduction

Genetic algorithms use simulated natural selection and reproduction to search for good solutions to a problem from among a great number of possible solutions. Since they were introduced by John Holland in the early seventies, they have been used in a variety of areas to solve complex problems because genetic algorithms are more robust, more efficient and more flexible than conventional artificial intelligence techniques.

Truckin' project is a platform on which to practice genetic algorithms. It was first presented by Mark Stefik and others at Xerox Palo Alto Research Center (PARC) during the eighties. It simulates a country in which trucks and retailers try to make a profit by trading commodities. Genetic algorithms are used in this project to evolve trucks and retailers.

Three people previously contributed to Truckin' project under the supervision of Dr. Peter Grogono. They are Jeffrey Edelstein, Debbi Papoulis, and Qixia Deng. After Truckin' came to my hand, I first added a new gene to trucks, Gpopulation, so that trucks' population can be controlled by trucks themselves. Secondly, I increased retailers' strategy by adding a new gene, Gcapacity. And then, I allowed some of the most profitable retailers to open branches at empty intersections. After that, the composition of trucks' profit was analyzed. Finally, I investigated trucks' and retailers' genes one by one.

This thesis is structured in the following manner: Chapter 1 is an introduction of the whole thesis; Chapter 2 introduces genetic algorithms; Chapter 3 introduces the Truckin' project and the related work done by previous students; Chapter 4 describes my enhancement to the project which are respectively adding a gene to trucks and retailers, and allowing some most profitable retailers to open new branches. In this section, the result of each enhancement is recorded in detail. Chapter 5 analyzes the composition of trucks' profit and the performance of trucks' and retailers' genes. Conclusions are in Chapter 6.

2 Genetic Algorithms

This chapter reviews the basic theory of genetic algorithms and provides a general overview of genetic algorithms. It intends to give readers a rough idea of genetic algorithms.

2.1 What Are Genetic Algorithms?

Genetic algorithms (Gas) are heuristic optimization methods introduced by John Holland in the early seventies. They use simulated natural selection and reproduction to search for good solutions to a problem from among a great number of possible solutions.

GAs search methods are based on a population of candidate solutions. Initially, these solutions are selected randomly. The solutions are evaluated for their "fitness". Bad solutions tend to die away while good solutions mate to produce better solutions of the next generation using three genetic operators: *selection*, *crossover*, and *mutation*. This cycle is repeated until the best solution is good enough or the time limit is exceeded.

2.2 Representation

A solution is represented by a chromosome which contains a set of genes. Each gene decides a specific attribute of the solution, and is normally coded by one or more binary bits. The number of bits depends on the problem's complexity. The combination of all genes' binary bits comprises a binary string. So we can say that a chromosome is often represented by a binary string.

2.3 Three Basic Operations

The basic operations of genetic algorithms are:

- Selection – Selection is the process of determining which solutions are chosen to reproduce new solutions. A solution with higher fitness has a higher probability of reproduction over others with less fitness.
- Crossover – Crossover is the process of taking two parent chromosomes to reproduce an offspring chromosome. A commonly used crossover method is one-point crossover. In this method, a split point is chosen randomly to divide each parent's chromosome into two parts: left sub-chromosome and right sub-chromosome. The offspring chromosome is created by combining the left sub-chromosome of one parent and the right cub-chromosome of another parent.
- Mutation – Mutation is the process of randomly choosing some bits of offspring chromosome to flip. Mutation adds "fresh blood" to a population. But the mutation probability should be kept very low since a high mutation rate will destroy fit strings and degenerate the GA algorithm into a random walk.

2.4 Basic Process

The process of a genetic algorithm is roughly as follows:

1) Create an initial population of randomly-generated individuals.
2) Evaluate each individual's fitness.
3) Kill the bottom x% (least fit) of the population.
4) Choose two members for mating.

5) Crossover.
6) Apply mutation.
7) Go back to step 4 until new population is created.
8) The children and remaining good individuals form the new population of solutions.
9) Go back to step 2 until the best solution is good enough or the time limit is exceeded.

2.5 Why Genetic Algorithms?

Genetic algorithms provide robustness, efficiency and flexibility when searching for the optimum solution in a searching space.

1. Genetic algorithms are robust because exploration does not depend on continuous search spaces.
2. Genetic algorithms are efficient because they deal with a population of solutions to conduct a search, not just a single one on the search space. Instead of relying on a single point to search through the space, genetic algorithms look at many different areas of the problem space at once. This avoids the problem of finding a local maximum.
3. Genetic algorithms are flexible because they can easily be written for parallel processing.

3 Truckin' Project

This chapter first introduces the main idea of Truckin' project (the detailed specification of truckin' project, please refer to [1]), and then provides an overview of the work that has been done by previous students is taken.

3.1 Introduction

Truckin' is a test case for artificial intelligence. It was first presented by Mark Stefik and others at Xerox Palo Alto Research Center (PARC) during eighties. It simulates a country around which trucks travel trying to make a profit by trading with dealers.

In truckin' project, the country is a stage. All events happen on it. There are four kinds of dealers in the country: producers, retailers, consumers, and gas stations. The country contains ten avenues (north - south) and ten streets (east – west) in a form of a grid. Each intersection is either empty or occupied by one dealer. Commodities are distributed by trucks.

3.1.1 Commodities

There are three types of commodities in the country: crates, items, and gas. Crates are first produced by producers, and sold to trucks. Trucks then sell crates to retailers. Retailers unpack crates they bought; each crate contains 100 items. Trucks buy those items from retailers and finally sell them to consumers. Trucks consume gas while they travel around the country. Gas is stored in gas stations. Trucks come to gas station to buy gas from time to time.

3.1.2 Trucks

All trucks travel from one intersection to another. They buy crates from producers, sell them to retailers, buy items from retailers, and sell them to consumers at last. Trucks must be aware of their gas consumption. In order to have enough energy to support their travel, they have to go to a gas station to buy some gas when necessary.

Trucks make a profit from the difference between the crate selling price and the buying price, and the difference between the item selling price and the buying price. And every truck must keep some amount of capital to buy gas. A truck will be removed when it goes into debt. At the end of each generation, some profitable trucks are chosen to reproduce new trucks.

3.1.3 Dealers

There are four types of dealers: producers, retailers, consumers, and gas stations. The total number of them must be less than or equal to the number of intersections because each intersection is allowed to contain at most one dealer. Once a dealer is created and positioned, it can not be moved until it is killed for some reason.

- Producers - The producers' unique function is to produce crates for the country. The producing rate is fixed. The stock capacity is viewed as infinite. Producers sell crates to trucks at a fixed price (In the simulation, the crate selling price is 60$ per crate). Producers are created at the beginning of the simulation and their lives last to the end of the simulation. In the simulation, the number of producers is predetermined.

- Retailers - Retailers buy crates from trucks, unpack crate into items, and then sell items to trucks. Each retailer has a storage capacity and it pays rent for it. Firstly, in order to make a profit, a retailer must keep its crate buying price less than its item selling price multiplied by 100 (the number of items in one crate). Secondly, in order to survive, a retailer must have the ability to pay rent periodically. A retailer will be removed when it goes into debt. At the end of each generation, some profitable retailers are chosen to reproduce new retailers.
- Consumers - Consumers purchase items at a fixed price: $1 per item. They have no capacity limit, which means they buy items whenever trucks come to them asking for making a deal. Consumers are created at the beginning of the simulation and are killed at the end of the simulation. In the simulation, the number of consumers is predetermined.
- Gas Stations - The gas stations' single task of is to sell gas to trucks at a fixed price: $1 per litre. Gas stations have unlimited capacity and the amount of gas they sell to trucks just depends on trucks' request. Gas stations are created at the beginning of the simulation and are killed at the end of the simulation. In the simulation, the number of gas stations is predetermined.

So, the potential profit made from one crate is the difference between customers' item buying items and producers' crate selling price, which is

100 items/per crate * $1 - $60 = $40

Trucks and retailers compete to share this profit as much as possible.

3.2 An Overview of Previous Work

Three graduate students, Jeffrey Edelstein[4], Debbie Papoulis[5], and Qixia Deng[6] contributed their theses to the Trukin' project under the supervision of Dr. Peter Grogono. This section briefly introduces their work.

3.2.1 Jeffrey Edelstein's Contribution [4]

The truckin' project started as a coding comprehension to find good truck strategies. Several genes were defined to simulate trucks' behavior.

In the simulated country, apart from trucks, three types of commodities, and four types of dealers, Jeffrey Edelstein introduced two other components: controllers and managers.

- Controllers – Each truck is assigned a controller which is used to provide some services to trucks. It "limits a trucks' behavior and prevents cheating" [4]. For example, the controller limits the number of crates a truck can purchase at one time based on the amount of capital the truck has and the available stock capacity.
- Managers – Managers provide the same service to retailers as controllers do to trucks. "All deals between a truck and a dealer are mediated by a controller and manager to prevent cheating in the competition" [4].

Jeffrey Edelstein brought forward eight genes for trucks. They are illustrated in Figure 3-1, according to which the search space is:

$$2 * 4 * 4 * 3 * 3 * 1 * 2 * 5 = 2880$$

Figure 3-2 shows the running result (The Y-axis represents the total profit earned by all trucks in each generation). Edelstein thought it was not ideal because the total profit didn't constantly increase. The probable explanation he gave is: "The population of trucks with a generation becomes smarter and the gap between the stronger and weaker trucks narrows" [1].

Gene Number	Truck Trait	Gene Values
0	Init	0,1
1	Gas	0,1,2,3
2	Trade	0,1,2,3
3	Buy	0,1,2
4	Sell	0,1,2
5	Go	0
6	Move	0,1
7	Deal	0,1,2,3,4

Figure 3-1 Trucks' Eight Genes Presented by Jeffrey Edelstein [4]

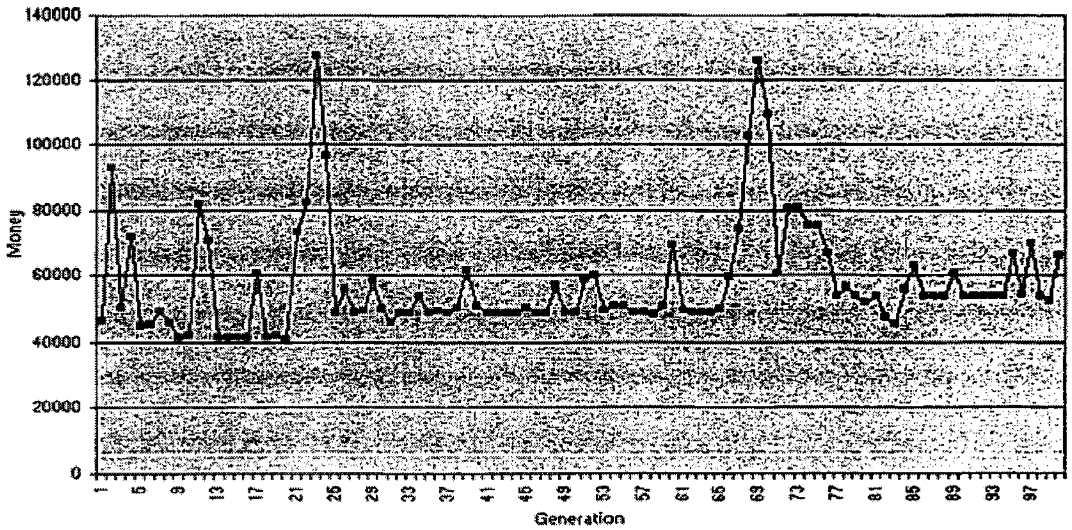

Figure 3-2 Simulation Result by Jeffrey Edelstein [4]

I think of another reason that might cause this. The number of trucks in each generation is small. Edelstein created 10 trucks to travel around the country in each generation. Since one simulation comprised 100 generation, the maximum solution number explored in one simulation is only 500 which are relatively small compared with the search space (2880).

3.2.2 Debbie Papoulis' Contribution [5]

Debbie Papoulis worked on Truckin' project after Jeffrey Edelstein. She removed controllers and managers from the simulation. All functions controllers were integrated in trucks; managers in retailers. Trucks' eight genes were kept the same.

Papoulis examined the affects of four parameters to the simulation result. Those four parameters and the affects she found were:

- Start positions of trucks – The trucks' start position affects the trucks' performance. The trucks starting in the middle of the country are the most profitable.
- Number of trucks in one generation – The number of trucks in one generation also affects the result greatly. More trucks mean that more "winning" combinations are likely to be explored. But too many trucks competing for limit commodities cause the decrease of trucks' average profit they make. Her experiment showed that the simulation with 100 trucks in each generation run most successfully.
- Number of generations in a simulation – The number of generations in a simulation doesn't seem to affect the amount of money that trucks can make. But it affects the amount of convergence. "The more generations in the simulation, the more of the trucks in each generation converge to a certain gene for some of the strategies" [5].
- Time length for one generation – The time length for one generation seems to affect the simulation result. Trucks make more profit if they are allowed to run longer time. But the amount of convergence isn't affected by the time length for one generation.

Papoulis also found some sign of convergence in the simulation.

3.2.3 Qixia Deng's Contribution [6]

Qixia Deng took a big step forward. Firstly, the number of genes every truck had was increased from 8 to 16, as shown in Table 3-1. Secondly, retailers participated in evolution. Three new genes were introduced for retailers, as shown in Figure 3-2. Finally, two different economy types were explored: supply-driven and demand-driven. In a supply-driven economy, the amount of crates circulating in the country depends on producer's production rate. In a demand-driven economy, the amount of items circulating in the country depends on consumer's consummation rate.

Table 3-1 Summary of Trucks' Genes [6]

Gene number	Symbol	Strategy	Value Range	Number of Values
1	G_g	Lowest gas limit	[0, 20]	21
2	G_s	Selling price of crates	[0, 255]	256
3	G_b	Buying price of items	[0, 255]	256
4	G_α	Capital transfer proportion	[0, 100]	101
5	$G_{capacity}$	Carrying capacity	[1, 10]	10
6	G_{tank}	Gas tank capacity	[1, 60]	60
7	$G_{reserve}$	The amount of money reserved for buying gas	[0, 2]	3
8	G_{corner}	Move to the nearest corner or not	[0, 1]	2
9	G_{scan}	Scan the country or not	[0, 1]	2
10	G_{deal}	How to look for deals	[0, 2]	3
11	$G_{priority}$	Priority of the deals	[0, 2]	3
12	G_{best}	The criteria of the best retailer	[0, 2]	3
13	G_{phone}	How to use the cell phone	[0, 2]	3
14	$G_{badsell}$	Permit unprofitable sells or not	[0, 1]	2
15	G_{badbuy}	Permit unprofitable buys or not	[0, 1]	2
16	$G_{congestion}$	How to response to traffic congestions	[0, 2]	3

Table 3-2 Summary of Retailers' Genes [6]

Gene number	Symbol	Strategy	Value Range	Number of Values
1	G_b	Buying price of crates	[0, 255]	256
2	G_s	Selling price of items	[0, 255]	256
3	G_α	Capital transfer proportion	[0, 255]	256

Simulating results are depicted from Figure 3-3 to Figure 3-6. Deng concluded that "on average, demand-driven economy can support more trucks running in the country than supply-driven economy" [6].

Deng also explained why the profit curve fell into a repeating peak-lull pattern. She thought the main reason was traffic congestion because she discovered a pattern: the time when truck number arrived at the peak was always the time when TPT (Total Profit of Trucks) reached the bottom. Traffic congestion caused the decrease of TPT in two ways. First of all, trucks which had been blocked by the traffic for more than 2,000 time units were removed. Secondly, "congestions brought poor performance of the remaining trucks, which further decreased the number of surviving trucks and newborn trucks" [6]. Her experiments showed that heavy congestion tended to happen when there were more than 500 trucks in the country corresponding to an average of five trucks at each intersection.

The TPR (Total Profit of Retailers) curve was similar to the TPT (Total Profit of Trucks). The retailers' profit curve was in the peak-lull pattern as well because the amount of deals they made with trucks decreased when congestion happens.

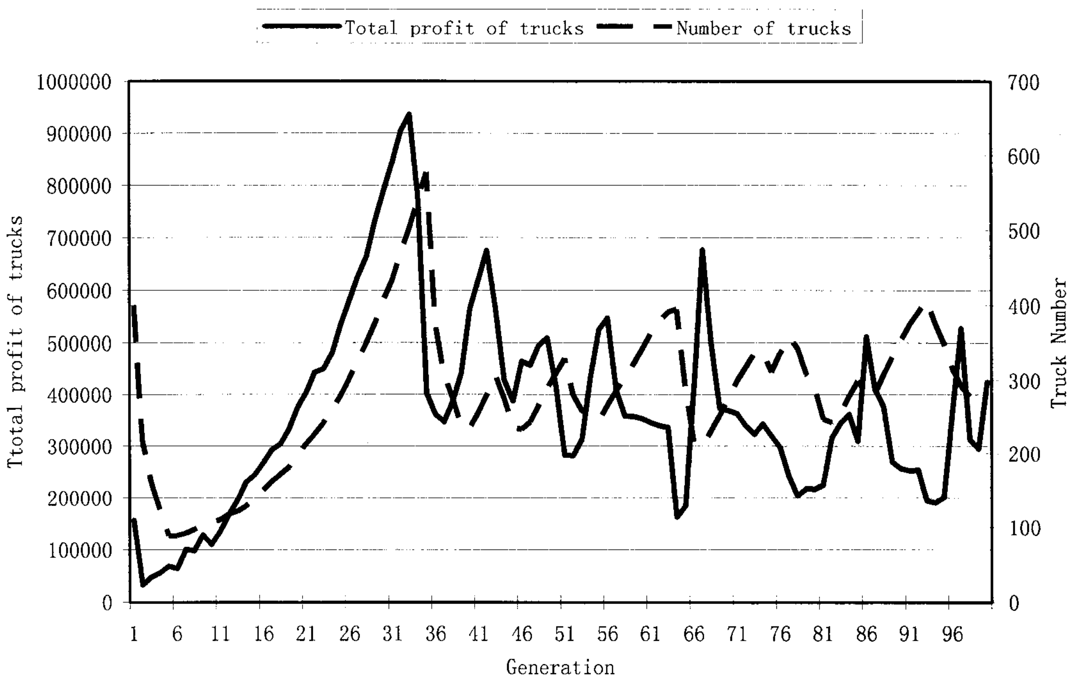

Figure 3-3 TPT and NT in a Supply-driven Economy [6]

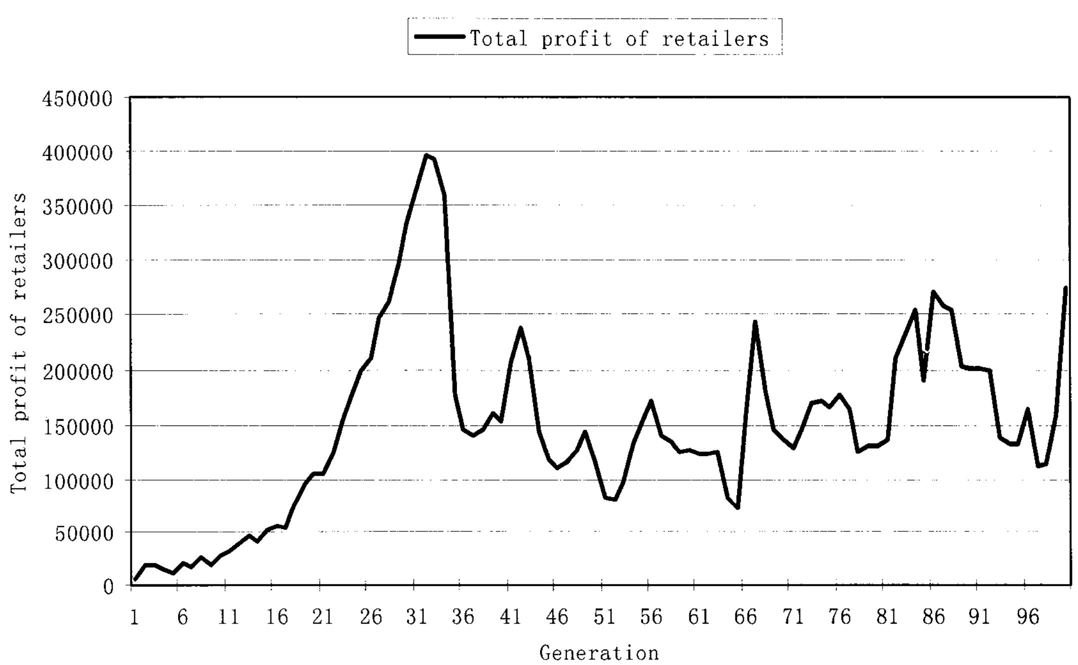

Figure 3-4 TPR in a Supply-driven Economy [6]

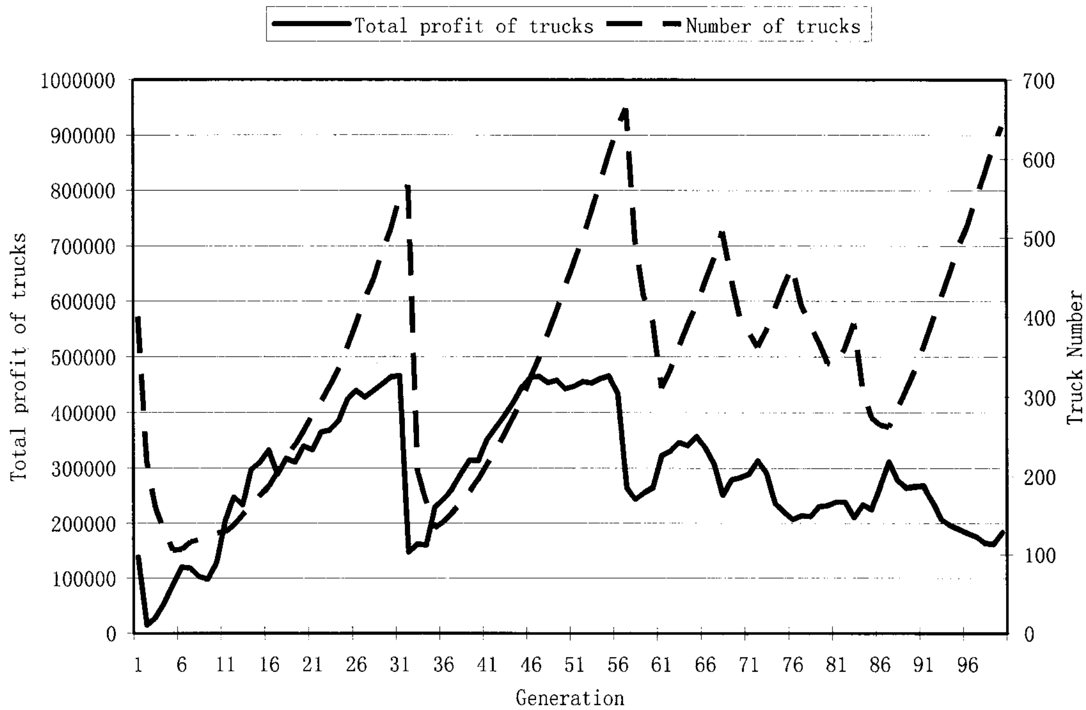

Figure 3-5 TPT and NT in a demand-driven economy [6]

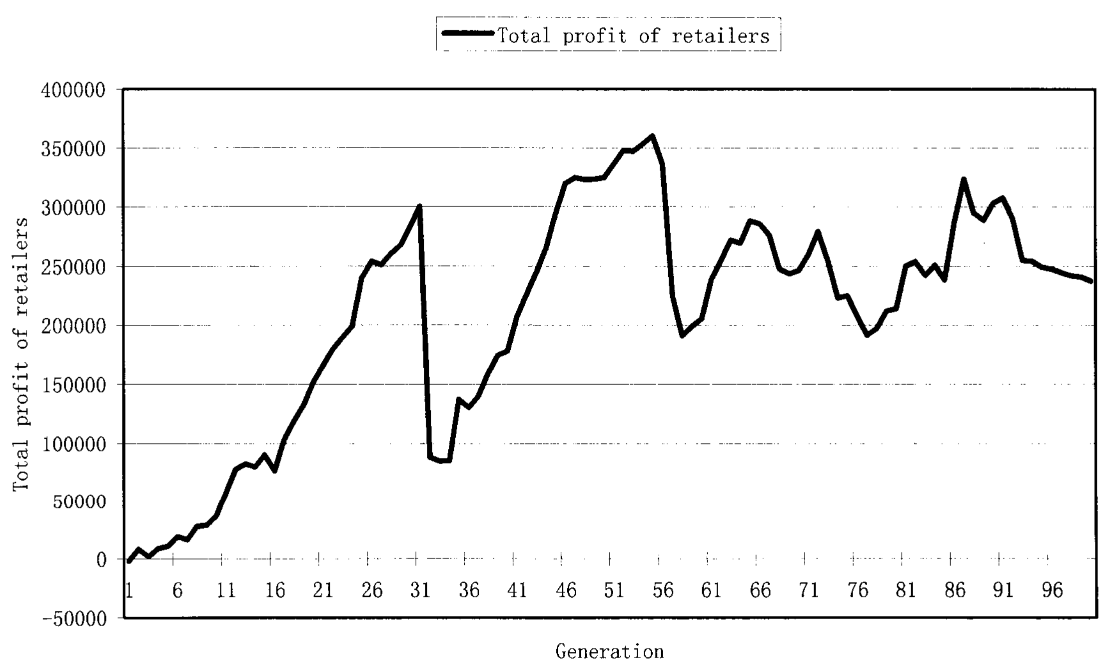

Figure 3-6 TPR in a Demand-driven Economy [6]

3.3 Related Work

GAs have been used for problem-solving and for modeling. GAs are applied to many scientific, engineering problems, in business and entertainment, including: optimization, automatic programming, machine and robot learning, economic models, immune system models, ecological models, population genetics models, interactions between evolution and learning, models of social systems [14].

The Traveling Salesman Problem (TSP) is an interesting test case for genetic algorithms. In this hypothetical problem, a salesman must visit a number of cities once and return to the city of origin. The distances between the cities are known. The salesman must determine the order of visiting each city to minimize the total distance. Many different genetic algorithms have been presented to solve the TSP, such as a "knowledge-augmented" genetic algorithm presented by Rong Yang [11], an "adaptive multi-start" genetic algorithm presented by Dan Bonachea, Eugene Ingerman, Joshua Levy, and Scott McPeak [12], and a "multilevel" genetic algorithm presented by Noureddine Bouhmala [13].

In the normal GAs, only the better performing individuals are allowed mate to generate new children. Some researchers have been trying to expand the standard GAs. For example, Tapabrata Ray and K. M. Liew proposed another optimization algorithm based on the simulation of social behavior [15], which improved the performance of all individuals in every society through intra or intersociety information exchange. They demonstrated that "Social interactions enable individuals to adapt and improve faster than biological evolution based on genetic inheritance alone" [15].

4 Enhancements

This chapter introduces three enhancements I did with the Truckin' project: adding a new gene Gpopulation to trucks, adding a new gene Gcapicity to retailers, and allowing the most profitable retailers to open new branches.

4.1 Add a Gene 'Gpopulation' to Trucks

4.1.1 Goal

According to Deng's testing result, all the curves of truck number, trucks' total/average profit, and retailers' total/average profit are a peak-lull pattern. An obvious drawback of such pattern is that, from time to time, many trucks are removed not because of their bad behavior, but because of traffic congestion. This is unfair to them. Every truck should be given a fair opportunity to compete. But congestion prevents some trucks from doing so.

Congestion is caused by an excessive population. In order to avoid congestion, we need to find a way to control the truck population. I introduced a new gene, Gpopulation, to control the truck population.

The new gene 'Gpopulation' decides each trucks' local population limit (LPL).

$$LPL = Gpopulation$$

A truck's local population is the number of other trucks which are less than or equal to 10 kilometers away from it (Two adjacent intersections are 10 kilometers apart).

During the simulation, each truck checks its local population periodically. At the end of one generation, each truck's average local population is calculated. When a truck is selected to reproduce new truck, it compares the recorded average local population with its LPL. If the average local population is greater than LPL, it refused to reproduce. Otherwise, it did.

4.1.2 Result

After a gene 'Gpopulation' has been added, the simulation can be divided into three phases:

- Phase 1: The truck number drops rapidly because the initial trucks' genomes are created randomly. Many trucks are deleted because of their bad performance. This phase generally last about 4 or 5 generations.
- Phase 2: The truck number jumps in this phase. How long this phase takes mainly depends on the reproducing rate. Figure 4-1 and Figure 4-2 were obtained from three simulations. From Figure 4-1, we can see that the greater the reproduction rate is the shorter this period takes. Figure 4-2 shows that the influence of the new gene Gpopulation on the second phase is minor.

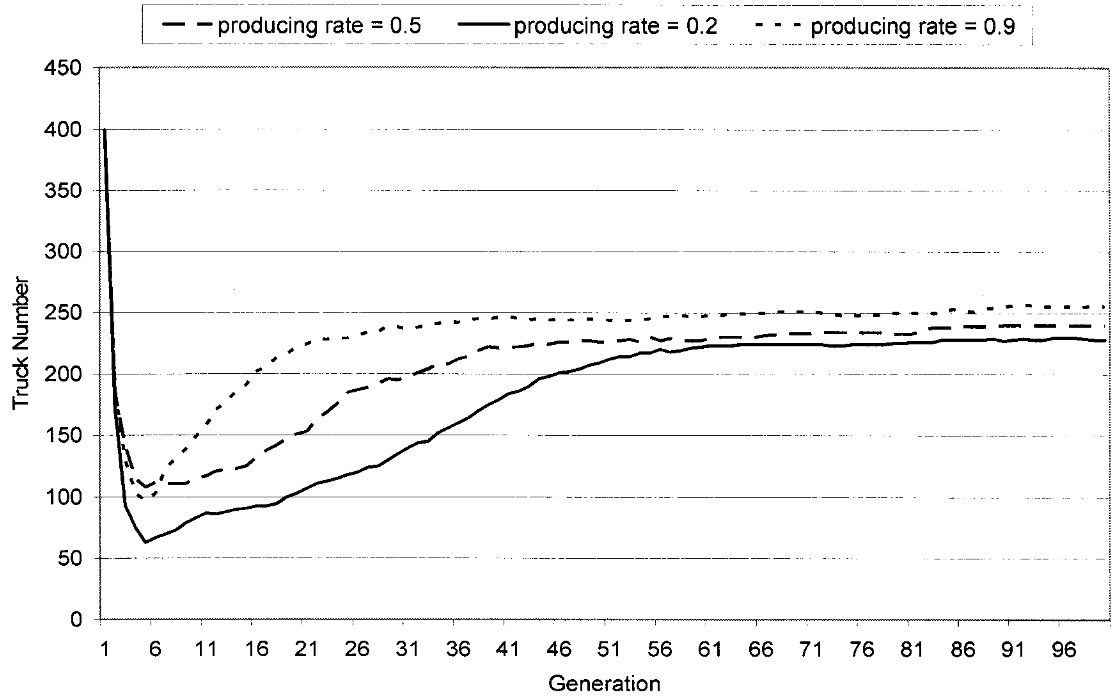

Figure 4-1 the Influence of the Reproduction Rate on Phase 2

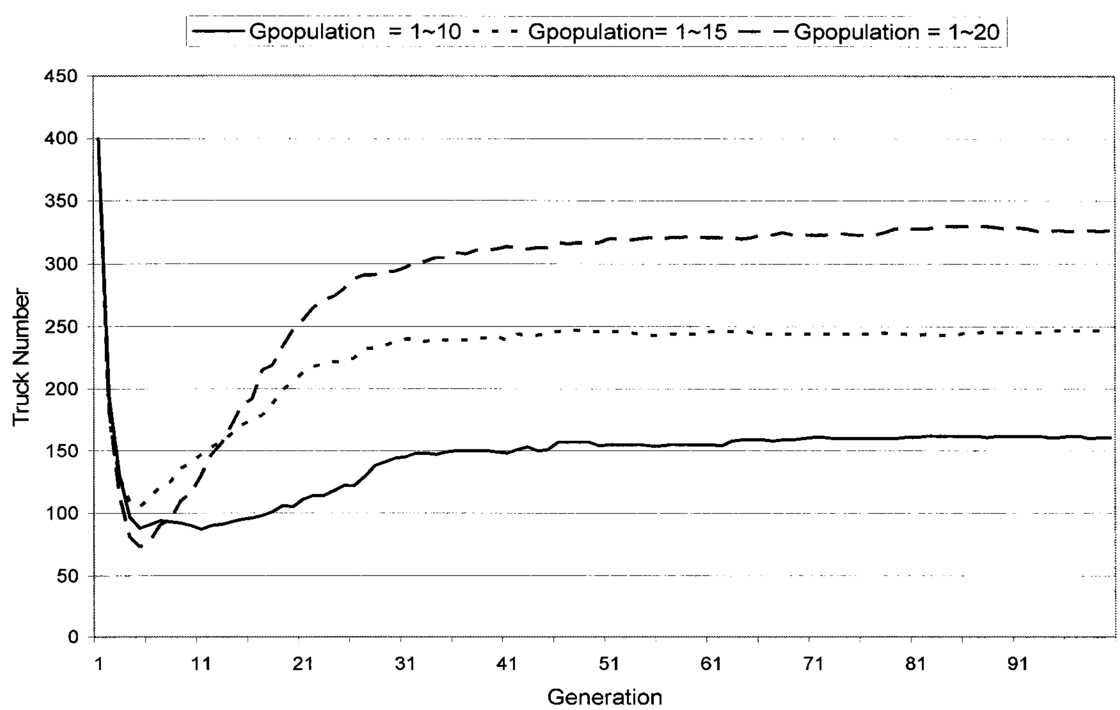

Figure 4-2 the Influence of Gpopulation on Phase 2

- Phase 3: In this phase, the truck number grows very slowly. If Gpopulation is in a reasonable range, the truck number can reach a stable value finally. Otherwise, the original pattern ("Congestions brought poor performance of the remaining trucks, which further decreased the number of surviving trucks and newborn trucks." [6]) is likely to come to play, as shown in Figure 4-3. Figure 4-3 was obtained when maximum Gpopulaiton was 50. In Figure 4-3, the spike near generation 80 was caused by traffic congestion. Hence Gpopulation has to be restricted in a suitable range so that the truck number can be completely controlled by LPL. Based on our experiments, it's best to set the maximum value of Gpopulation to be less than 25.

The maximum Gpopulation decides the final stable number of trucks. From Figure 4-2, we can see that the stable value of truck number increased along with the maximum Gpopulation. When the maximum Gpopulation was 10, the stable value was 161; when the maximum Gpopulation was 15, the stable value was about 247; the maximum Gpopulation of 20 resulted in the stable value of 327.

In each generation, the amount of commodities is fixed by producers. Hence, the trucks' average profit decreases when truck number increases. The following four Figures, from Figure 4-4 to Figure 4-7, tell us that it's appropriate to set the upper bound of Gpopulation to be less than or equal to 12.

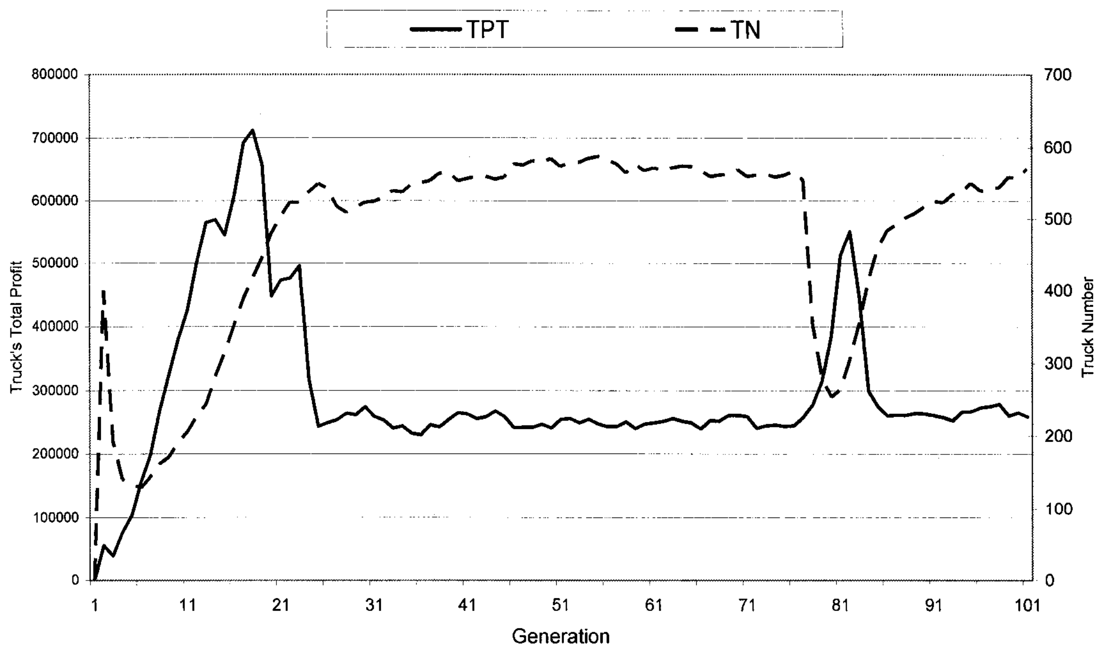

Figure 4-3 the Original Pattern Comes to Play

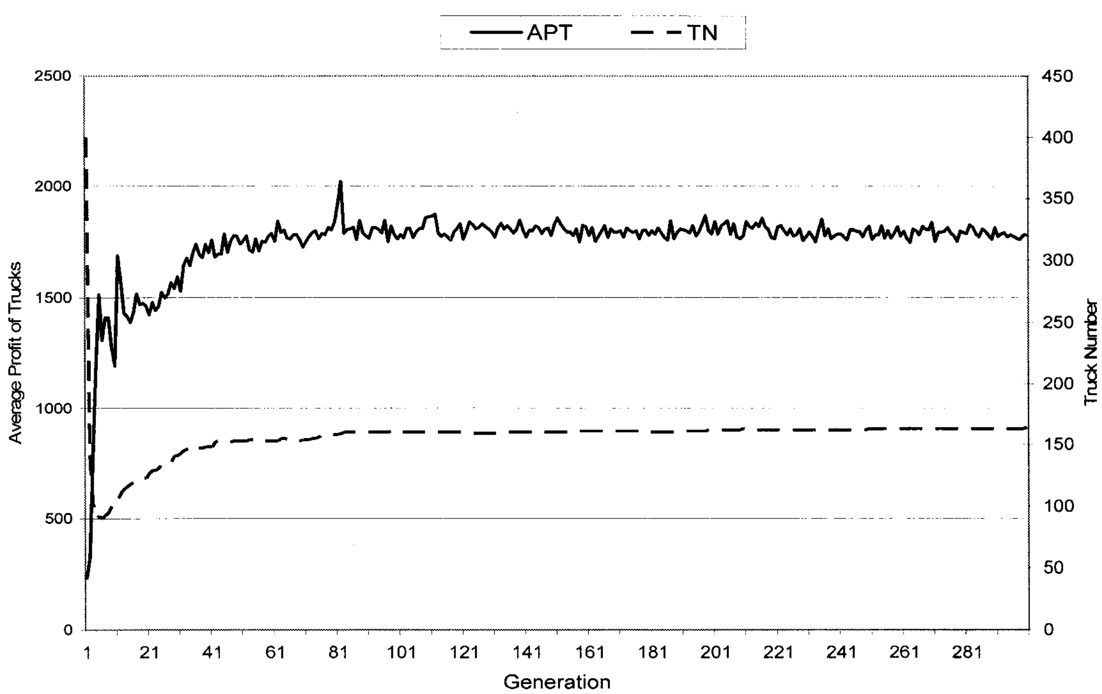

Figure 4-4 Gpopulation is between 1 and 10

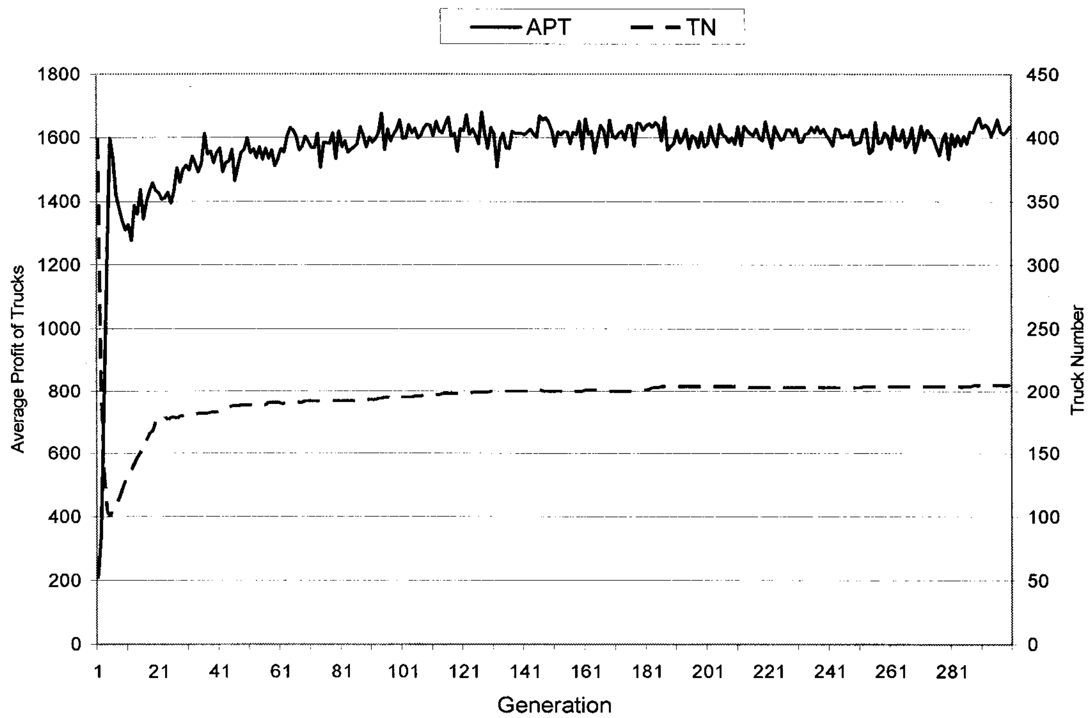

Figure 4-5 Gpopulation is Between 1 and 12

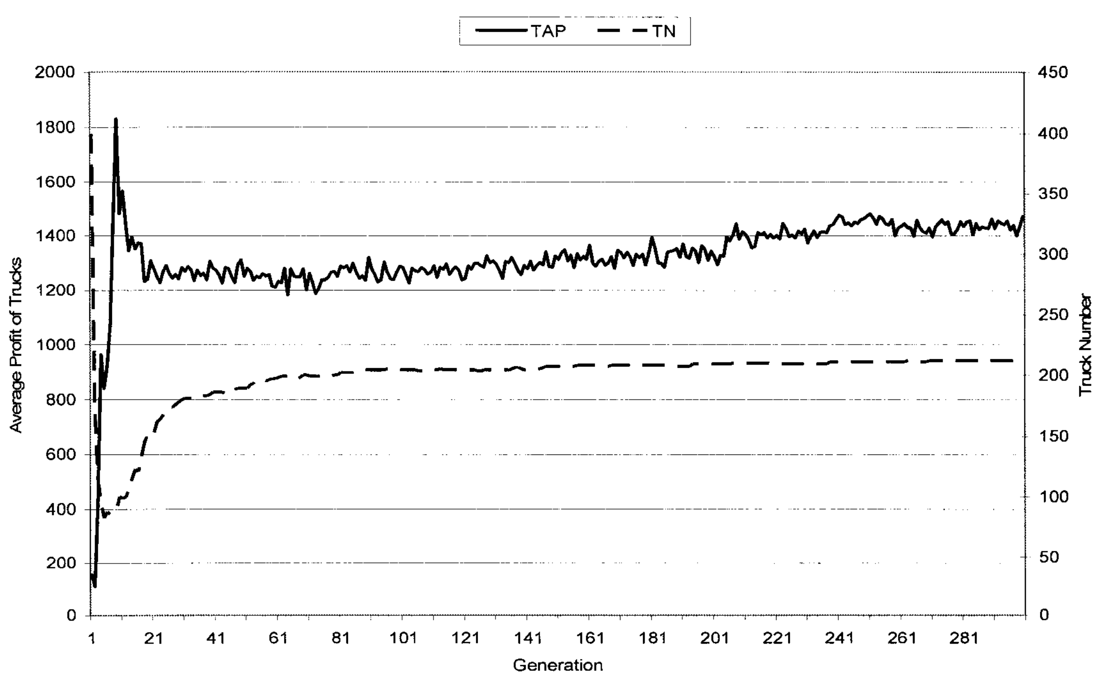

Figure 4-6 Gpopulation is Between 1 and 13

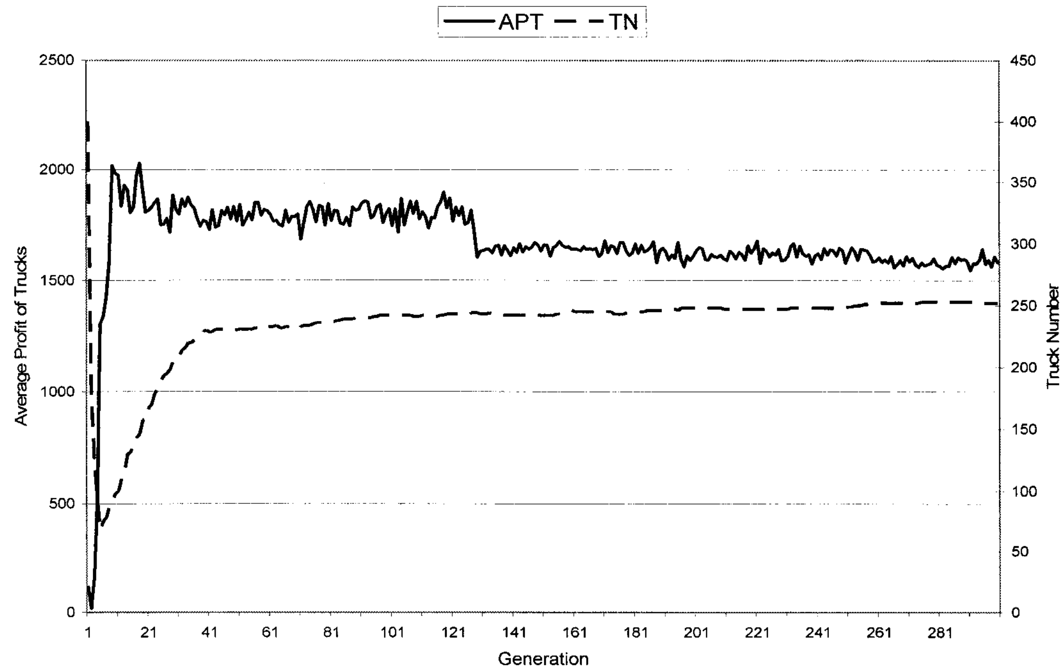

Figure 4-7 Gpopulation is between 1 and 15

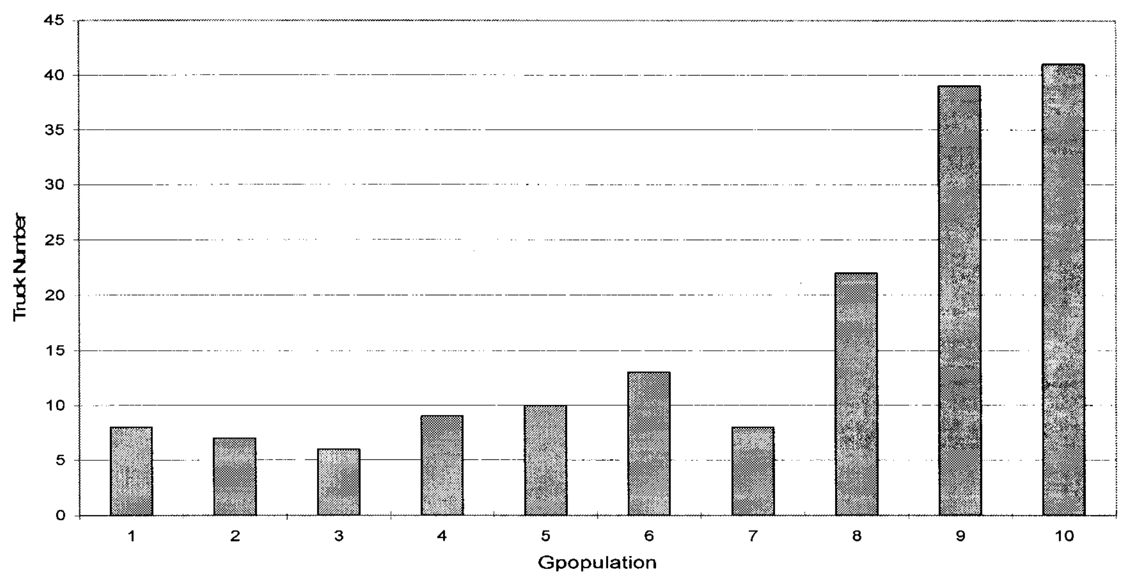

Figure 4-8 Truck Distributions According to LPL (1~10)

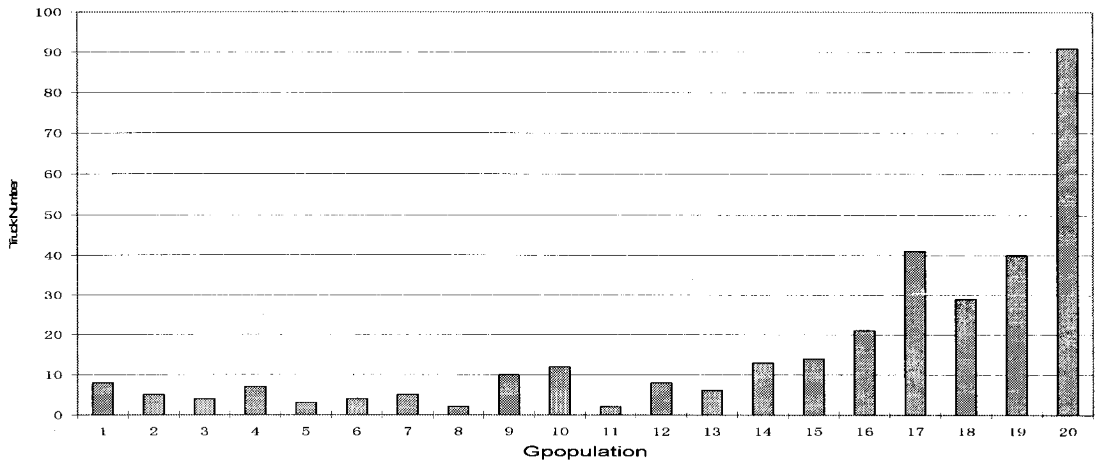

Figure 4-9 Truck Distributions According to LPL (1~20)

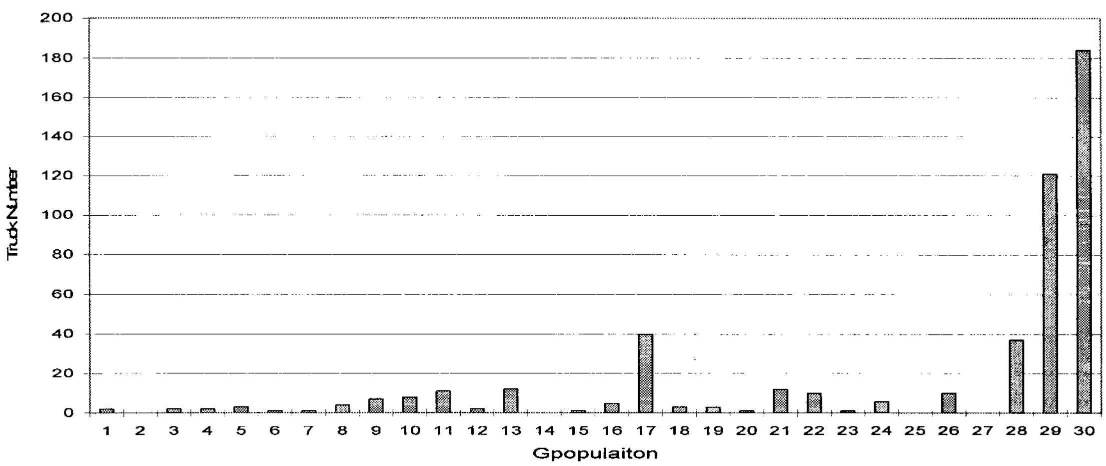

Figure 4-10 Truck Distributions According to LPL (1~30)

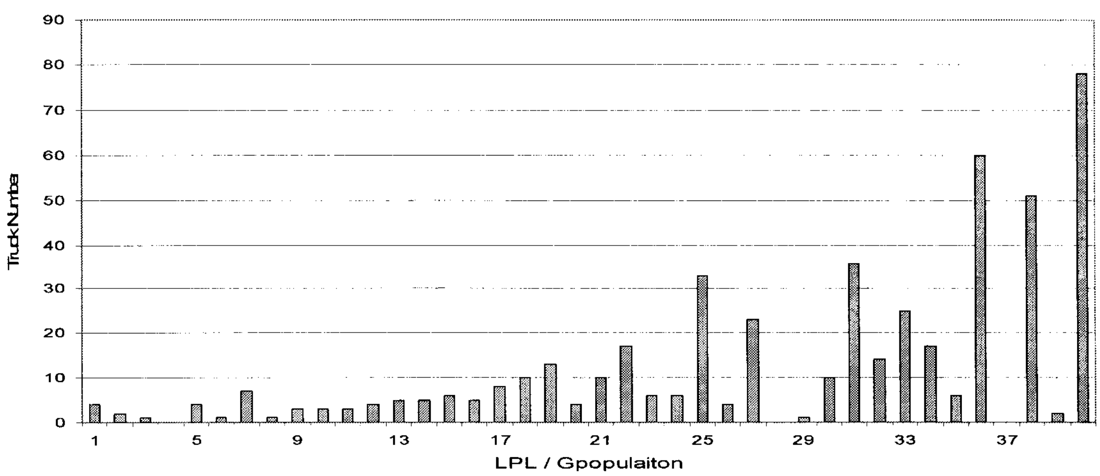

Figure 4-11 Truck Distributions According to LPL (1~40)

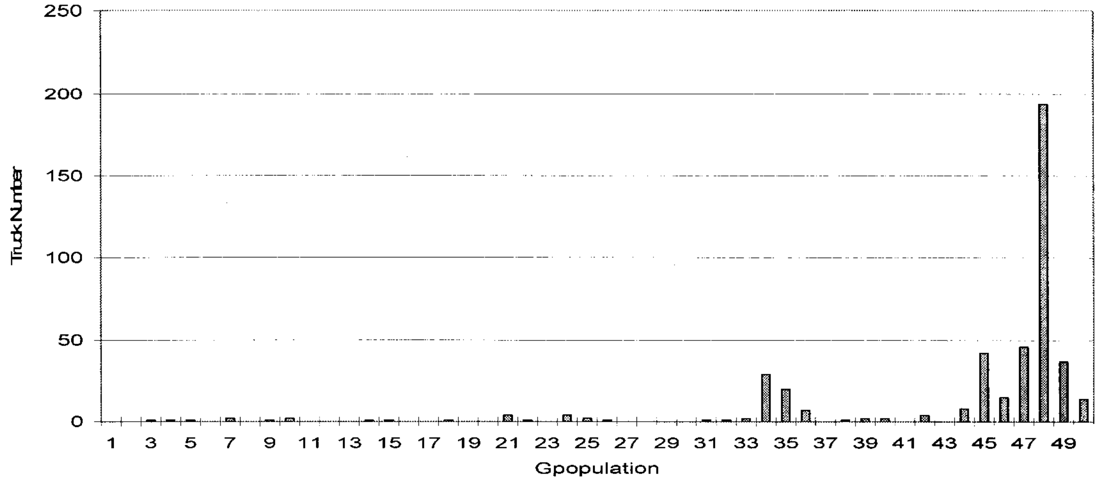

Figure 4-12 Truck Distributions According to LPL (1~50)

Figure 4-8 to Figure 4-12 show the trucks' distributions according to their new genes Gpopulation at the end of a simulation. We find that most trucks tend to distribute on high Gpopulation values. This is logical, because the trucks with higher gene Gpopulation have more chance to reproduce new trucks than those with lower Gpopulation, and further more, new trucks have a great chance to contain the same gene values as their parents because of the very low mutation rate.

But sometimes, the highest truck number doesn't appear at the highest Gpopulation value. Like Figure 4-12, the Gpopulation is between 1 and 50. But the highest truck number is at 48, not at 50. Why? In order to answer this question, we have to look through the whole simulation process where Figure 4-12 was obtained.

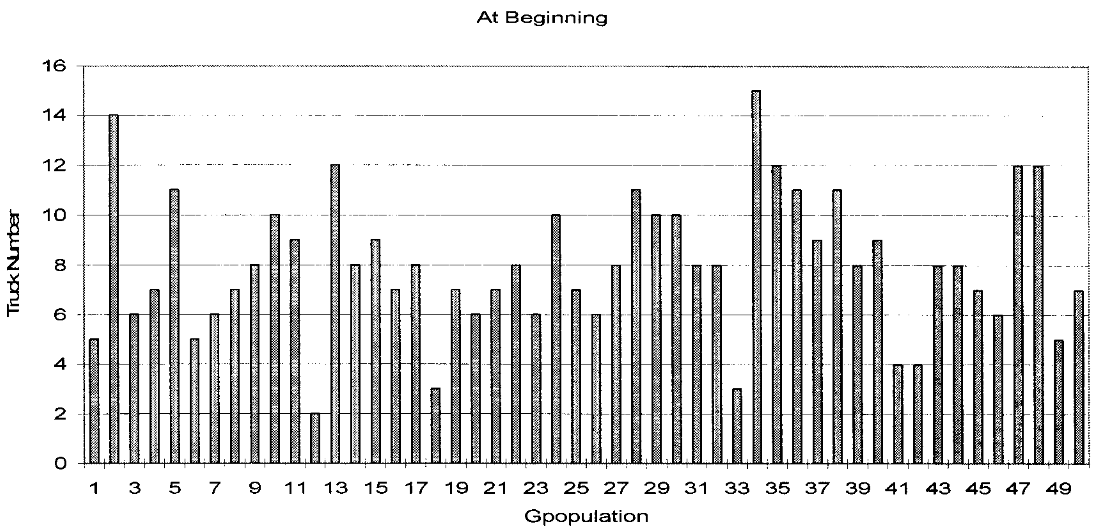

Figure 4-13 Truck Distributions At the very Beginning

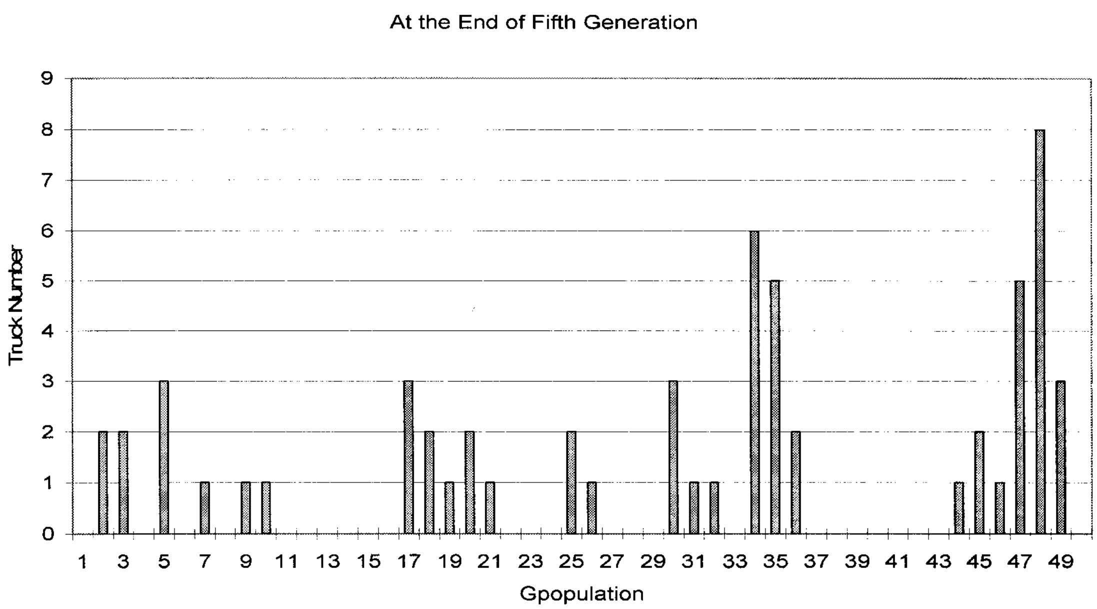

Figure 4-14 Truck Distributions At the end of the Fifth Generation

Figure 4-13 is the truck distributions at the very beginning of the simulation. The population of trucks is distributed randomly at first. Figure 4-14 is the distribution of Gpopulation at the end of fifth generation. Since generation 5 was the last generation of the first phase, the number of trucks in the fifth generation was the least of the whole simulation process. From Figure 4-14, we see that many trucks were removed because of their bad performance; the truck number at Gpopulation 48 was greater than at 49. There were in fact no trucks with the Gpopulation 50 because all trucks with Gpopulation 50 were removed.

Afterwards, the average local truck population increased along with the increase of truck number (in Figure 4-15). But except some spikes caused by congestion, in most generations, the average truck populations were below 48, which means in most generations, the trucks with Gpopulation greater than or equal to 48 didn't refuse to reproduce when they were selected to do so. Since trucks were selected randomly to reproduce, those trucks with Gpopulation 48 certainly had more probability to be selected than those with Gpopulation 49 and 50 because the base truck number with Gpopulation of 48 was greater than those of 49 and 50 in the fifth generation. That's why at the end of simulation, the highest truck number appeared at Gpopulation of value 48, not at the highest Gpopulaton.

4.1.3 Conclusion

The new gene, Gpopulaton, can control the trucks' population effectively. The trucks' population increases along with the increase of Gpopulation's upper bound. In order to avoid traffic congestion's influence to the simulation, the range of Gpopulation should be selected carefully. It's safe to set the upper bound of Gpopulation to be less than 25. But if we want to let trucks' average profit achieve the maximum, it's appropriate to set the upper bound of Gpopulation to be less than or equal to 12.

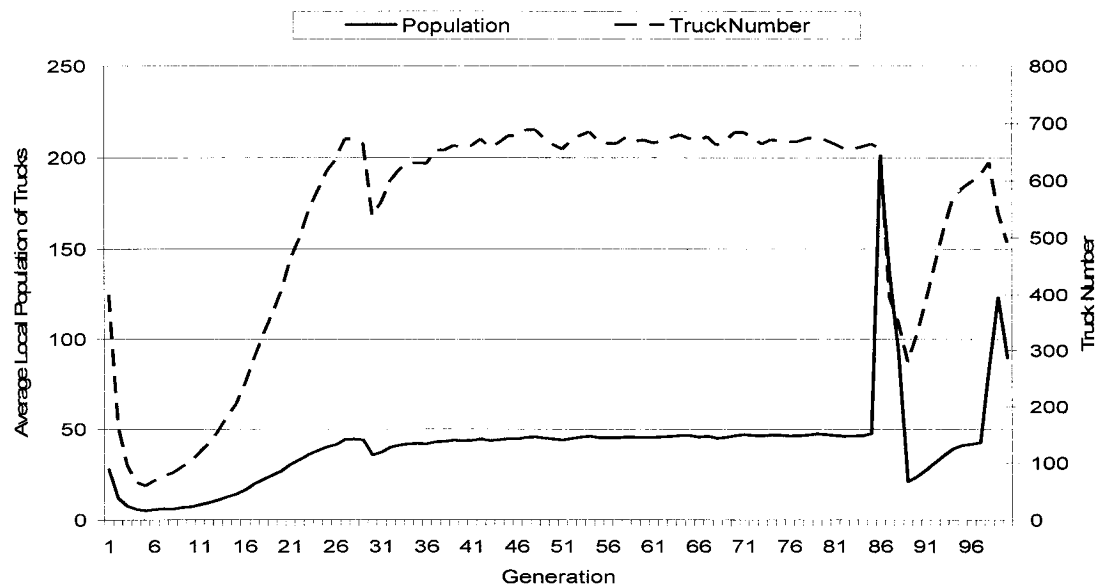

Figure 4-15 Average Local Populations of Trucks

4.2 Add a Gene 'Gcapacity' to Retailers

4.2.1 Goal

Previously, all retailers' storage capacities (SC) were the same: 8 crates. In order to increase retailers' strategies, a new gene, Gcapacity, was introduced to decide each retailer's storage capacity.

SC = Gcapacity

Gcapacity is between 1 and 10. Each retailer must pay rent for the stock capacity.

4.2.2 Result

When each retailer pay rent for its stock capacity periodically, there are two situations: the rent is fixed, which means all retailers pay the same rent no matter how big their storage capacities are; or the rent depends on the storage capacity.

Situation 1) the rent is fixed.

When the rent is low, the distribution of new gene 'Gcapacity' has no obvious patterns, as shown in Figure 4-16 to Figure 4-18.

From Figure 4-19 to Figure 4-22, along with the increase of the rent, retailers tend to concentrate on large Gcapacity values since large storage capacities enable them to make large deals to support the high rent.

At last, when the rent is high enough, the simulation has a great chance to terminate since no retailers can afford it.

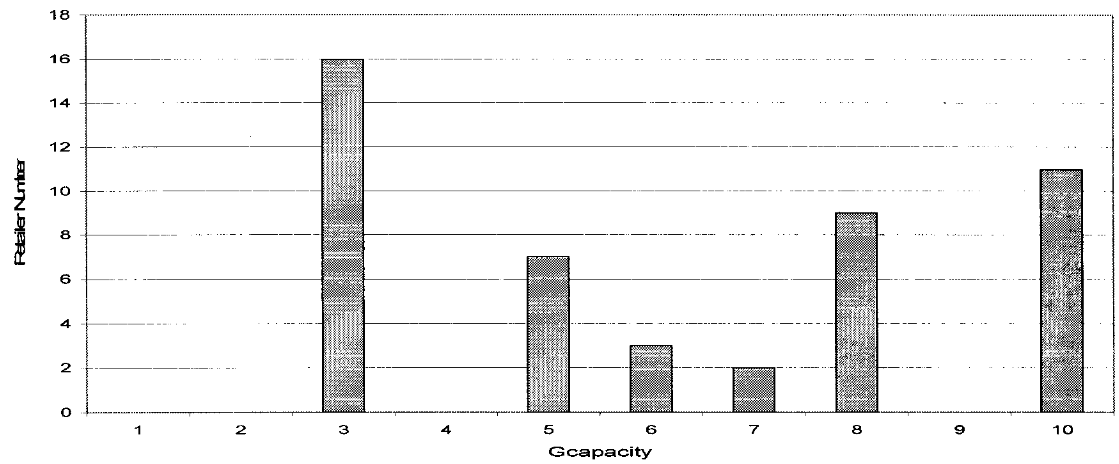

Figure 4-16 Retailer Distributions When rent = $0

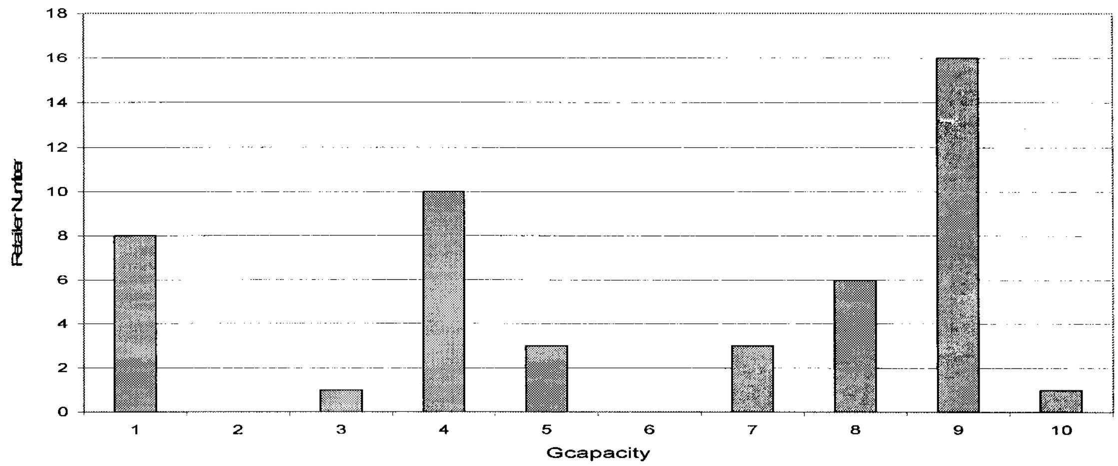

Figure 4-17 Retailer Distributions When rent = $50

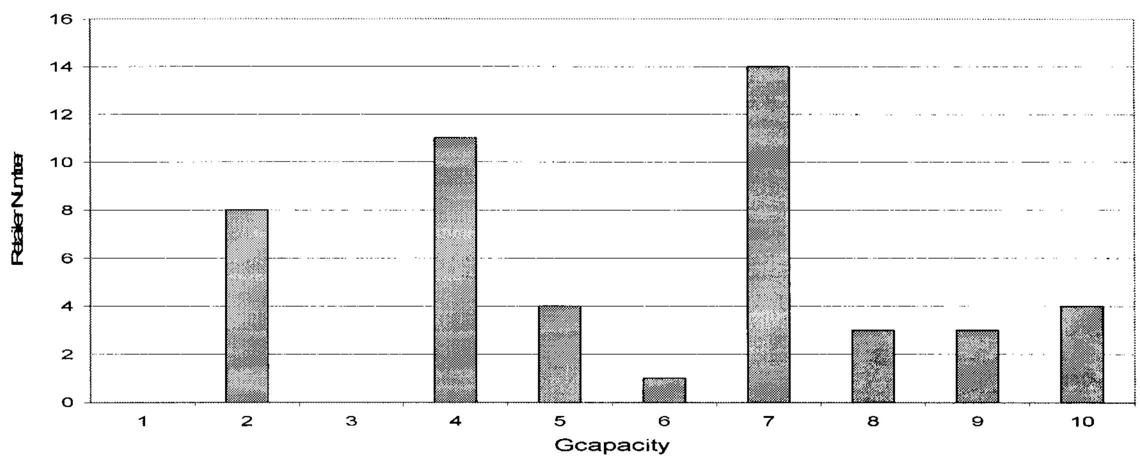

Figure 4-18 Retailer Distributions When rent =$100

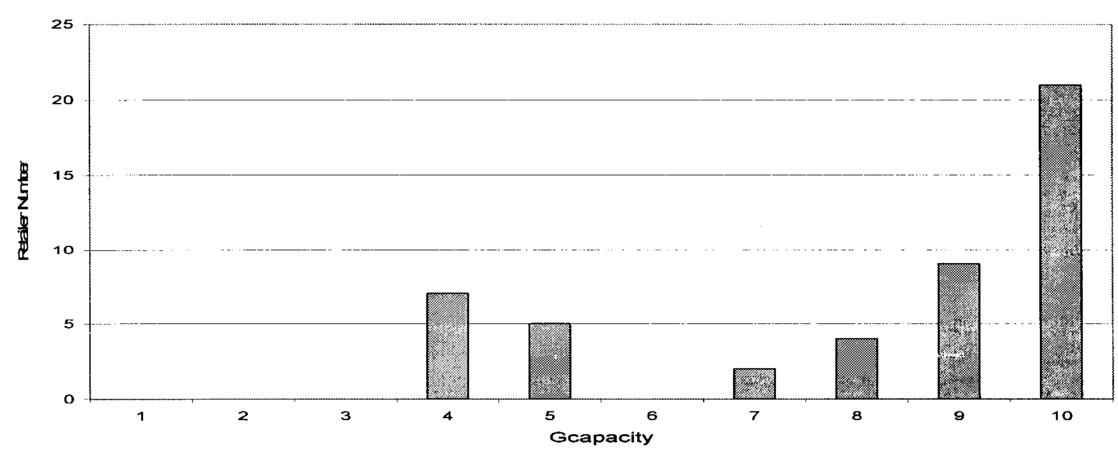

Figure 4-19 Retailer Distributions When rent = $200

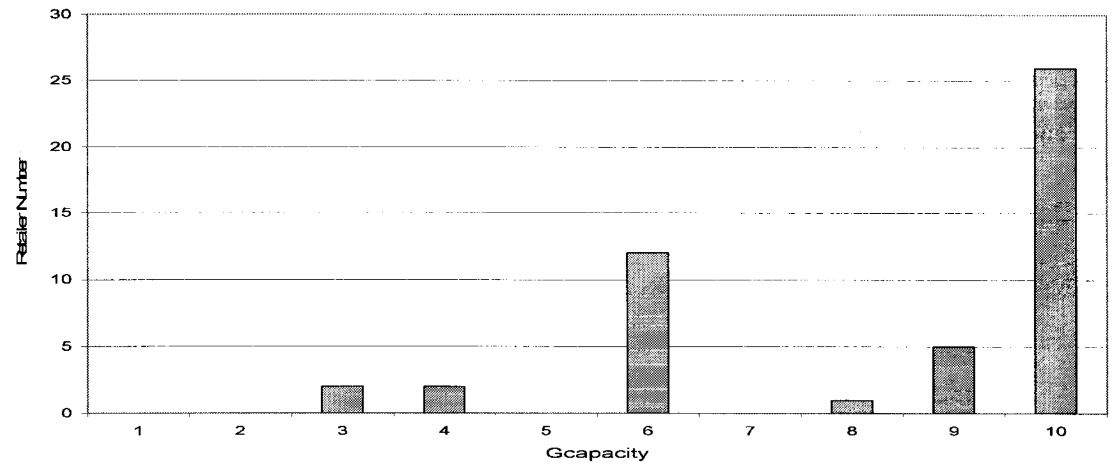

Figure 4-20 Retailer Distributions When rent = \$300

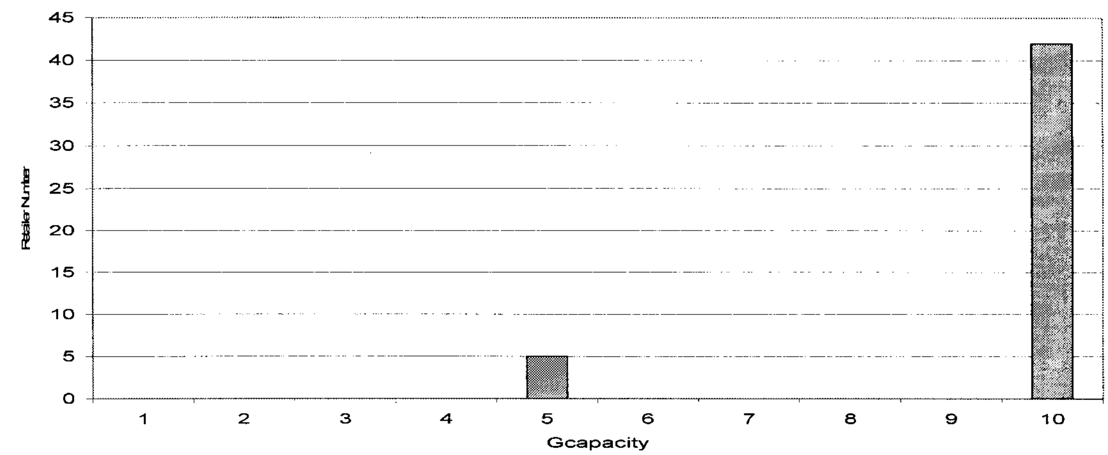

Figure 4-21 Retailer Distributions When Rent = \$400

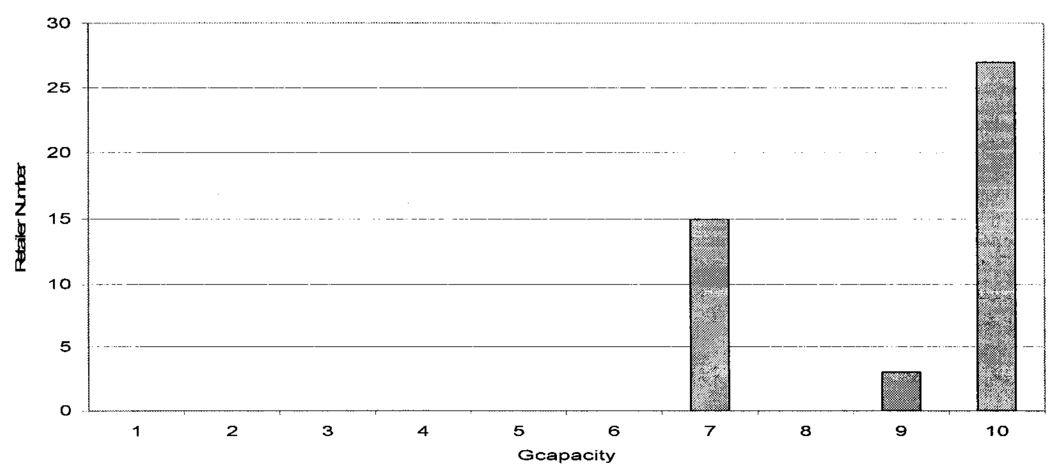

Figure 4-22 Retailer Distributions When rent = \$500

Situation 2) the rent depends on the storage capacity

Rent = unitPrice * stock_capacity

In this situation, the retailers with large storage capacities can survive because large storage capacities help them to make large deals to support the high rents; while the retailers with small storage capacities can also survive because their rents are low. From this point of view, all retailers can survive no matter how much their rents are. Hence the simulations showed that no matter what the unit price is, retailers' distributions have no pattern.

Figure 4-23 and Figure 4-24 illustrate two totally different distributions when the unit prices are the same.

4.2.3 Conclusion

The new gene, Gcapacity, increases retailers' strategies. When retailers pay the same rent despite their different storage capacities, and when the rent is roughly between 200 and 500, most retailers' new genes, Gcapacity, is convergent at high value since they need large storage capacity to allow them to make big deal to support the high rent. No patterns have been found when retailers rent depended on their storage capacities because all retailers can afford the rent.

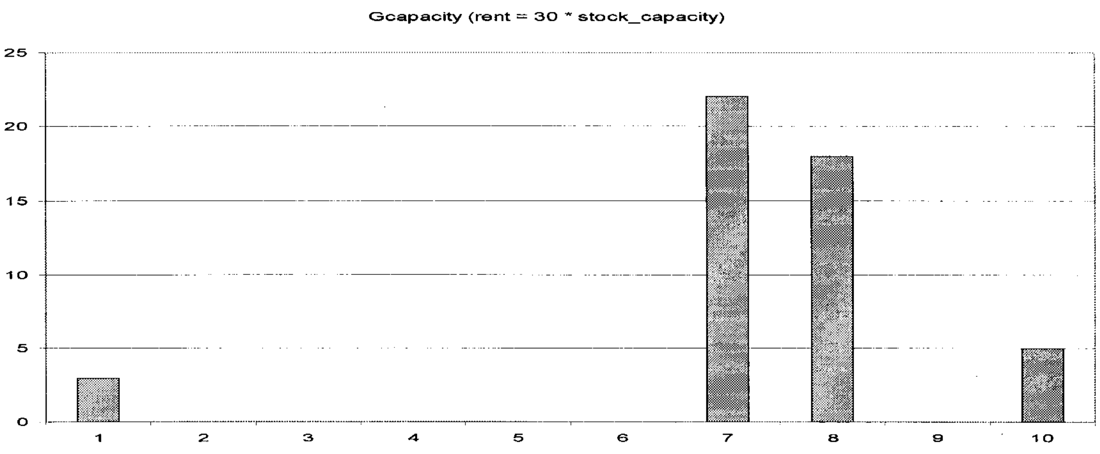

Figure 4-23 Retailer Distributions When rent = $30 * stock_capacity

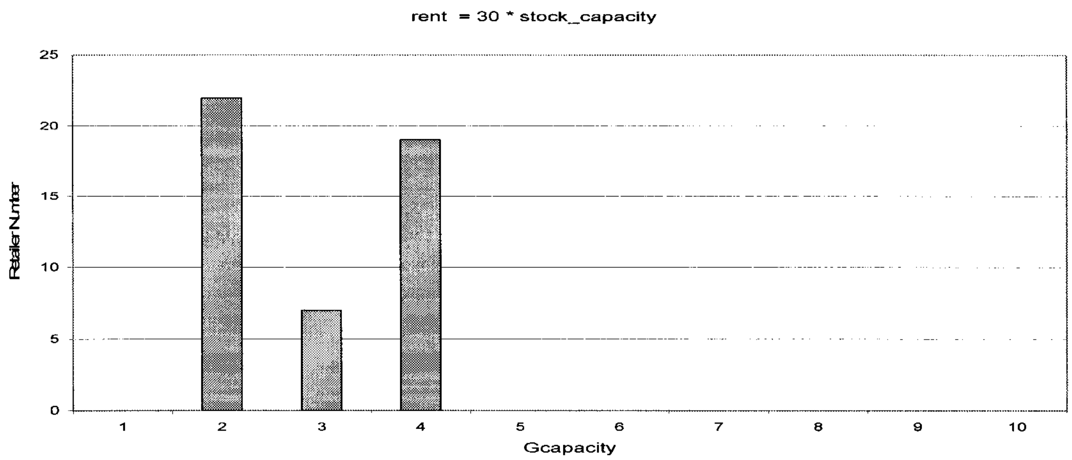

Figure 4-24 Retailer Distributions When rent = $30 * stock_capacity

4.3 Allow the Most profitable Retailers to Open New Branches

4.3.1 Goal

The third enhancement is to allow the most profitable retailers to open new branches.

With this enhancement, there are two types of retailers in the country: headquarters retailers and branch retailers. Whether a headquarters can open a new branch depends on its behavior. Only the most profitable headquarters are allowed to open branches. A profitable branch can open another branch which has the same headquarters. Every branch retailer belongs to a headquarter retailer, and the genome of a branch retailer is exactly the same as its headquarters'.

The purpose of this enhancement is to see whether those most profitable retailers can grab deals from other retailers and force them into bankruptcy.

4.3.2 Result

In the simulation, the maximum number of headquarters was limited to 30; the maximum number of retailers was decided by the number of avenues and streets, number of gas stations, producers and consumers.

$$\begin{aligned} & \text{MaximumRetailerNumber} \\ = \ & \text{AvenueNumber} * \text{StreetNumber} - \text{GasStationNumber} - \text{ProducerNumber} - \\ & \text{ConsumerNumber} \end{aligned}$$

in which MaximumRetailerNumber is the maximal possible number of retailers in the country; AvenueNumber and StreetNumber are the numbers of avenues and number of streets in the country respectively; GasStationNumber, ProducerNumber, and ConsumerNumber are the numbers of gas stations, producers and consumers in the country. In the simulation, there are 10 avenues and 10 streets; there are 8 gas stations, 4 producers, and 40 consumers. So

$$\text{MaxRetailerNumber} = 10 * 10 - 8 - 4 - 40 = 48$$

Which means, in the simulation, there could be at most 48 retailers in the country. But the maximal number of headquarters retailers is restricted to be thirty.

In the first fifteen generations, no branches were opened so that thirty headquarters could be created without disturbance. After the sixteenth generation, no new headquarters were created. At the end of each generation after the sixteenth, each retailer's profit was compared with retailers' average profit. If a retailer's profit was good enough (In the simulation, if it was over two times higher than the average), and if there was at least one empty intersection, it opened a new branch.

Figure 4-25 and 4-26 are the result of one simulation. Figure 4-25 demonstrates that the headquarters without branches can survive in the competition. Figure 4-26 demonstrates that the average profit of headquarters together with their branches is higher than the average profit of headquarters without branches. This is reasonable because the genomes of the most profitable retailers are the best which insure those headquarters retailers and their branches to succeed in competition.

From Figure 4-25, we also noticed that occasionally, some branches are removed in the simulation since there had been unprofitable for five generations. But later on, the empty intersections were filled by other new branches which survived to the end of the simulation. Why some branches couldn't survive while others could, even though they all had good genomes? The reason is unknown.

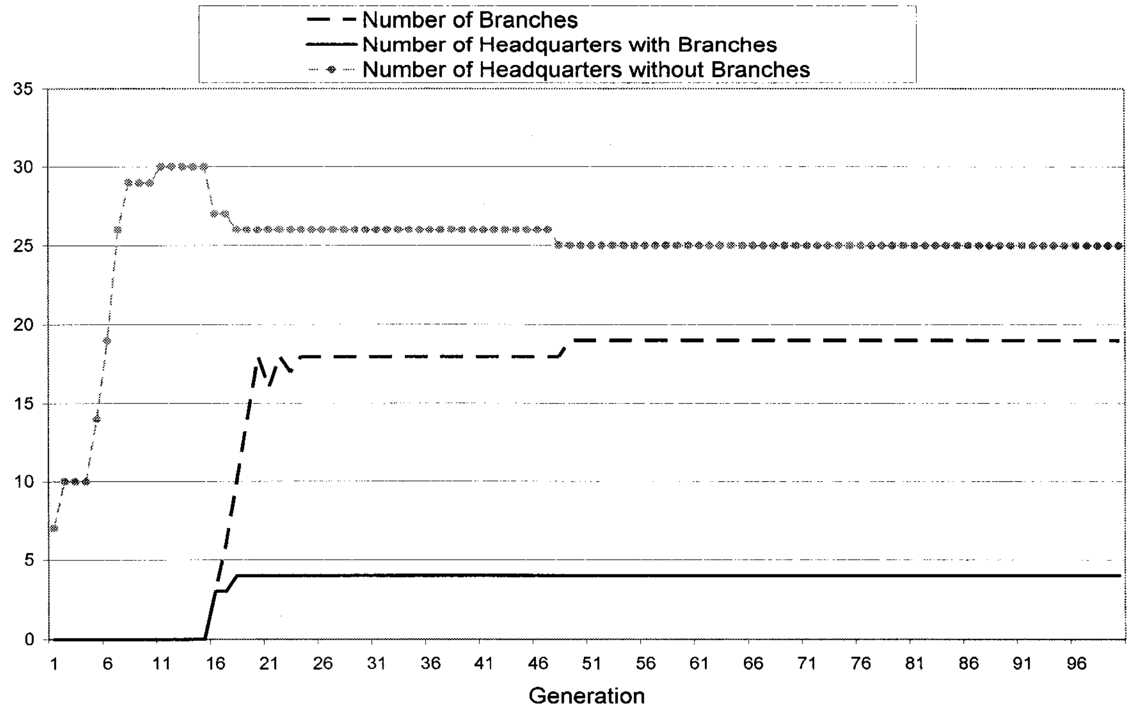

Figure 4-25 Number of Different Retailers

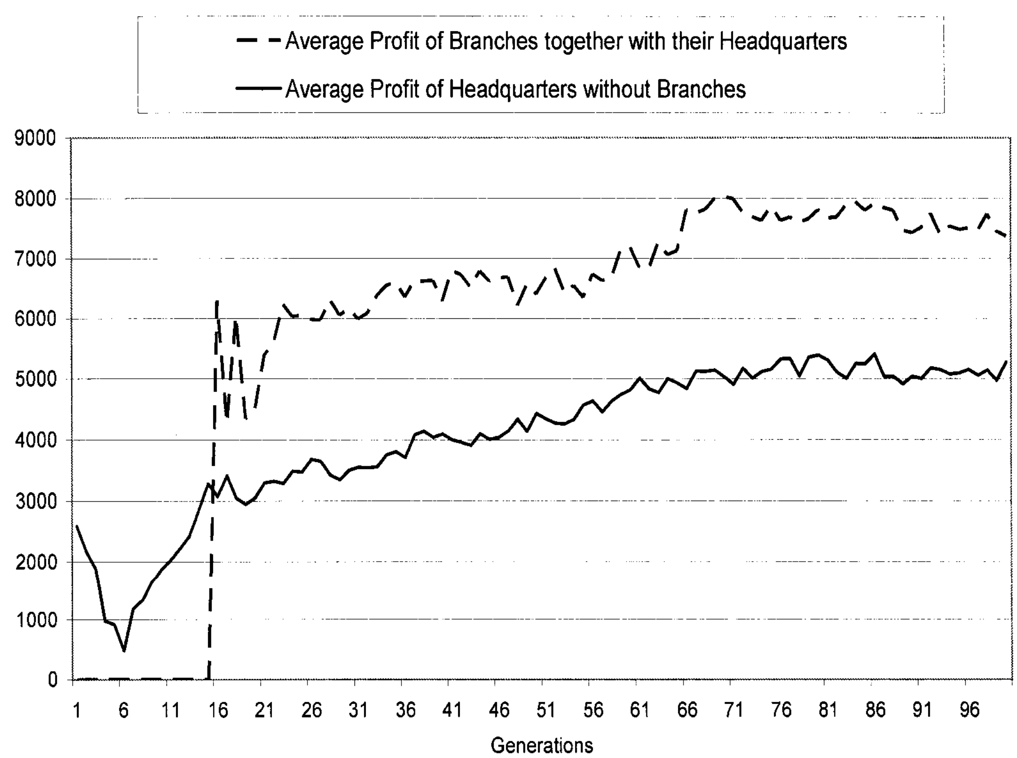

Figure 4-26 Average Profit of Different Retailers

4.3.3 Conclusion

When the most profitable retailers are allowed to open branches, other retailers that don't open branches can still survive till the end of simulation. The average profit of the most profitable headquarters along with their branches is higher than the average profit of headquarters without branches because the genomes of the former headquarters are better than those of the latter. But even though all branches had the same good genome as their headquarters had, some of them made no profits. The reason is unknown.

5 Analysis

5.1 Trucks' Profit Proportion Analysis

Trucks have two ways to gain profits: by trading crates and by trading items. So their profits may be divided into two parts based on the trade type.

Figure 5-1 shows trucks' distributions at the end of a simulation based on the average percentage of the profit gained by trading crates over the total profit.

In Figure 5-1, 60 trucks completely specialized in trading crates. 16 trucks 99-percent specialized in trading crates. 4 trucks 98-percent specialized in trading crates. Why? In Appendix 1, Table 1 to Table 4 depict some genes of remaining trucks which totally, ninety nine, and ninety eight percent specialized in trading crates and the genome of remaining headquarters retailers. By analyzing the genes of those 60 trucks and of the headquarters retailers of the last generation, we found that the maximum Gb (a gene deciding trucks' item buying price) of those 60 trucks was 180, which means the highest item buying price of those trucks was

$$1 * 180 / 255 = 0.706\$$$

However the minimum Gs (a gene deciding retailers' item selling price) of remaining retailers was 190, which means the lowest item selling price of remaining retailers was

$$1 * 190 / 255 = 0.745\$$$

So those 60 trucks had to specialize in trading crates because the item buying prices they offered were too low to buy any items from retailers.

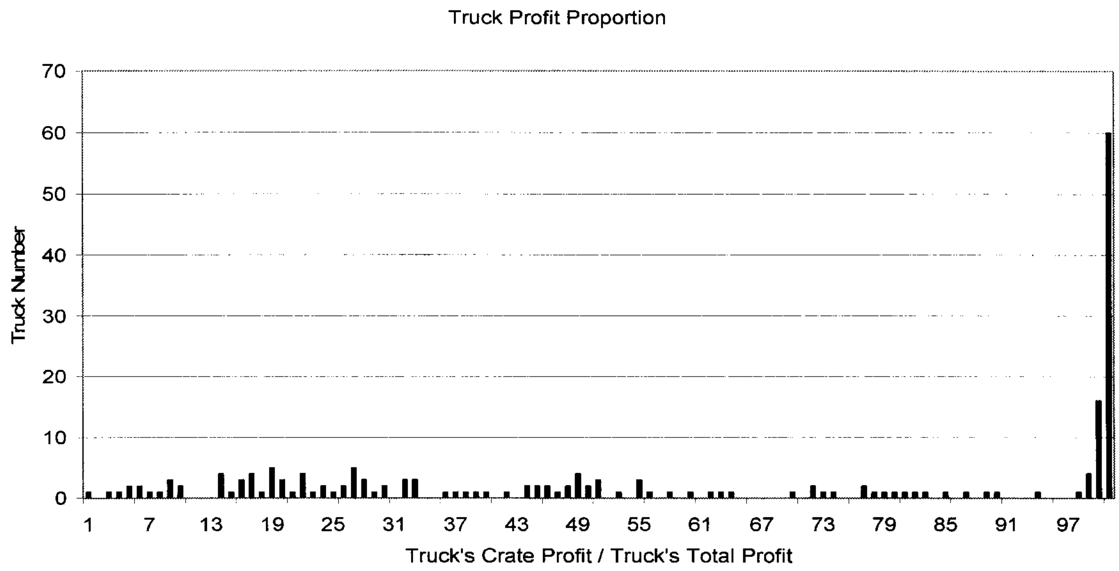

Figure 5-1 Trucks' distribution Based on the Ratio of Crate Profit / Total Profit

But why other 20 trucks, whose highest Gb was 186, didn't completely specialize in trading crates even though their highest item buying prices were also lower than remaining retailers' lowest item selling price? This is because during the simulation, some retailers offering low item selling prices were created, which allowed some trucks to buy some items from them with low price. But those retailers were removed later on. Table 5 in Appendix 1 proves this by showing that in the first generation, 35 retailers had the gene Gs lower than 186.

196 trucks survived at the end of simulation, among which 80 trucks (forty percent of the total trucks) completely or almost completely specialized in trading crates. Why so many trucks completely or almost completely specialized in trading crates? This question can be answered using probability theory. We know that the range of trucks' gene Gb is

between 0 and 255. Each truck's genome is selected randomly. Theoretically speaking, in this simulation, there might be

$$\text{Total truck number} * (\text{Retailers' lowest Gs}+1) / (255+1)$$

$$= 196 * (190 + 1) / (255 + 1)$$

$$\approx 146$$

trucks that specialized in trading crates. So in this simulation, 80 trucks completely or almost completely specializing in trading crates is acceptable.

5.2 Detailed Analysis of Trucks' Genome

There are seventeen genes in trucks' genome. They can be sorted into two types. Each gene in the first type, including Gdeal, Gscan, Gpriority, Gbest, Gphone, Gbadsell, Gbadbuy, Greserve, Gcongestion, and Gcorner, has only two or three values. For this type of genes, I mainly investigated whether some special value for each gene can make trucks more profitable than others. Each gene in the second type, including Gg, Gs, Gb, Gtransfer, Gcapacity, and Gtank, has a wide range. For this type of genes, I mainly searched for the appropriate range that can keep on the simulation and some extreme values. Gpopulation is an exception. How it affects the simulation was discussed in detail in section 4.1.

5.2.1 Gdeal - (0 ~ 2)

This gene determines how the truck looks for deals.

- 0 - The truck looks for a deal after it has completed one, but on the way to completing a deal (d1), if it discovers that it can make another deal (d2) with the dealer at its current location, it will make the deal (d2) and then continue to complete the holding deal (d1) if the holding deal is still valid (such as having room available for buying crates or items).
- 1 - The truck looks for a deal only after completing the current deal. It doesn't check other possible deals before it completes the current one.
- 2 - The truck doesn't look for deals beforehand. It only wanders around the country. When it arrives at an intersection where a dealer (producer, retailer, consumer, or gas station) is located, it checks if it can make a deal with the dealer. If so, it completes the deal; otherwise it passes it and goes to the next intersection. It will go to buy gas when it is low on gas. It doesn't memorize any information of dealers except the locations of gas stations. The truck ignores Gscan, Gpriority, Gbest, Gphone and Gbadbuy in this strategy.

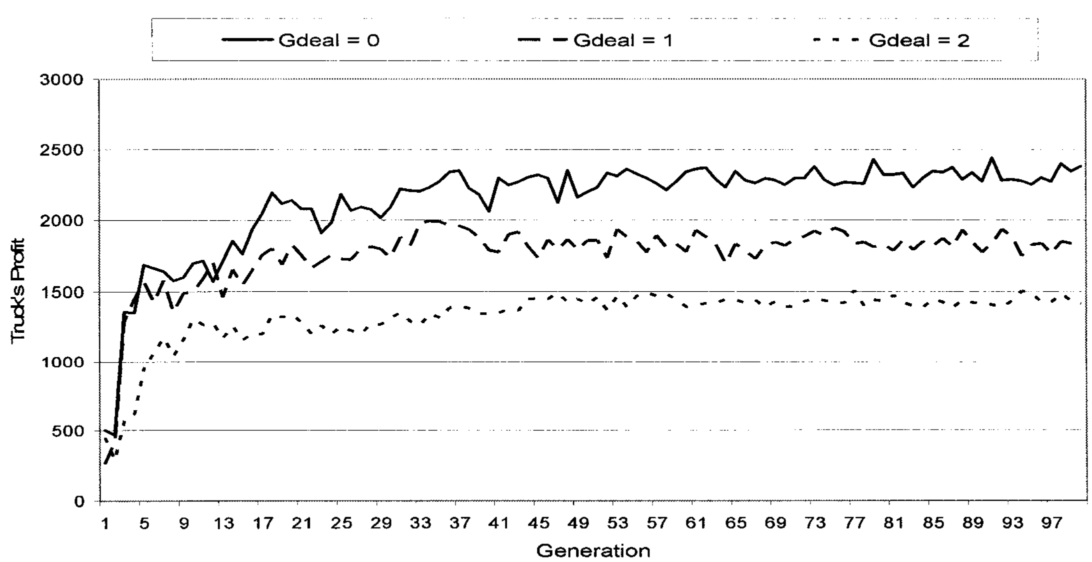

Figure 5-2 Testing Result of Gdeal

Testing Results:

The first group of trucks with Gdeal of 0 always made more profits than other two groups with Gdeal of 1 and 2 because the first group is more flexible than the second group, and the last group wasted much time and gas doing nothing but wandering.

5.2.2 Gscan - (0 ~ 1)

This gene determines whether the truck scans the country before looking for deals.

- 0 - The truck scans the whole country to mark down the locations of the dealers before looking for any deal.
- 1 - The truck doesn't scan the country.

Testing Results:

Sometimes the first group of trucks with Gscan of 0 made more profit; but some other time, it was the second group with Gscan of 1 that made more profit. So our conclusion is that the gene Gscan doesn't affect trucks' performance.

5.2.3 Gpriority - (0 ~ 2)

This gene determines the priority of deals when the truck needs to look for a deal.

- 0 - Their priorities are: buy gas when the remaining gas falls below the gas limit, sell crates if any, sell items if any, buy crates & buy items in turn.

- 1 - Their priorities are: buy gas when the remaining gas falls below the gas limit, sell crates if any, sell items if any, buy crates or buy items depending on which kind of dealer is closer.
- 2 - The truck chooses a deal following these steps: buy gas when the remaining gas falls below the gas limit or the truck is right at a gas station, otherwise check its status to decide what kinds of deal it may make (for example, if the truck's status is: has crates; has capital; has space available for at least one crate, the deals it may make are: buy crates, buy items, sell crates), then decide the deal depending on which kind of dealer is the closest (according to the available deals it may make). In this case, Gbest is useless.

Testing Results:

Experiments showed that any of the three groups of trucks might become the most profitable. We concluded that Gpriority had little or no effect on performance.

5.2.4 Gbest - (0 ~ 2)

This gene determines the criteria of the best retailer.

- 0 - The one that offers the best price.
- 1 - The one that is at the nearest location with its buying price higher than or equal to the truck's selling price when the truck's current deal is selling crates, or with its selling price lower than or equal to the truck's buying price when the truck's current deal is buying items.

- 2 - In addition to calculating the capital it will gain in selling (or capital it will cost in buying), also calculate the cost on the gas consumed to complete the deal, then the best dealer is the one that it can gain the biggest profit from (or the one that it will cost the least when buying items from).

Testing Results:

Experiments showed that any of the three groups of trucks might become the most profitable. We concluded that Gbest had little or no effect on performance.

5.2.5 Gphone - (0 ~ 2)

This gene determines how the truck uses the phone to get and verify the dealer's information.

- 0 - Don't use the phone. The truck only uses the information it got when it traveled the country.
- 1 - Use the phone to verify dealers' information. When it hasn't enough information for making a decision, use the phone to get new information of randomly chosen intersections. There is a limitation of five calls.
- 2 - This strategy is almost the same with the above except that the truck doesn't randomly choose intersections to get new information, but starts from the nearest intersections and extends to the further ones gradually. The maximum is also five calls.

Testing Results:

Experiments showed that any of the three groups of trucks might become the most profitable. We concluded that Gphone had little or no effect on performance.

5.2.6 Gbadsell - (0 ~ 1)

This gene determines if the truck permits unprofitable sells.

- 0 - Sell the crates only at a good price that is higher than or equal to its crate selling price.
- 1 - Sell the crates at a good price if possible. Also sell the crates even at a bad price that may lead to a capital loss after having continuously failed for three times in selling them.

Testing Results:

Experiments showed that any of the two groups of trucks might become the most profitable. We concluded that Gbadsell cannot help trucks to make more profits. But it can help trucks to survive (see 5.2.12).

5.2.7 Gbadbuy - (0 ~ 1)

This gene determines if the truck permits unprofitable buys.

- 0 - Buy items only at a good price that is lower than or equal to its item buying price.

- 1 - Buy items at a good price if possible. Also buy items even at a bad price that may lead to a capital loss after having continuously failed for three times in buying items.

Testing Results:

Experiments showed that any of the two groups of trucks might become the most profitable. We concluded that Gbadbuy had little or no effect on performance.

5.2.8 Greserve - (0 ~ 2)

This gene determines how much a truck reserves for buying gas, how the truck makes the decision on the amount of crates and items it is going to purchase.

- 0 - Reserve $0. Buy as many as it can according to its available capital and space.
- 1 - Reserve capital for buying a full tank of gas.
- 2 - Reserve capital for buying gas that is enough for traveling the longest distance across the country.

Testing Results:

Experiments showed that any of the three groups of trucks might become the most profitable. We concluded that Greserve had little or no effect on performance.

5.2.9 Gcongestion - (0 ~ 2)

This gene determines how the truck responses to traffic congestion.

- 0 - Stay still until the congestion disappears.
- 1 - If the congestion happens at the truck's destination, stand still until the congestion disappears; otherwise go back to the previous intersection and move along another route.
- 2 - Cancel the current destination and look for a new one.

Testing Results:

Experiments showed that any of the three groups of trucks might become the most profitable. We concluded that Gcongestion had little or no effect on performance.

5.2.10 Gcorner - (0 ~ 1)

This gene determines if the truck goes to the nearest corner after its creation.

- 0 - Do not go anywhere. It simply begins to run from its initial position.
- 1 - Go to the nearest corner at first.

Testing Results:

Experiments showed that any of the two groups of trucks might become the most profitable. We concluded that Gcorner had little or no effect on performance.

5.2.11 Gg - (0 ~ 20)

This gene determines the lowest gas limit. Suppose that the truck has N liters of gas and the capacity of its gas tank is T. Then the truck will search for gas if $N/T < Gg/20$.

Testing Results:

When all trucks' genes Gg were 0, most simulations were terminated because all trucks ran out of gas. But occasionally, the simulation could keep going. We found that in such a simulation, all survival trucks' genes Gdeal are 2. Those trucks with Gdeal of 2 bought gas whenever they passed a gas station, which helped them avoid being unmovable for lack of gas. Along with the Gene Gg's increase, more and more simulations could keep going, and the number of trucks with Gdeal of 2 was gradually in a reasonable range.

The interesting observation is that when all trucks' genes Gg were 20, no trucks with genes Gdeal of 2 survived in the simulation. By analyzing the behavior of this group of trucks, we can understand why. The first operation this group of trucks must do after they reached to an intersection was to check whether they were low on gas. So when they arrived at first intersection after created, they noticed that they were low on gas. They went to the nearest gas station right away to get it filled. And then they left the gas station and got to a neighbor intersection, where they found they were low on gas again. They had to return to the gas station to fill their tanks. They returned, filled, left, and return to the gas station again and again. This cycle continued until their capitals came to and end. If they didn't have stocks, they were removed right away. But if they had some stocks, they suspended the cycle, moved around the city trying to sell the stocks to get cash, and to resume the cycle. At last their stocks came to an end. They just wandered in the city doing nothing. After the last drop of gas was consumed, they died.

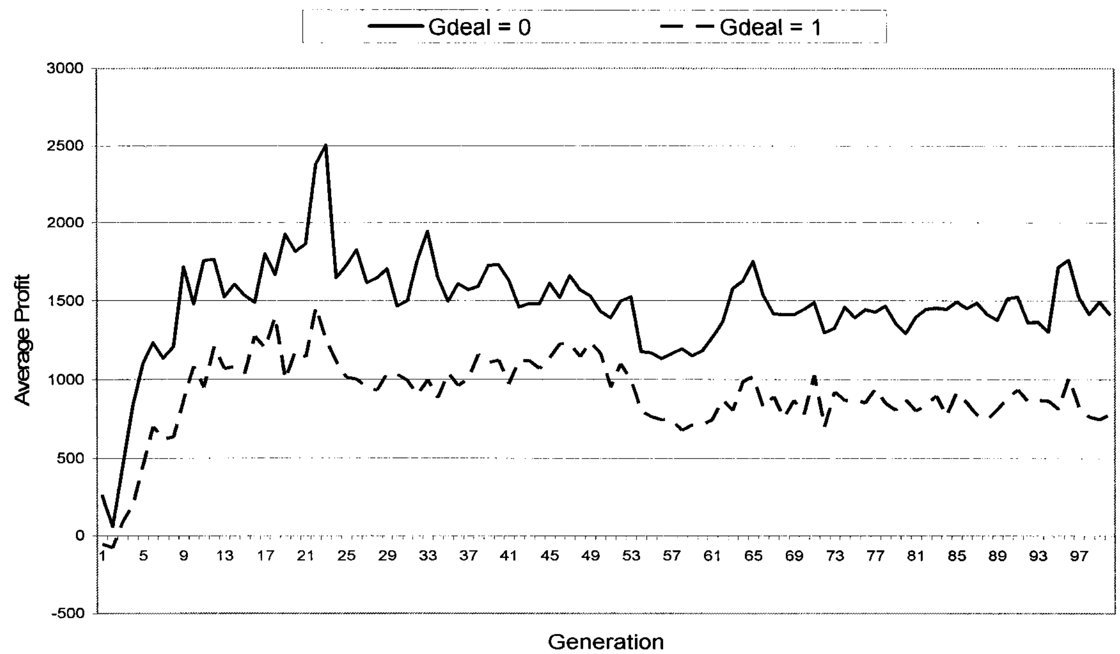

Figure 5-3 Testing Result when Gg = 20

As for the other two groups of trucks, the first group with Gdeal of 0 made more profit than the second with Gdeal of 1, as we can see in Figure 5-3.

The strategy of the second group of trucks is that they never make other deals before completion of the current deal even if it is at the intersection where a dealer offers a good deal of other kinds. So these trucks are always in a cycle: to buy gas, to make another deal. But the first group of trucks is more flexible. On their way to buy gas or to make another deal, if they find an appropriate deal at the intersection they were currently at, they make it and then keep going to finish the holding one if it was still valid. This flexibility makes the first group of truck to be more profitable than the second group. And it also makes the first group to have much more survivals than the second group, as we can see in Figure 5-4.

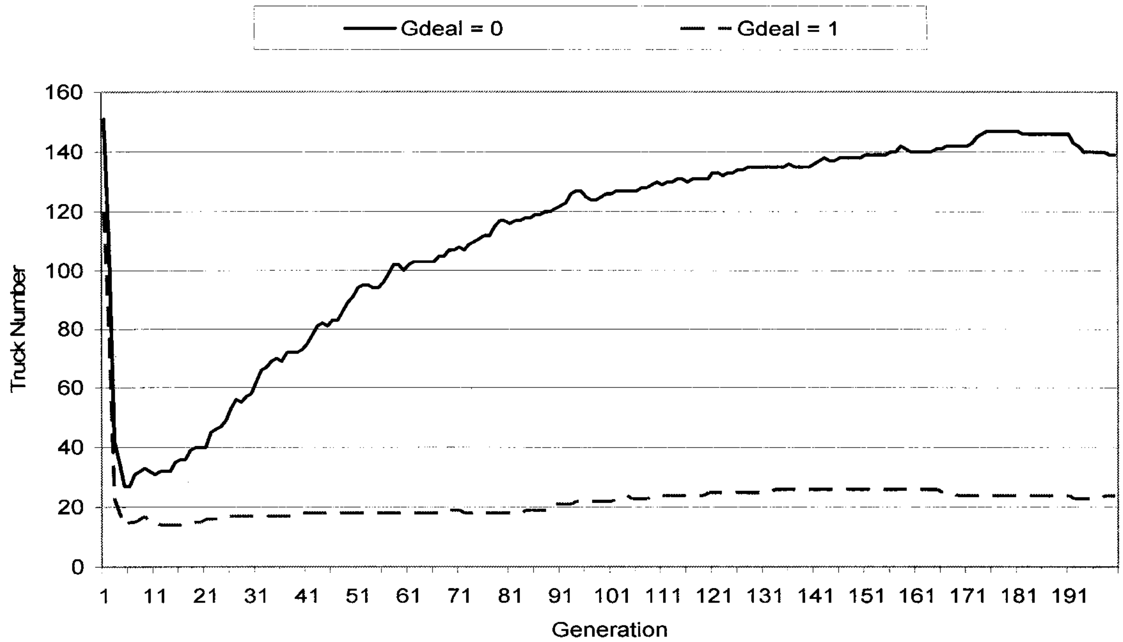

Figure 5-4 Testing Result when Gg = 20

5.2.12 Gs - (0 ~ 255)

This gene determines the price of selling one crate to retailers, Ts. Ts is calculated as the following (as described in the Trucking specification [1]):

$$T_s = N_i C_i \left(1 - \pi + \pi \frac{Gs}{255} \right)$$

(Ni: number of items in a crate; Ci: the price that a consumer pays for one item; π: profitability factor, π ≈ 0.4)

Testing Results:

When the genes Gs of all trucks were very high (higher than 200), most simulations were terminated. But some simulations could continue. By studying these simulations, I found that the genes Gbadsell of all remaining trucks were 1, which means that those trucks that would like to sell crates with a bad price might survived.

5.2.13 Gb - (0 ~ 255)

This gene determines the price of buying one item from retailers, Tb. Tb is calculated as the following (as described in the Trucking specification [1]):

$$T_b = C_i \frac{Gb}{255}$$

(Ci: the price that a consumer pays for one item)

<u>*Testing Results:*</u>

When the genes Gb of all trucks were low, all trucks had to specialize in trading crates because they couldn't buy any items from retailers. The retailer couldn't sell even one item as well. After retailers' stocks were full of crates, they couldn't make deals anymore. The simulation had to be terminated. When Gb was less than 180, a few simulations could continue. When Gb was greater than 190, most simulations could continue.

5.2.14 Gtransfer - (0 ~ 100)

This gene determines the proportion of capital that will be transferred to the child truck, as described in the Trucking specification [1]. We have:

$$\alpha = \frac{Gtransfer}{100}$$

The capital which a child truck gets from its parent trucks is divided into two parts in this way: If it's greater than or equal to the price of the child's full tank gas, give the child truck a full tank gas, and the remaining capital is left as its initial capital. Otherwise, divide the transferred capital in half. One half is for the gas; another one half is the initial capital.

Testing Results:

If the genes Gtransfer of all trucks were 0, no new trucks could survive because they had no gas. They couldn't move one step. But if the genes Gtransfer of all trucks were 100, which means parent trucks gave their child truck all capitals they had, things were different. Not all parent trucks died. Those trucks without stocks had to die. But those trucks with stocks might survive. Since many profitable trucks died, the truck number changed in an unpredictably way even if the simulation could go on.

5.2.15 Gcapacity - (1 ~ 10 crates)

This gene determines each truck's carrying capacity.

Testing Results:

The simulation could go on no matter what Gcapacity was. But obviously, the trucks with higher Gcapacity made more profits than those with lower Gcapacity. And the reason is obvious too.

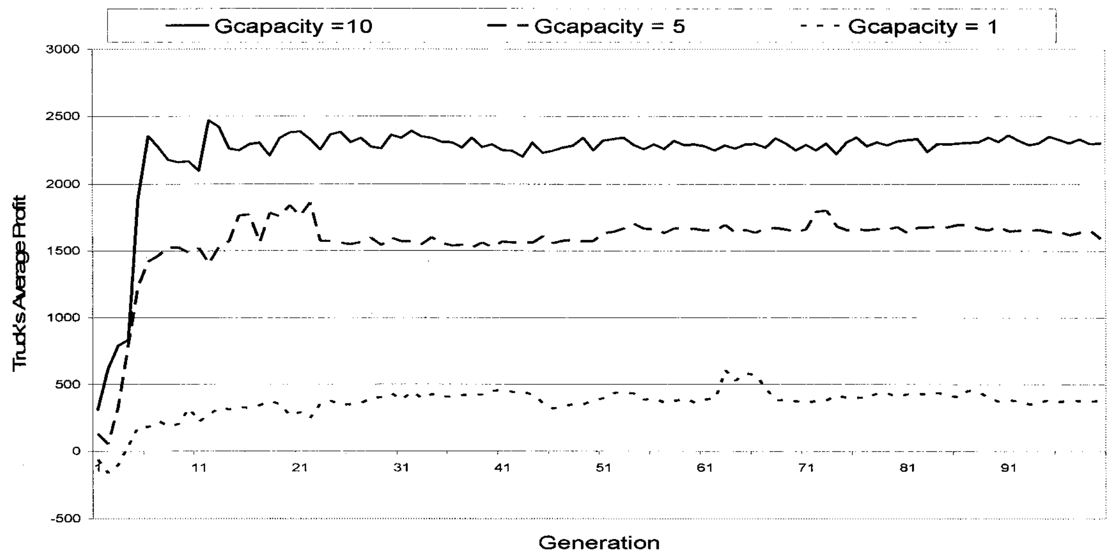

Figure 5-5 Testing Result of Gcapacity

5.2.16 Gtank - (1 ~ 60 liters)

This gene determines each truck's gas tank capacities.

Testing Results:

When Gtank was less than or equal to 5, all trucks were dead in the first several generations. The simulation had to be terminated. When Gtank was between 6 and 10, most simulations were terminated. But occasionally, a simulation could keep on since a few trucks and retailers were alive. When Gtank rose, more and more simulations could keep on, and all data gradually went to normal. Figure 5-6 shows that most trucks' Gtank had to be greater than 15 in order to survive.

Now let's take one simulation as an example to investigate why a few trucks with Gtank of very small value could survive. In this simulation, the genes Gtank of all trucks were 6, but only one truck with ID of t285 survived. And also, only one retailer with ID of r8 survived. Table 5-1 and Table 5-2 are their genomes:

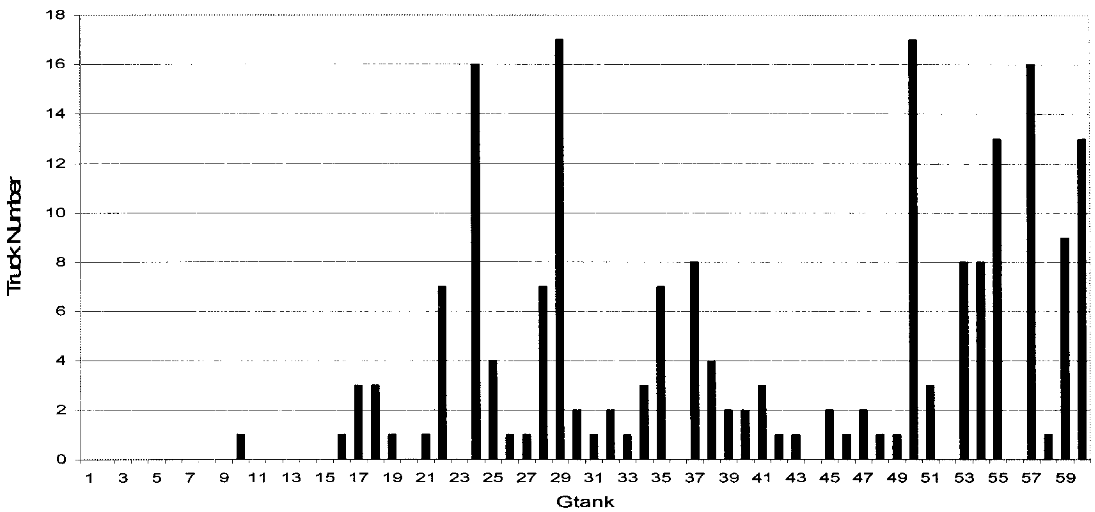

Figure 5-6 Trucks' Distributions According to Gtank

Table 5-1 Genes of Remaining Truck t285

Gg	Gs	Gb	Gtransfer	Gcapacity	Gtank	Gdeal	Gscan
18	26	242	0	8	6	0	1
Gpriority	Gbest	Gphone	Gbadsell	Gbadbuy	Greserve	Gcongestion	Gcorner
2	1	0	0	0	0	0	1

Table 5-2 Genes of Remaining Retailer

Gb	Gs	Gtransfer
44	107	33

According to Table 5-1, the truck's crate selling price was:

$$T_s = N_i C_i \left(1 - \pi + \pi \frac{Gs}{255} \right)$$

$$= 100 * 1.0 (1 – 0.4 + 0.4 * 26 / 255)$$

$$= 64.1$$

The truck's item buying price was:

$$T_b = C_i \frac{Gb}{255}$$

$$= 1.0 * 242 / 255$$

$$= 0.95$$

According to Table 5-2, the retailer's crate buying price was:

$$R_b = N_i C_i \left(1 - \pi + \pi \frac{G_b}{255}\right)$$

$$= 100 * 1.0 (1 – 0.4 + 0.4 * 44 / 255)$$

$$= 66.9$$

The retailer's item selling price was:

$$R_s = C_i \frac{G_s}{255}$$

$= 1.0 * 107/255$

$= 0.42$

The truck's crate selling price was higher than the retailer's crate buying price. The truck's item buying price was lower than the retailer's item selling price. So this truck could sell crates to retailer r8, and buy items from the same retailer.

Gas consumption rate is 0.1 liters/km. The distance between two adjacent intersections is 10 km. This truck's tank was only 6 liters which means it could travel among at most seven intersections. Table 5-3 is the country map of this simulation. The initial position of truck t285 was at (6, 0). The retailer r8 was at (8, 0), around which there were one producer, one consumer, and two gas stations. This made that truck's survival possible. It could finish the whole business cycle in a very small area: to buy and sell crates, to buy and sell items, to buy gas. Figure 5-7 and Figure 5-8 also show this. Starting from the fifth generation, there was only one truck left, t285. After one generation, all retailers except retailer r8 were dead. Since the fifth generation, the truck's average profit and retailer's average profit fell into a cycle (Even though we don't know exactly how this activity cycle was composed of.), which implied that the truck always went to the same producer, same consumer, and same gas station to make deals. Say nothing of the same retailer.

Table 5-3 Country Map of the Simulation

	0	1	2	3	4	5	6	7	8	9
0			C			C			C	G
1		P		C			C		P	
2	C		C		C			C		C
3		C		C	G	C	C		C	
4			C		C	C		C		C
5	C		C	C		C	C			
6	T	C		C	C		C		C	
7	C		C	G	C			C	G	C
8	R	P		C		G	C		P	
9	G	G	C		C		G	C		

P – Producer, C – Consumer, G – Gas station, R – Retailer, T – Truck

5.2.17 Gpopulation – (1 ~ max value)

Decide the local population limit (LPL) of each truck.

LPL = Gpopulation

During the simulation, each truck checks its local population every ten time units. At the end of one generation, the average local population of each truck is calculated. When a truck is selected to reproduce new truck, it compares the recorded average local population with its LPL. If it's greater than LPL, it refuses to reproduce. Otherwise, it does.

Testing Results:

See section 4.1.3.

5.3 Analysis of Retailers' Genome

A retailer's genome has four genes: Gb, Gs, Gtransfer and Gcapacity. I mainly studied the first three since the last gene Gcapacity had been studied in section 4.2.

5.3.1 Gb – (0~255)

This gene determines each retailer's crate buying price. Let R_b be the price at which the retailer buys crates from trucks. R_b is calculated as the following:

$$R_b = N_i C_i \left(1 - \pi + \pi \frac{G_b}{255}\right)$$

(Ni: number of items in a crate; Ci: the price that a consumer pays for one item; π: profitability factor, $\pi \approx 0.4$)

<u>*Testing Results:*</u>

When Gb of all retailers were very high (higher than 220), all simulations were terminated. Along with the decrease of Gb, more and more simulations could continue. In the successful simulations, there were some remaining trucks with Gs higher than the Gb of retailers. But all their genes Gbadsell were 1.

5.3.2 Gs – (0~255)

This gene determines the item selling price. Let R_s be the retailers' item selling price, then R_s is calculated as the following:

$$R_s = C_i \frac{G_s}{255}$$

Testing results:

When Gs of all retailers were low (normally lower than 140), all simulations were terminated. Along with the increase of Gs, more and more simulations could continue.

5.3.3 Gα – (0~255)

This gene determines the proportion of capital that will be transferred to the child retailer. Let α be the proportion of capital that will be transferred to the child retailer during reproduction. We define

$$\alpha = \frac{G_\alpha}{255}$$

Testing results:

When the genes G_α of all retailers were 0, no new retailers could survive because they had no capitals at all. When the genes G_α of all retailers were 255, whether the parents retailer could survive mostly depended on whether they had stocks.

5.4 Conclusions

Many trucks completely or almost completely specialized in trading crates because their item buying prices were lower than retailers' lowest item selling prices.

Many of the genes in a truck's genome do not seem to have much effect on the trucks' performance. This suggests running the simulation without these genes.

6. Conclusions

In this thesis, I have made two contributions to the Truckin' project: enhancement and analysis.

Firstly, three aspects of enhancement were done: adding a new gene Gpopulation to trucks, adding a new gene Gcapacity to retailers, and allowing the most profitable retailers to open new branches.

- The number of trucks could be controlled by adding a new gene Gpopulation to trucks. The result was satisfactory. But in order to avoid traffic congestion's influence to the simulation, the range of Gpopulation should be selected carefully.
- The new gene Gcapacity did increase the retailers' strategy. It decided retailers' stock capacities. When the rent was fixed to a large value no matter how large the stock capacity was, the genes Gcapacity of most retailers were high in that they needed big space to make big deal to support the rent.
- After the most profitable retailers were allowed to open new branches, other retailers without branches could still survive. But their average profit was less than the average profit of those headquarters with branches because of the difference between their genomes.

And then, three kinds of analysis were done.

- We analyzed the source of trucks' profit. We found a large part of trucks specialized in trading crates because their item buying price was lower than all retailers' item selling price. Probability theory helped us answer another question: why many trucks specializing in trading crates is because the remaining retailers' lowest item selling price is high.

- We analyzed trucks' other 16 genes. Some interesting results were found. But some genes seemed not to affect trucks' performance. Such genes include Gscan, Gpriority, Gbest, Gphone, Gbadsell, Gbadbuy, Greserve, Gcongestion, and Gcorner.
- Retailers' genes were analyzed too.

Since in a truck's genome, many genes seem not to affect trucks' performance, one direction for future work would be to remove those genes and see how trucks perform without them.

A further study may be done to investigate what prevents good genomes from successful performance all the time. Like what we encountered in section 4.3.2, a few branches made negative profits even if they had the same genomes as their successful headquarters. Whether there are some causes behind?

References

[1] P. Grogono, "Trucking", Unpublished Technical Report, Concordia University, Montreal, 2001.

[2] John H. Holland, "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975.

[3] Alden H. Wright, "Genetic Algorighms for Real Parameter Optimization", *Foundations of Genetic Algorithms*, Morgan Kaufmann Publishers, San Mateo, California, P205-218, 1991.

[4] Jeffrey Edelstein, "The Genetic Algorithm Way", Masters of Computer Science, Concordia University, April 2001.

[5] Debbie Papoulis, "The Truckin' Project Experimenting with Genetic Algorithms", Masters of Computer Science, Concordia University, October 2001.

[6] Qixia Deng, "Trucking Simulation Using Genetic Algorithms", Masters of Computer Science, Concordia University, April 2003.

[7] Melanie Mitchell, "An Introduction to Genetic Algorithms", the MIT Press, 1996.

[8] Rennard.org, "Introduction to Genetic Algorithms", http://www.rennard.org/alife/english/gavintrgb.html (currently Sep 06, 2003).

[9] Mehul M. L. Khimasia, "Overview of Genetic Algorithms", http://www.tcm.phy.cam.ac.uk/~mmlk2/report13/node6.html (currently Sep 03, 2003).

[10] Liang Yu, "Truckin' Simulation and Visual Interface", Masters of Computer Science, Concordia University, March 2001.

[11] Rong Yang, "Solving Large Travelling Salesman Problems with Small Populations",

http://www.google.ca/search?q=cache:T2oGgn2ArKgJ:www.cs.bris.ac.uk/~rong/gzipped/tsp.ps.gz+%22genetic+algorithms%22+and+%22traveling+sales+man%22+&hl=en&ie=UTF-8 (currently March 23, 2004).

[12] Dan Bonachea, Eugene Ingerman, Joshua Levy, and Scott McPeak, "An Improved Adaptive Multi-Start Approach to Finding Near-Optimal Solutions to the Euclidean TSP", http://www.google.ca/search?q=cache:qACI1aPFB9YJ:www.cs.berkeley.edu/~bonachea/tsp/AMSGECCO00.ps+%22genetic+algorithms%22+and+%22traveling+sales+man%22+&hl=en&ie=UTF-8 (currently March 23, 2004).

[13] Noureddine Bouhmala, "A Multilevel Genetic Algorithm for the Traveling Salesman Problem", http://www.google.ca/search?q=cache:SduncLPDbFsJ:www.hive.no/www-sa/fag/inf/alg85/gentsp.ps+%22genetic+algorithms%22+and+%22traveling+sales+man%22+&hl=en&ie=UTF-8 (currently March 23, 2004).

[14] Anonymous, "Genetic Algorithms" http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/tcw2/report.html#Resource (currently March 23, 2004).

[15] Tapabrata Ray, K. M. Liew, "Society and Civilization: An Optimization Algorithm Based on the Simulation of Social Behavior", *IEEE Transactions on Evolutionary Computation*, Vol. 7, No. 4, pp386-396, August 2003.

Appendix A: Tables Obtaimed From a Simulation

Table 1 Some Genes of Trucks Which Completely Specialized in Trading Crate

Gg	Gs	Gb	Gtransfer	Gcapacity	Gtank	Gdeal	Gscan	Gpriority	Gbest	Gphone
17	91	74	15	9	44	2	0	1	0	1
15	185	32	40	6	42	1	1	1	1	0
15	130	62	85	10	35	2	0	1	2	0
16	115	141	88	4	39	0	0	1	1	2
15	151	41	14	1	25	0	0	2	0	0
10	199	87	78	7	42	1	1	1	1	1
9	33	180	33	9	42	0	1	2	1	0
18	41	32	40	7	59	2	1	1	1	0
15	33	32	33	6	42	0	1	1	1	0
14	194	151	37	9	56	0	1	0	1	1
3	66	133	21	5	49	2	1	2	0	1
18	101	32	26	8	49	1	1	1	1	0
13	162	14	87	9	30	2	0	0	2	1
16	115	141	88	3	42	0	0	1	1	1
7	123	141	60	4	39	0	1	1	1	1
8	101	163	42	8	48	1	1	1	1	2
9	162	71	37	9	57	2	1	0	2	2
18	149	71	14	10	45	2	0	2	2	2
13	123	141	85	10	60	0	1	1	1	1
3	32	122	26	10	42	2	1	2	0	2
10	106	1	66	6	26	2	0	1	2	2
16	208	71	88	3	57	2	0	2	2	1
18	38	151	87	9	56	2	1	0	1	2
13	127	62	60	10	60	0	1	1	1	0
11	47	163	91	8	48	2	1	1	1	1

18	37	32	87	7	30	1	1	1	1	0
20	101	163	42	4	43	1	1	0	1	2
17	66	133	21	8	49	1	1	2	1	2
8	95	133	76	7	42	1	1	1	1	0
15	115	141	66	6	42	2	1	1	1	1
15	32	122	26	6	50	2	1	2	1	2
3	66	133	12	8	42	2	1	1	0	2
13	65	14	91	8	41	2	1	0	2	0
17	91	133	87	8	49	2	1	1	1	2
17	90	12	85	4	41	2	0	0	2	0
13	26	14	91	9	60	2	1	0	2	0
9	101	163	60	8	37	2	1	0	1	1
12	66	133	12	8	59	2	1	0	0	2
12	47	119	85	10	49	0	1	2	2	0
17	91	74	51	9	38	2	0	2	1	1
17	80	122	76	6	37	2	1	0	0	2
17	37	74	15	9	44	2	1	0	2	2
14	66	133	76	5	42	2	1	2	0	1
16	47	141	88	3	49	0	1	1	0	0
12	6	133	88	8	59	2	1	2	1	2
17	37	141	87	3	37	1	1	2	0	0
12	91	74	15	9	52	2	1	1	0	2
12	115	141	88	8	42	2	1	1	1	2
3	47	133	88	8	49	2	1	1	0	0
12	6	141	88	8	59	2	1	2	1	2
12	6	122	26	8	59	2	1	2	0	2
17	47	133	78	8	49	1	1	2	0	2
3	32	74	21	5	49	2	1	2	0	2
17	47	122	78	8	59	2	1	2	1	0
17	37	74	78	8	44	2	1	1	2	2

17	66	74	78	8	44	2	0	2	0	1
17	6	122	76	6	37	2	0	0	1	2
17	37	141	87	3	37	1	1	2	0	2
12	115	141	88	8	42	2	1	1	1	2
17	47	122	21	10	39	2	0	1	0	0

Table 2 Some Genes of Trucks Which Ninety Nine Percent Specialized in Trading Crate

Gg	Gs	Gb	Gtransfer	Gcapacity	Gtank	Gdeal	Gscan	Gpriority	Gbest	Gphone
6	51	86	34	5	52	2	1	1	1	2
11	66	133	58	5	29	2	1	0	0	1
10	154	130	49	4	39	2	1	1	1	0
12	90	12	30	4	41	2	1	2	0	0
0	48	108	3	7	58	2	1	1	0	1
9	82	122	87	8	44	2	0	1	2	1
16	98	138	51	9	59	1	1	2	2	1
17	184	177	50	6	56	2	1	0	2	0
3	47	122	21	10	49	2	1	2	0	0
17	101	163	26	8	48	1	1	1	1	2
9	123	151	60	9	56	0	1	1	1	1
13	162	71	14	9	57	2	0	0	2	2
16	100	73	27	10	54	2	1	0	1	1
7	124	144	2	6	32	2	0	1	2	1
5	136	162	10	6	41	2	0	0	1	0
6	7	99	33	9	59	0	1	1	1	2

Table 3 Some Genes of Trucks Which Ninety Eight Percent Specialized in Trading Crate

Gg	Gs	Gb	Gtransfer	Gcapacity	Gtank	Gdeal	Gscan	Gpriority	Gbest	Gphone
9	33	180	93	3	27	0	1	2	1	1
17	45	170	84	8	56	2	0	2	0	0
11	133	186	40	3	42	0	1	2	1	1
15	45	185	19	7	31	1	1	2	2	2

Table 4 Genomes of Headquarter Retailers

Gb	Gs	Gtransfer	Gcapacity	Gb	Gs	Gtransfer	Gcapacity
151	194	158	5	140	210	144	8
53	191	132	4	96	232	10	3
100	205	50	8	140	210	4	8
140	226	183	8	100	227	163	8
210	232	10	3	140	226	183	5
2	229	195	10	53	191	132	8
210	232	138	5	135	234	138	5
48	222	176	9	210	232	2	5
96	205	34	2	208	232	10	3
5	249	4	8	125	191	10	4
66	229	69	10	53	190	68	10

Table 5 some Genes of Retailers in the First Generation

Gb	Gs	Gtransfer		Gb	Gs	Gtransfer
64	161	40		2	229	195
116	51	4		29	16	65
83	95	247		7	42	90
88	151	122		212	174	188
106	68	34		246	251	203
119	2	144		145	26	56
0	174	38		69	83	117
151	194	158		213	127	143
53	191	132		247	221	97
59	188	185		140	226	183
235	109	50		172	126	37
214	23	76		117	52	130
36	3	239		221	206	40
136	22	164		201	201	56
244	17	10		210	232	10
220	170	157		150	89	42
119	70	210		103	72	86
5	166	251		48	10	177
175	150	123		6	112	144
10	45	155		163	156	46
112	222	240		217	176	205
100	205	50		97	23	39
173	37	117		238	196	177
155	137	27		211	99	32