TRUCKING SIMULATION

USING GENETIC ALGORITHMS

QIXIA DENG

A THESIS

IN

THE DEPARTMENT

OF

COMPUTER SCIENCE

PRESENTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE AT
CONCORDIA UNIVERSITY
MONTREAL, QUEBEC, CANADA

MARCH 2003

0-612-77709-X

ABSTRACT

Trucking Simulation Using Genetic Algorithms

Qixia Deng

Genetic Algorithms (GAs) are stochastic search and optimization methods inspired by the mechanisms of natural adaptation. In the last two decades they have been researched and applied in a variety of areas. Currently GAs are used extensively in solving complex optimization problems with large but finite search space. This thesis studies two genetic algorithms applied to a trucking simulation problem where trucks travel among dealers in a country and transport commodities from producers to retailers and from retailers to consumers. Both trucks and retailers attempt to survive and make the most individual profits. Trucks and retailers evolve simultaneously in the simulation. Their evolution progress in two economy types is examined. The results show different effectiveness of these two algorithms in the two economy types.

ACKNOWLEDGEMENTS

I would like to use this opportunity to give grateful thanks to my supervisor, Dr. Peter Grogono, for his support and guidance throughout the project.

I would also like to express thanks to my husband, Weining Zhong, for his support and patience.

Table of Contents

LIST OF FIGURES

1 INTRODUCTION

Trucking is a platform for researching the application of evolutionary computation methods. It simulates a small country where trucks travel among dealers and transport commodities from producers to retailers and from retailers to consumers. Both trucks and retailers attempt to survive and make the most individual profit. Trucking originated from a game invented by Mark Stefik and others at Xerox Palo Alto Research Center (PARC) [16] during the eighties as part of their research into expert systems.

Genetic Algorithms (GAs) are a form of optimization and search technique. They solve problems by simulating the natural phenomenon of adaptation by evolution. Genetic algorithms have been widely used to solve complex optimization problems with a large but finite search space.

In this thesis, we use Genetic Algorithms to evolve trucks and retailers so as to find the strategies that enable them to achieve the best performances. There is an earlier version of Trucking done by Jeff Edelstein [4] and Debbie Papoulis [14] that also used the genetic algorithm to solve this problem. However, an insufficiency of their simulation was lack of the evolution of retailers. In addition, the search space of trucks was quite small. The purpose of this thesis is to make improvements to this topic by designing more strategies of trucks to increase their search space and by adding the evolution of retailers. Besides, we also added two forms of limitation to resources to increase the competition of trucks and retailers. On the other hand, we also want to find out if the genetic algorithm is a good solution to the Trucking problem.

In the next section, an overview of optimization and genetic algorithms is described. The details of Trucking are introduced in Section 3. Section 4 explains the two genetic algorithms used in Trucking. Some simulation parameters are described in Section 5. The simulation results and a discussion are presented in Section 6. The last section draws a conclusion.

2 AN OVERVIEW OF OPTIMIZATION AND GENETIC ALGORITHMS

"Optimization is the process of adjusting the inputs to or characteristics of a device, mathematical process, or experiment to find the minimum or maximum output or result." [9] Optimization problems exist everywhere. For example, what is the fastest route from home to the school? In what order should we arrange the things so that we can finish them in the shortest time period? Optimization is also an important task in many fields of scientific research and engineering. Many methods have already been developed to solve optimization problems in different areas.

2.1 Categories of Optimization

R. L. Haupt and S. E. Haupt break optimization algorithms into six categories. [9] We give a brief introduction to these categories below.

1. Trial-and-error and mathematical function optimization

   In trial-and-error optimization, the process that produces the output is like a black box. Different input parameters are fed to one side of the box, and the results are seen on another side. How the results are changed is invisible. In mathematical function optimization, the process is like a white box. The actions inside the box, which are assumed to be represented as a mathematical formula, are known and can be controlled. Filling the box with different formula produces different results.

2. One-dimensional and multi-dimensional optimization

   One-dimensional optimization handles the problems that have only one parameter. Multi-dimensional optimization deals with the problems with more than one parameter.

3. Static and dynamic optimization

   Dynamic optimization means time is one of the factors that affect the optimum solution. Static optimization means that the solution of the problem is not related with time.

4. Discrete and continuous parameter optimization

   Discrete parameter optimization solves problems whose parameters have a finite number of possible values. Continuous parameter optimization solves problems whose parameters have an infinite number of possible values.

5. Constrained and unconstrained optimization

   Constrained optimization takes the limitations to the parameters into consideration. Parameters of unconstrained optimization do not have limitations. They can take any value.

6. Minimum seeking and random optimization

   Since the output from the process is usually seen as the cost that needs to be minimized, optimization becomes minimization. The process is also called the cost function. Minimum seeking optimization is to find the optimum solution by starting from an initial set of parameter values and then search in a determinant direction.

This kind of optimization methods tends to be fast but is easily stuck in local optima. Random optimization also starts from an initial set of parameter values, but it guides itself by using the previous results combined with random choice. Random optimization tends to be slower but has greater chance of finding the global optimum. [9]

2.2 Minimum Seeking Algorithms

Minimum seeking algorithms are traditional optimization methods. We introduce two of them: exhaustive search and analytical optimization.

Exhaustive Search – Exhaustive search algorithm finds the global minimum by sampling the cost function as sufficiently as necessary. Since the evaluation of the cost function required is extremely large, exhaustive search methods are only suitable to problems with small number of parameters and in a limited search space.

Analytical Optimization – Analytical optimization uses calculus to find the minimum of the cost function. It can quickly find a minimum by computing the derivatives of the cost function with a single parameter. To functions with two or more parameters, gradient, Laplacian of the functions are calculated to find the extrema. It is difficult to find all the extrema if there are too many parameters. In addition, a search scheme needs to be applied to all the extrema to find the global minimum. Furthermore, "Continuous functions with analytical derivatives are necessary (unless derivatives are taken numerically, which results in even more function evaluations plus a loss of accuracy)." [9] These shortfalls make it difficult for analytical optimization to solve practical problems.

2.3 Natural Optimization Methods

Natural optimization methods model processes in nature. Like minimum seeking methods, natural optimization methods also start from an arbitrary point or points. They explore new points in the search space by analyzing the results that have got, combined with applying stochastic factors at the same time.

Simulated Annealing – Simulated annealing was introduced by Kirkpatrick and coworkers [12] in the early 1980s. This method models the process of annealing. It uses a control parameter, which is analogous to the annealing temperature, to control the speed of converging to the optimum.

The Genetic Algorithm – The genetic algorithm was introduced by John Holland [10] during the 1960s and 1970s. The idea originated from Darwin's theories of *natural selection* and *survival of the fittest*. In his book *Adaptation in Natural and Artificial Systems*, Holland [10] presented the concept of GA as a method of modeling the process of natural adaptation, and developed the theoretical framework for adaptation. De Jong [3] found the usefulness of GA for function optimization and began his study on finding optimal control parameters for GAs. David Goldberg [6], a student of Holland, further developed GA and made it popular.

2.4 Genetic Algorithms

In genetic algorithms, each candidate solution to a problem is represented as a genome, which includes one or more chromosomes (most applications of genetic algorithms use a single chromosome). A chromosome is a set of genes, each of which determines a particular aspect of the solution. An initial population of candidate solutions is then created. After that, a series of genetic operators are applied, driving the population to evolve so as to find the best solution(s).

The typical process of genetic algorithms works as follows:

a. An initial population is created randomly.

b. The individuals of the population are evaluated based on the fitness function that is used to measure how well the individuals perform. The better an individual performs, the higher its fitness score.

c. Two individuals are selected for reproduction. The higher an individual's fitness score, the higher its chance of being selected.

d. The two selected individuals reproduce, yielding one or more offspring(s). The usual technique of reproduction is to perform crossover.

e. With probability p, the newly created offspring(s) are randomly mutated.

f. Replace the old population with the new one, then go to step b if the termination conditions have not been met.

Each iteration of the above process is called a *generation*.

2.4.1 Components of A Genetic Algorithm

2.4.1.1 Encoding

To use a genetic algorithm, the first thing to do is to represent each solution as a set of genes. Each gene encodes a particular aspect of the solution. The most common encoding method is binary encodings, which encode solutions into binary strings (strings of 1s and 0s). Other methods include many-character and real-valued encodings, tree encodings, etc. Many-character and real-valued encodings define the value of each gene (or say, parameter) as a character from an alphabet of many characters or a real number, and thus are more natural for many problems. Which encoding works best? Some researchers (e.g., Janikow and Michalewicz [11]; Wright [17]; Haupt and Haupt [9]) indicate that real-valued encodings perform better than binary encodings. However, as Mitchell [13] says, "The performance depends very much on the problem and the details of the GA being used, and at present there are no rigorous guidelines for predicting which encoding will work best."

2.4.1.2 Basic GA Operators

The basic GA operators include selection, crossover and mutation.

Selection - This operator chooses individuals in the population for reproduction based on their fitness. Individuals with higher fitness are more likely to be selected, thus they are more likely to have offspring. "Selection has to be balanced with variation from crossover and mutation (the 'exploitation/exploration balance'): too-strong selection means that suboptimal highly fit individuals will take over the population, reducing the diversity needed for further change and progress; too-weak selection will result in too-slow evolution." [13] There are many selection methods. The most common ones include Roulette Wheel selection [6], Stochastic Universal sampling [2], Sigma Scaling [5], Elitism [3], Rank selection [1], Tournament selection [7], etc.

Crossover - Crossover is the feature that distinguishes GA from other optimization techniques. There are different approaches or different implementation details for crossover in different contexts of encodings. The simplest method is the *single-point crossover*. In the context of binary encodings, a crossover point is randomly chosen and the corresponding binary strings of the parents are exchanged up to this point to form new offspring. The difference in the context of many-character and real-valued encodings is the unit of parts being swapped is a whole gene (parameter), rather than a binary bit. Two-point and multi-point crossovers are extensions of the single-point crossover. Another common approach is *uniform crossover*, in which each bit/gene of the offspring is selected randomly from the corresponding bit/gene of the parents.

Mutation - Mutation is to randomly alter a small percentage of bits/genes of the offspring. The purpose of mutation is the same as crossover; that is to achieve variation. The mutation probability is usually very low, e.g. 0.01.

2.4.1.3 Convergence and The Termination Conditions

A genetic algorithm converges when all or most of the genes of the population are identical. John Holland proposed the schema theorem [10] for convergence analysis of genetic algorithms. A global convergence proof through a Markov chain model for genetic algorithms using elitism was presented by G. Rudolph. [15] An algorithm must comply strictly with the assumptions of this proof in order to converge. In practice, we usually apply some termination conditions to stop the algorithm. Some common examples are: a satisfactory answer has been reached; no improvement in the chromosomes; a predetermined number of generations or time has elapsed.

2.4.2 GAs vs. Other Traditional Optimization Methods

As an optimization technique, genetic algorithms are relatively new. However, they have achieved remarkable success during the last two decades. Compared with traditional optimization methods, some of GAs advantages include: [9]

- GAs search from a wide range of the solution space simultaneously instead of starting from a single solution to search through the space, thus they have more chance to reach the global optimum.
- GAs work well in solving complex problems with large but finite search space.
- GAs encode the parameters rather than search with the actual values of the solutions.
- GAs only use the fitness value of a solution to guide themselves through the search space. They do not require derivative information.
- GAs are easy to be implemented in parallel.
- GAs can handle optimization problems with a large number of parameters.

However, genetic algorithms have their disadvantages. A problem is that genes from comparatively highly fit (but not optimal) individuals may take over the population, thus making the algorithm converge on a local optimum. Since GAs need to sample and evaluate a large population of potential solutions, they are usually slower than other methods if they run on a serial computer. Moreover, genetic algorithms are not suitable for every optimization problem. Some simple problems can be solved easily and quickly by using traditional methods. [9]

2.4.3 Applications of Genetic Algorithms

"Genetic algorithms have been used in two ways: as techniques for solving technological problems, and as simplified scientific models that can answer questions about nature." [13]

Some examples of the application in problem solving include: evolving computer programs, data analysis and prediction, evolving neural networks, scheduling, signal processing, etc.

Some examples of the application in scientific models include: modeling interactions between learning and evolution, modeling sexual selection, modeling ecosystems, measuring evolutionary activity, etc.

3 THE TRUCKING PROJECT

In this section, we describe the details of the Trucking project in three parts: its composite elements, economy types and numerical analysis, and the simulation.

3.1 Elements

The elements of the Trucking project include a small country, four kinds of dealers (producers, consumers, retailers, and gas stations), trucks, and three kinds of commodities (crates, items, and gas). One of the differences between the current version and the previous one is that there are no controllers and managers in this version. Controllers and managers acted as information providers and mediators to trucks and retailers in the previous version to control their behaviors. They are not necessary any more since Trucking is not seen as a competition between programmers in this version.

3.1.1 The Country

The country is a square. It is made up of a grid of roads: 10 streets running from West to East and 10 avenues running from South to North (Figure 3-1). Grid points are 10 kilometers apart. Each intersection may be either empty or occupied by at most one dealer. We use (street number, avenue number) to indicate the intersection formed by the crossing of these two numbered roads. Then as shown in Figure 3-1, (1, 2) is the intersection of street No.1 and avenue No.2. As mentioned above, if it is not empty at (1, 2), there must be a producer, a consumer, a retailer, or a gas station there. Whatever it is, there must be only one.

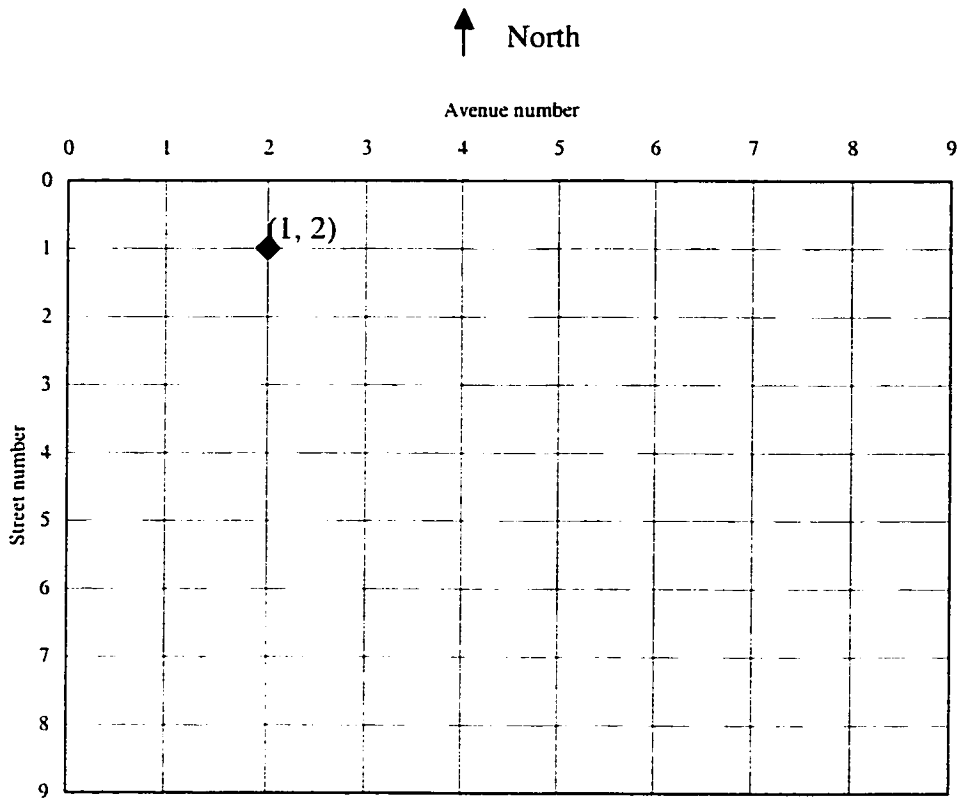

Figure 3-1 The country

Trucks run along the streets and avenues. Traffics on all roads are in both directions. Since the country is a square, not a torus, its north edge does not connect to the south edge, and the east does not connect to the west. A truck moving in a direction must change to another direction once it has arrived at the edge of the country and it wants to continue to move.

Traffic congestion is taken into consideration in this version of Trucking. A maximum number is added to limit the total number of trucks at a place at the same time, no matter traveling through or stopping there. This maximum number is predetermined as a simulation parameter. Once the maximum number is reached, that place becomes congested. No more trucks can move in. Only trucks that have already been at that place can move out. Congestion disappears after at least one truck has moved out.

3.1.2 Commodities

There are three kinds of commodities flowing in the country: crates, items, and gas.

The flow of gas is quite simple. Gas can only be sold by gas stations to trucks. Gas stations supply whatever amount of gas that trucks request.

A crate is actually a package of a number of items (100 items in this simulation). Crates are manufactured by producers. Trucks buy crates from producers and then sell them to retailers. After unpacking the crates, retailers get items and then sell items to trucks, which then sell the items to consumers. Consumers consume the items.

We can see that the flow of crates starts from producers, and then passes to trucks when trucks buy crates, and to retailers when retailers buy crates, where it turns into the flow of items. With trucks purchasing items from retailers and selling them to consumers, the flow of items moves to trucks from retailers, and finally ends at consumers. Therefore the number of crates in the flow determines the number of items in the flow. There is a limitation of the number of crates and items in the flow. This limitation is discussed in detail in Section 3.2 Economy Types and Numerical Analysis.

3.1.3 Producers

Producers manufacture crates and sell them to trucks. However, they do not sell actively. Only when trucks want to buy can producers sell crates. All producers sell crates at a fixed price. In this simulation, the price is $60 per crate. They may keep a stock of crates. A producer's warehouse is assumed with unlimited capacity, but its supply of crates is not guaranteed to be unlimited. The total number of crates that producers produce depends on which economy type the simulation is running. The economy types and their limitations to the number of crates that producers produce are discussed in Section 3.2 Economy Types and Numerical Analysis.

3.1.4 Consumers

Consumers buy items from trucks and consume the items. Like producers, they do not consume actively. Only when trucks want to sell can consumers buy items. Consumers buy items at a fixed price. In this simulation, the price is $1 per item. Consumers may maintain a stock of items. A consumer's warehouse is also assumed with unlimited capacity, but its capacity of consuming is not guaranteed to be unlimited. The total number of items that consumers can consume depends on which economy type the simulation is running.

3.1.5 Retailers

Retailers buy crates from trucks, unpack the crates to yield items, and sell the items back to trucks. They are passive too; that is they always wait for trucks to trade with them. They gain profit by selling items in a higher price than the price in which they buy a crate divided by the number of items yielded from the crate, just like dealers in real life. At the same time, since retailers need to rent space for keeping their stock, they must pay rent. The amount of rent depends on the size of the warehouse. For instance, a retailer pays $500 every 1000 simulation time units for a warehouse of 8 crates capacity. To keep the simulation simple, all retailers have the same size of warehouse. A retailer needs to ensure that its stock does not exceed the storage limit.

Each retailer has its buying and selling prices. These prices are given when the retailer is created. They are fixed during the retailer's lifetime. One major difference of this version from the previous one is that retailers and trucks may negotiate their trading prices in this version. How is negotiation performed? Let's look at the following example.

Truck: My selling price for a crate is $70.

Retailer: My buying price is $68.

Truck: Sorry, no deal.

Or:

Truck: My selling price for a crate is $69.

Retailer: My buying price is $73.

Truck: Fine! Let's trade at 0.5*(69+73) = $71.

In other words, the buying price must be higher than or equal to the selling price if they want to get a deal. Their actual trading price is a half of the sum of their announcing prices. However, this price is only the price for the current deal. Their original announcing prices are not changed.

Retailers become bankrupt by going into debt. Retailers that are bankrupt are removed from the simulation.

3.1.6 Gas stations

The only business of gas stations is to sell gas to trucks. We assume their supply of gas is unlimited. Therefore a truck's request of buying gas is always satisfied. To make things simple, all gas stations sell gas at a same price. In this simulation, it is $1 per litre. This price is fixed during the simulation.

3.1.7 Trucks

Trucks travel across the country transporting goods between dealers, which are distributed across the country. A truck may be: a PR truck moving crates from producers to retailers; an RC truck moving items from retailers to consumers; or both. A truck makes profit by keeping the selling price higher than the buying price of the same commodity. As mentioned above, producers sell crates at $60 and consumers buy items at $1, thus if a truck does not want to lose money, it must sell crates at a price higher than $60 and buy items at a price lower than $1. A truck negotiates a trading price with the retailer that it tries to make a deal with (see Section 3.1.5 Retailers).

We assume trucks move at a steady speed. In this simulation, their average speed is 60 km/hour; that is 1 km/minute. They consume gas in the course of moving. Their gas-consuming rate is fixed at 0.1 litres/km. When a truck is low on gas, it needs to find a gas station and purchase gas. A truck becomes inactive when it has run out of gas and is not able to buy gas immediately. Inactive trucks are removed from the simulation.

Each truck has a carrying capacity that is determined when the truck is created. Trucks must ensure their stock does not exceed the storage limits.

A truck may use a cell phone to call intersections of the country. If the called intersection has a dealer, the truck gets the dealer's information such as the type of the dealer, price, etc. If no dealers exist at that intersection, the truck knows the intersection is empty. No matter whether the intersection has dealer or not, the truck must pay a certain amount of money for the call.

Trucks move along the streets and avenues, and make deals at intersections. Sometimes a truck can encounter traffic congestions. It cannot move into the congested sites, but must wait for the congestions to disappear or go away.

3.2 Economy Types and Numerical Analysis

In this version of Trucking, the concept of economy types is introduced. Two economy types have been defined: supply-driven and demand-driven.

- "In a supply-driven economy, producers manufacture creates at a constant rate and consumers buy as many items as they can." [8]
- "In a demand-driven economy, consumers buy items at a constant rate and producers provide as many crates as necessary to keep up with the demand." [8]

Therefore, in a supply-driven economy consumers do not need to consider how many items they can consume. They simply consume as many as provided by trucks. That means a truck will never be refused when it wants to sell items to consumers. On the other hand, it is possible that the truck is turned down when it requests to buy crates from a producer because the producer's turnout is limited by the rate. Similarly, in a demand-driven economy producers' turnouts are not limited. They simply manufacture as many as requested by trucks. That means a truck will never be refused when it asks to buy crates from producers, but it can be turned down when it tries to sell items to a consumer.

The concept of economy types embodies the prevalent reality of limited resources. Both of these two types are provided as options in the simulation. Their rates will be computed later in this section.

Symbol	Value	Unit	Meaning
N_r	10		Number of roads in the country
d	10	km	Distance between grid points
N_i	100		Number of items in a crate
v	1	km/min	Average speed of a truck
k	0.1	litres/km	Gas-consuming rate of trucks
C_g	1	\$/litre	Cost of gas
π	0.4		Profitability rate
C_i	1	\$	Cost of one item
N_t			Number of trucks
N_c			Average carrying capacity of trucks
μ			Estimated factor of the length of a round trip
λ			Estimated proportion of the total time spent on traveling

Figure 3-2 Basic simulation parameters

Let the simulation time unit be 1 minute. Figure 3-2 lists some of the simulation parameters.

The values of the first eight parameters in the table are built into the simulation. They have already been mentioned in the previous sections except π. N_t and N_c may vary in the process of the simulation. At last, μ and λ are two estimated values that are required to be inputted at the beginning of the simulation. They are only used to predict the performance of the simulation. π, μ and λ are explained below.

π is the profitability rate that indicates the profit space of trucks and retailers. π satisfies $0 < \pi < 1$. According to Figure 3-2, the cost of buying one crate from a producer is $(1 - \pi)N_iC_i$; the total profit made from a crate, which is the difference between the money earned by selling the items to consumers and the money spent on buying the crate, is πN_iC_i. Given $\pi = 0.4$ as shown in Figure 3-2, we get the cost of a crate is $(1 - 0.4) * 100 * 1 = 60$, as mentioned in Section 3.1.3 Producers, and the profit trucks and retailers can get from a crate is $0.4 * 100 * 1 = 40$.

Before explaining parameter μ, the concept of "round trip" needs to be introduced first. A round trip is the journey within which a truck acts as a PR truck followed by as an RC truck. In other words, during a round trip, a truck performs the following actions in order: go to a producer to buy crates, go from the producer to a retailer (R1) and sell the crates, go from R1 to a retailer (R2) (R2 and R1 may be the same retailer) to buy items, go from R2 to a consumer to sell the items, go to a producer. Trucks are actually not necessary to move in round trips. This concept is only used to estimate the performance of trucks. The average length of a round trip is measured by multiplying the length of a road by factor μ. In our simulation, typically $1 < \mu < 4$. For instance, let $\mu = 3$, the road length is $N_r * d = 10 * 10 = 100$ km according to Figure 3-2, then the average length of a round trip would be $\mu N_r d = 3 * 10 * 10 = 300$ km.

A truck spends time on traveling, trading crates or items, and buying gas during its lifetime. Parameter λ is the estimated proportion of the total time spent traveling. It satisfies $0 < \lambda < 1$. For example, if $\lambda = 0.6$, that means in average a truck spends 60% of its time traveling and the remaining 40% trading crates or items, and buying gas.

To make the simulation meaningful, "the flow of goods through the system must be sufficient to ensure profitability and the carrying capacity of trucks must be sufficient to support this flow." [8] How many crates (or items) are sufficient? The analysis is shown below.

The following calculations apply to a single truck making a round trip.

Distance traveled:	$\mu N_r d$
Cost of gas:	$\mu N_r d k C_g$
Time spent traveling:	$\mu N_r d / v$
Total time for the round trip:	$\mu N_r d / \lambda v$
Profit made from one crate:	$\pi N_i C_i$

Profit made from Nc crates: $N_c\pi N_i C_i$

In order for a truck to make a profit, the money earned during a round trip must be more than the money spent on gas for the trip; that is $N_c\pi N_i C_i > \mu N_r dkC_g$. [8] According to Figure 3-2:

- The profit made from N_c crate is: $N_c\pi N_i C_i = N_c * 0.4 * 100 * 1 = 40N_c$ ($)
- The cost of gas spent on the round trip is: $\mu N_r dkC_g = 3 * 10 * 10 * 0.1 * 1 = 30$ ($)

Obviously, $40N_c > 30$ ($N_c \geq 1$). So it is possible that the truck can make a profit.

One truck needs $\mu N_r d / \lambda v$ minutes to sell N_c crates. There are N_t trucks running simultaneously in the country, so in $\mu N_r d / \lambda v$ minutes N_t trucks can sell $N_c N_t$ crates. In another word, N_t trucks can sell $\lambda v N_c N_t / \mu N_r d$ crates in one minute. Therefore, a supply-driven simulation must ensure that the manufacture rate of producers is at least $\lambda v N_c N_t / \mu N_r d$ crates per time unit, and a demand-driven simulation must ensure that the consumption capability of consumers is not lower than $\lambda v N_c N_t N_i / \mu N_r d$ items per time unit. [8] $\lambda v N_c N_t / \mu N_r d$ crates per time unit and $\lambda v N_c N_t N_i / \mu N_r d$ items per time unit are set as the manufacture rate of producers and the consumption capability of consumers respectively in this simulation.

3.3 The Simulation

The control program first reads parameters from the user, then creates and initializes the country, all dealers (producers, consumers, retailers, and gas stations), and trucks. After that, trucks begin to run. So does the evolutionary process. Here, several problems need to be addressed.

The first is the positions of dealers and trucks. In the simulation, the positions of producers and consumers are predetermined, not randomly created. They are given by the user at the time of initialization. "The idea is that producers are in 'rural districts' near the boundaries of the country and consumers are in 'urban districts' near the centre of the country." [8] Retailers and gas stations are positioned randomly, but not on top of the existing dealers. A dealer is fixed once it is positioned. If a retailer is removed from the simulation, the intersection that it occupied is released. Trucks are initialized at random locations.

Secondly, trucks and retailers are allowed to run for a period of time before they are evaluated. The length of this period is determined by the user as a simulation parameter. At the end of each period, inactive trucks and retailers are removed and all active trucks and retailers are evaluated. After that the selected individuals perform adaptation. A major difference from the previous version is that all active trucks and retailers will remain in the simulation. They make up of next generation's population along with the newborn individuals. Moreover, in the following generation they will resume their final statuses and activities of the previous generation.

Next, different population policies are applied to retailers and trucks. Retailers have a constant population. An initial population of retailers of size P_r is created at the beginning of the simulation. In the process of evolution a new retailer can be born only when a retailer has been removed. The actual population size of retailers may be less than P_r, but it is not allowed to exceed P_r. In contrast to retailers, trucks have a variable population. Trucks who are dead are removed from the simulation. New trucks are born whenever there are enough healthy trucks at the time of adaptation.

Finally, there are three conditions, any one of which leads to the termination of the simulation. They are: there are no active trucks left; there are no retailers left; a predetermined number (given by the user) of simulation time units have elapsed.

4 THE GENETIC ALGORITHMS FOR TRUCKING

As described above, Trucking is a complex search problem. To find out the most profitable trucks and retailers, we need to find out the strategies they use. Two genetic algorithms that are used in the present Trucking simulation are described below.

4.1 The Genetic Algorithm for Trucks

4.1.1 Encoding

A truck has many activities when it is active. It moves around the country, finds deals, buys and sells crates, buys and sells items, buys gas, and makes phone calls. Sixteen areas of strategies are developed to represent a truck's solution. We encode these areas of strategies into a single chromosome consisting of 16 genes. They are described below.

1. G_g - Determine the lowest gas limit. We divide the gas tank into 20 levels. Suppose that the truck has γ litres of gas and its gas tank capacity is Γ. The truck will search for gas if

$$\frac{\gamma}{\Gamma} < \frac{G_g}{20}$$

Here, $0 \leq G_g \leq 20$. We have the following situations:

- If $G_g = 0$, the lowest gas limit is 0; that means the truck will never look for gas. It will eventually run out of gas and become inactive.

- If $G_g = 20$, the lowest gas limit is 20; that means the truck will always search for gas unless it has full tank of gas ($\gamma = \Gamma$). This kind of trucks has little chance to do other trading.

- If $0 < G_g < 20$, the truck will search for gas whenever the amount of gas in its tank (γ) falls below

$$\frac{G_g}{20} * \Gamma$$

2. G_s - Determine the selling price of crates. Let T_s be the price at which the truck sells crates to retailers, then T_s is calculated as follows:

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

Here, $0 \leq G_s \leq 255$. We have the following situations:

- If $G_s = 0$, $T_s = N_i C_i \ (1 - \pi)$, which is the price at which producers sell crates to trucks. That means such trucks do not earn money in the PR trips.

- If $G_s = 255$, $T_s = N_i C_i$. The truck will sell crates at the highest price. If the truck succeeds in selling one crate, it will earn all profits created from a crate only in the PR trip and leave no profit margin for the crate's journey there after.

- If $0 < G_s < 255$, the truck will earn part of the profits created from crates in the PR trips. The remaining profit margin will flow to the remaining journey of the crates.

3. G_b - Determine the buying price of items. Let T_b be the price at which the truck buys items from retailers. T_b is calculated as follows:

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

Here, $0 \le G_b \le 255$. We have the following situations:

- If $G_b = 0$, $T_b = 0$. The truck will not get any item unless a retailer gives to it for free.
- If $G_b = 255$, $T_b = C_i$. The truck will buy items at the highest price.
- If $0 < G_b < 255$, the truck will buy items at a price between 0 and C_i.

4. G_α - Determine the proportion of capital that will be transferred to the child truck. Let α be the proportion of capital that will be transferred to the child truck during reproduction. We define

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

Here $0 \le G_\alpha \le 100$. Let c be the capital of a truck. In the case of reproduction, suppose T1 and T2 are the parent trucks, and T is the new created child truck. We use "object oriented" notation, so that T1.c is the capital of truck T1, etc. Then the transfers of capital are:

$$T1.c \leftarrow (1 - T1.\alpha) * T1.c$$

$$T2.c \leftarrow (1 - T2.\alpha) * T2.c$$

$$T.c \leftarrow T1.\alpha * T1.c + T2.\alpha * T2.c$$

- If $G_\alpha = 0$, the truck does not transfer money to its child.
- If $G_\alpha = 100$, the truck transfers all of its capital to the child.
- If $0 < G_\alpha < 100$, the truck transfers part of its capital to the child.

5. $G_{capacity}$ - The truck's carrying capacity. Its unit is crates. It satisfies $1 \le G_{capacity} \le 10$.

6. G_{tank} - Capacity of the truck's gas tank. Its unit is litres. It satisfies $1 \leq G_{tank} \leq 60$.

7. $G_{reserve}$ - The strategy that determines the amount of money that the truck reserves for buying gas. It has three values.

0: Reserve $0.

1: Reserve just enough money for buying a full tank of gas.

2: Reserve money for buying enough gas to travel the longest distance across the country. According to Figure 3-1, the longest distance across the country is the distance between (0,0) and (9,9) or between (0, 9) and (9,0), which is calculated as: $2N_rd = 2 * 10 * 10 = 200$ (km). The cost of gas for traveling this distance will be $2N_rdkC_g = 200 * 0.1 * 1 = 20$ ($).

8. G_{corner} - The strategy that determines whether the truck needs to move to the nearest corner right after creation. It has two values. Each value and the corresponding meaning are described as follows.

0: Do not go to the nearest corner, but stay at the initial position.

1: Go to the nearest corner.

9. G_{scan} - The strategy that determines whether the truck scans the country after the strategy G_{corner}, but before looking for deals. It also has two values.

0: The truck scans the whole country to mark down the locations of dealers before looking for any deal.

1: The truck does not scan the country.

10. G_{deal} - The strategy that determines how the truck looks for deals. It has three values.

0: The truck usually looks for a deal after it has completed one. However, in the course of trying to complete the current deal (say d1), if the truck discovers that it can make another deal (say d2) with the dealer at its present location, it will make the deal (d2) and then continue to complete the holding deal (d1) if the holding deal is still valid (e.g. still have room available for at least one crate if d1 is to buy crates).

1: The truck always looks for a deal and then goes to complete this deal. The difference with the above scheme is that it never makes other deals before the completion of the current deal even if it is at the right intersection where a dealer offers a good deal of other kinds.

2: The truck does not look for deals beforehand. It only keeps traveling across the country. It will search for gas when it finds itself low on gas. When the truck arrives an intersection where a dealer is located, it will check if it can make a deal with the dealer. If it can, it makes the deal; otherwise it just passes the dealer and goes to the next intersection. The truck does not memorize any information of dealers except the locations of gas stations. The truck will ignore G_{scan}, $G_{priority}$, G_{phone} and G_{badbuy} with this scheme.

11. $G_{priority}$ - The strategy that determines the priority of deals. It has three values as well.

0: In descending order: buy gas when the amount of gas in the tank falls below the lowest gas limit; sell crates if any; sell items if any; buy crates and buy items in turn.

1: In descending order: buy gas when the amount of gas in the tank falls below the lowest 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 amount of gas in the tank falls below the lowest gas limit or the truck is right at a gas station, otherwise check its status to see what kinds of deal it may make (for example, if the truck's status is: has at least one crate in stock; has no items; has capital that is enough for buying a crate; has space available for at least one crate, the deals it may make are: sell crates; buy crates; buy items), then choose the closest deal. G_{best} is useless with this scheme.

12. G_{best} - The strategy that determines the criteria of the best retailer. It has three values.

0: The one that offers the best price.

1: The one that is at the nearest location with a good price. What is a good price? When the truck's current deal is selling crates, a good buying price is the one that is higher than or equal to the truck's selling price. When the truck's current deal is buying items, any selling price that is lower than or equal to the truck's buying price is a good selling price.

2: When the deal is selling crates, in addition to calculating the money the truck will gain on selling crates, also calculate the cost on the gas consumed in order to complete the deal, then the best retailer is the one that the truck can gain the biggest profit from. When the deal is buying items, in addition to calculating the money the truck will spend on buying items, also calculate the cost on the gas consumed in order to complete the deal, then the best retailer is the one buying items from which the truck will cost the least.

13. G_{phone} - The strategy that determines whether the truck uses the cell phone and how it uses the cell phone. It has three values.

0: The truck does not use the cell phone. With this scheme the truck only uses the information kept in its memory, some of which could be out of date.

1: Use the cell phone to verify the dealers' information, or when the truck has not enough information for making a decision, use the cell phone to get new information of intersections chosen randomly. Maximum is five calls in the latter situation.

2: This scheme is almost the same with the above except that the truck does not randomly choose intersections to get new information, but starts from the nearest intersections and extends to the father ones gradually. The maximum is five calls again.

14. $G_{badsell}$ - The strategy that determines whether the truck permits unprofitable sells. It has two values.

0: The truck sells crates only at a good price; that is at a price higher than or equal to its selling price.

1: The truck sells crates at a good price if possible. However, after it has failed continuously for at least three times in selling the crates, the truck sells the crates even at a bad price that may lead to a capital loss.

15. G_{badbuy} - The strategy that determines whether the truck permits unprofitable buys. It has two values as well.

0: The truck buys items only at a good price; that is at a price lower than or equal to its buying price.

1: The truck buys items at a good price if possible. However, after it has failed continuously for at least three times in buying items, the truck buys even at a bad price that may lead to a capital loss.

16. $G_{congestion}$ - The strategy that determines how the truck responds to traffic congestions. We assume a truck always has a destination to go when it is moving. $G_{congestion}$ has three values.

0: Stay still until the congestion disappears. Then resume the journey.

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.

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 respond to traffic congestions	[0, 2]	3

Figure 4-1 Summary of truck's genes

The 16 genes are summarized in Figure 4-1. The total number of trucks can be explored in the search space is approximately:

21 * 256 * 256 * 101 * 10 * 60 * 3 * 2 * 2 * 3 * 3 * 3 * 3 * 2 * 2 * 3

= 972,790,589,030,400

The actual number will be a little smaller because changing a gene does not always change the behaviour of a truck. Take for example, two trucks that both have '2' for the gene G_{deal} and their values of corresponding genes are the same except G_{scan}, $G_{priority}$, G_{phone} and G_{badbuy}. These two trucks will have identical behaviour even if the values of their G_{scan}, $G_{priority}$, G_{phone} and G_{badbuy} genes are different. This is a huge space. Genetic algorithms are well known to solve such complex optimization problems with finite but large search space.

Real-valued encoding is used. The chromosome is composed of 16 fixed-length and fixed-order genes. Each gene is represented by an unsigned short integer.

4.1.2 Fitness

There is not a formula served as the fitness function of trucks in this simulation. The fitness of a truck is measured by the profit earned during the evaluated generation. In detail, we say the value of a truck is the sum of the following four quantities: its capital, the value of the crates in stock (VC), the value of the items in stock (VI), and the value of the gas in the tank (VG). VC is computed by multiplying the selling price of crates by the number of crates in stock. In the similar way, VI is obtained by multiplying the selling price of items by the number of items in stock. VG is got by multiplying the price of gas by the amount of gas in the tank. The profit a truck earned during a generation is the difference of the truck's value between the time of evaluation and the beginning of this generation. If the difference is positive, that means the truck made a profit in this generation. If the difference is zero or negative, the truck did not make a profit in this generation. The truck that made the largest profit has the highest fitness.

4.1.3 Selection

A truck will not be discarded unless it has become inactive. Each time when the simulation control program performs selection, the first step is to remove all inactive trucks out of the simulation.

Although all active trucks are kept for the next generation, not all of them can reproduce. The remaining trucks are divided into two parts: healthy trucks and unhealthy trucks. Healthy trucks are the trucks that have made profit in the current generation, and the ones that did not gain profit are unhealthy. In other words, healthy trucks have positive fitness values and unhealthy trucks have zero or negative fitness values. Though the unhealthy trucks are not discarded, they are not allowed to reproduce. Reproduction only happens to healthy trucks.

We apply a reproduction probability p to the remaining healthy trucks to yield the number of trucks that can reproduce. The number of reproducing trucks, N, is obtained by multiplying the number of healthy trucks by p. If N is odd, we subtract one from it so that N is always even. Then each pair of parent trucks is selected randomly from the healthy trucks for reproduction.

4.1.4 Crossover

After the trucks are selected and paired, they can reproduce. Each pair of parent trucks will perform crossover and yield a new child truck. The crossover method used here is uniform crossover. Half of the genes of the child truck are picked randomly from one parent and the remaining half are taken from the other. Unlike some other methods in which it is possible that the genes of the offspring are all taken from only one parent, this method makes the genes of every child truck taken from both parents evenly (crossover rate is 1), thus this method introduces more new chromosomes. Since the parents will remain in the new population, this method can explore the search space faster without destroying the existing solutions.

4.1.5 Mutation

Crossover can bring new chromosomes into the population. However, with the crossover method described above, the new chromosomes are only the recombination of the existing genes. No new gene values are introduced. So it is necessary to introduce mutation to child trucks to increase the variation. Mutation is performed to each gene of a new child truck with a very low probability known as mutation rate. A gene is mutated by replacing its value with a new random value.

A new truck will be created with the new chromosome at a random location, with initial capital given by its parents and with empty stock. There are two ways to have gas in its tank. The first one is that the simulation control program simply gives it full tank of gas to start. This way the new truck need not pay for the starting gas. Another solution is to deduct the cost of gas from the new truck's capital. If it has not enough money to pay for a full tank of gas, the new truck only can get its starting gas that costs half of its initial capital. The user can choose which solution to use in the simulation.

4.2 The Genetic Algorithm for Retailers

4.2.1 Encoding

In contrast to trucks, retailers are passive. They always wait for things to be done to them. A retailer, for example, can have something to do only when a truck approaches it and wants to make a deal with it. A retailer only responds to requests from a truck. So the behaviour of a retailer is simple: it buys crates when a truck sells to it, and it sells items when a truck buys from it. The only factors that a retailer has to control its profitability are its buying and selling prices. Retailers also need a strategy for reproducing. Therefore we encode these areas of strategies into a single chromosome consisting of three genes. They are described below.

1. G_b - Determine the buying price of crates. Let R_b be the price at which the retailer buys crates from trucks. R_b is calculated as follows:

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

Here, $0 \leq G_b \leq 255$. We have the following situations:

- If $G_b = 0$, $R_b = N_i C_i \, (1 - \pi)$, which is the price at which producers sell crates to trucks. In this case, the retailer buys crates at the lowest price. It can be satisfied only when a truck sells crates not requiring a profit.

- If $G_b = 255$, $R_b = N_iC_i$. The retailer will buy crates at the highest price.

- If $0 < G_b < 255$, the retailer will buy crates at a price between $N_iC_i\ (1 - \pi)$ and N_iC_i.

2. G_s - Determine the selling price of items. Let R_s be the price at which the retailer sells items to trucks, then R_s is calculated as follows:

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

Here, $0 \leq G_s \leq 255$. We have the following situations:

- If $G_s = 0$, $R_s = 0$. The retailer gives out items for free. Such retailers will surely become bankrupt in a short time.

- If $G_s = 255$, $R_s = C_i$. Which is the price at which consumers buy items. That is the highest selling price that a retailer can reach.

- If $0 < G_s < 255$, the retailer will sell items at a price between 0 and C_i.

3. G_α - Determine 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

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

Here $0 \leq G_\alpha \leq 255$. The way in which the capital is transferred is much like that of trucks. Let c be the capital of a retailer. In the case of reproduction, suppose R1 and R2 are the parent retailers, and R is the new created child retailer. Then the transfers of capital are:

$$R1.c \leftarrow (1 - R1.\alpha) * R1.c$$

$R2.c \leftarrow (1 - R2.\alpha) * R2.c$

$R.c \leftarrow R1.\alpha * R1.c + R2.\alpha * R2.c$

- If $G_\alpha = 0$, the retailer does not transfer money to its child.
- If $G_\alpha = 255$, the retailer transfers all of its capital to the child.
- If $0 < G_\alpha < 100$, the retailer transfers part of its capital to the child.

The three genes are summarized in Figure 4-2. The total number of retailers can be explored in the search space is

$256 * 256 * 256 = 2^{24} = 16777216$

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

Figure 4-2 Summary of retailer's genes

We use binary encoding to represent a retailer's solution. The chromosome is a 24-bit binary string. Each gene occupies 8 bits and is interpreted as an unsigned binary number in the range [0, 255].

4.2.2 Fitness

Just like trucks, retailers do not have a formulated fitness function either. The fitness of a retailer is the profit it has made during the evaluated generation. How to calculate the profit a retailer has gained in a period of time? We know that a retailer unpacks the crates it has into items. So we assume the stock of a retailer only consists of items. Then the value of a retailer is the sum of its capital and the value of the items in stock (VI). VI is obtained by multiplying the selling price of items by the number of items in stock. The profit is the difference of the retailer's value between the end and the beginning of the period. If the difference is positive, that means the retailer made a profit in this period. If the difference is zero or negative, the retailer did not earn a profit in this period. The retailer that makes the largest profit has the highest fitness.

4.2.3 Selection

A retailer will be removed from the simulation when it has become bankrupt. All retailers that are not bankrupt are kept for the next generation. Since retailers have a constant population, the reproduction of retailers is strictly controlled.

We also divide the remaining retailers into two parts: healthy retailers and unhealthy retailers. Healthy retailers are the retailers that have made profit in the current generation, and the ones that did not make profit are unhealthy. In other words, healthy retailers have positive fitness values and unhealthy retailers have zero or negative fitness values. Unhealthy retailers are not allowed to reproduce.

A quota is set up to control the number of new retailers can be created. Subtracting the allowed maximum number of retailers by the number of existing retailers yields the quota's value. A new retailer can be born only when the quota's value is greater than or equal to one.

When there is a need to create a new retailer, two parents will be selected randomly from the healthy retailers.

4.2.4 Crossover

Each pair of parent retailers performs crossover and yields a new child retailer. Here we use single-point crossover. A split point is randomly picked, then the first parent retailer copies its bits on the left of and including the split point to the child retailer, and the second parent retailer copies its bits on the right of the split point to the child retailer.

4.2.5 Mutation

Since the population of retailers is small, it is necessary to introduce mutation to achieve diversity. Mutation is performed to each bit of the new chromosome with probability p, where p is the mutation rate. A bit is mutated by inverting its value.

5 SIMULATION PARAMETERS

Some parameters are required to be inputted by the user prior to the simulation. They are classified into three categories. These three categories of parameters and their values used in the experiments discussed in Section 6 are discussed below.

The first is the initialization parameters. These parameters are used to initialize the simulation. They include the number and locations of producers, the number and locations of consumers, the number of gas stations, the number of retailers, the storage capacity of retailers, the initial capital and stocks given to each retailer, the initial number of trucks, and the initial capital and stocks given to each truck. We know from Section 3.3 that once created the positions of producers and consumers are fixed during the simulation.

· The number of producers does not influence the amount of goods flowing in the country. Neither does the number of consumers. However, the number and locations of producers and consumers may have impact on the traffic of the country. For example, if the number of producers is small and the producers are located closely, that can easily cause heavy traffic congestions because trucks wanting to buy crates all move to that small area where the producers are located. Increasing the number of producers can enlarge that area. Similarly, distributing the producers across the country widely can effectively prevent the congestion problem when the number of trucks is large. Four producers were established in the "rural districts" near the four corners of the country in the experiments. There were forty consumers. The consumers near the centre of the country were more than those located near the boundaries, as shown in Figure 5-1.

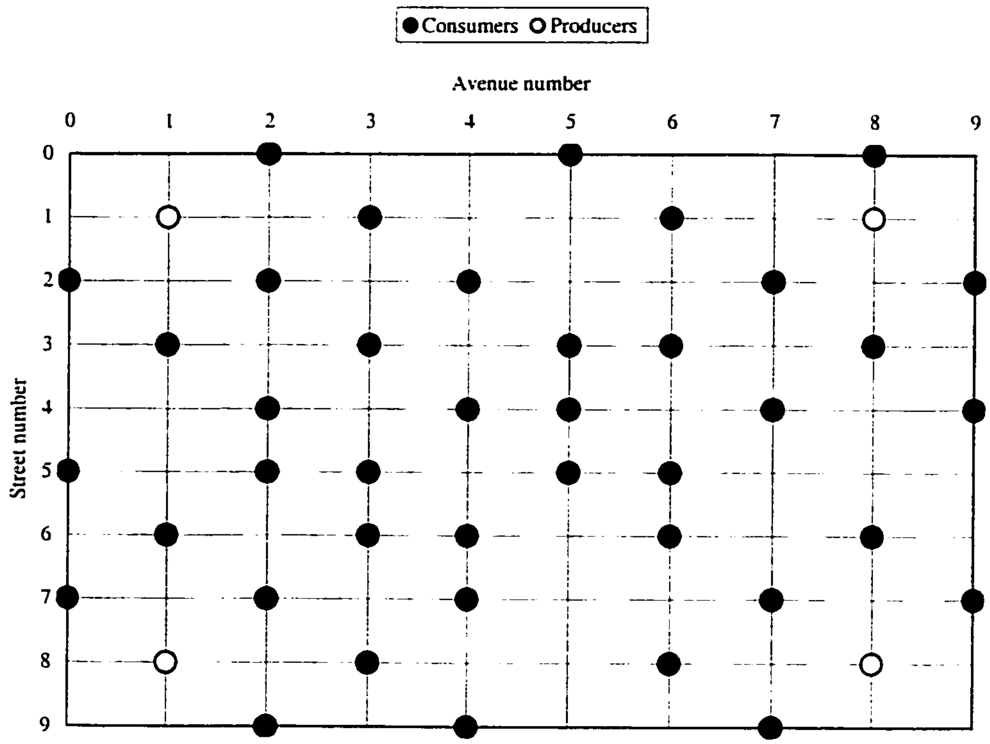

Figure 5-1 Locations of producers and consumers

· There were eight gas stations in the country.

· From the discussion above, producers, consumers and gas stations have already occupied 4 + 40 + 8 = 52 intersections. So retailers should be no more than 48 (there are total 100 intersections). The number of retailers in the experiments was set to thirty. All retailers have the same storage capacity that was 8 crates. They started with $3000 initial capital. No stock was given.

· In order to explore more search space, the population size of trucks must be large. However, the country is so small that it actually has become a limitation to the number of trucks that can be developed. In addition, trucks have a variable population. Its size changes with the death of inactive trucks and born of new trucks. Only the initial population size needs to be set. This initial size will be used to calculate the amount of goods flowing through the country per time unit; that is $\lambda v N_c N_t / \mu N_r d$ crates in a supply-driven economy and $\lambda v N_c N_t N_i / \mu N_r d$ items in a demand-driven economy (see Section 3.2). Here, N_t is the initial number of trucks. In the experiments N_t was set to 400. Each truck started with \$600 initial capital and full tank of gas. No stock was given.

The second category is the adaptation parameters. They include how long the adaptation interval will be, the reproduction rate and mutation rate of trucks, the mutation rate of retailers, and whether a new child truck needs to deduct the cost of initial gas from its capital (as described in Section 4.1).

- The adaptation interval should be long enough to let trucks and retailers fully apply their strategies and develop their business. In the experiments the adaptation interval was 4000 simulation time units.
- To control the population size of trucks, the reproduction rate should be kept low to prevent a rapid shooting up. It was 0.2 in the experiments.
- Mutation rates are usually low. 0.05 and 0.04 were used respectively as the mutation rates of trucks and retailers in the experiments. In other words, in average, the

probability of a truck's chromosome being mutated was 0.05 * 16 = 0.8, and the probability of a retailer's chromosome being mutated was 0.04 * 24 = 0.96.

- Child trucks are required to deduct the cost of initial gas from their capital in the experiments.

The third category includes all other parameters: the pay rent interval of retailers and the rent, the average cost of making a phone call, the estimated factor of the length of a round trip μ (see Section 3.2), the estimated proportion of the total time spent on traveling λ (see Section 3.2), the economy type of the simulation and the simulation time.

- Retailers pay rent every a period of time. The rent and interval in the experiments were $500 for every 1000 simulation time units.
- The average cost for a truck making a phone call was $2.
- The estimated factor of the length of a round trip was 3.
- The estimated proportion of the total time spent on traveling was 0.6.
- Both of supply-driven and demand-driven economies were experimented with different lengths of simulation time. A length of 400,000 simulation time units (100 generations) will be used as an example for the discussion of the simulation results.

A summary of the parameters that need to be inputted by users is listed in Figure 5-2.

Category	**Name**	**Value in the experiments**
	Number of producers	4

Initialization parameters	Locations of producers	See Figure 5-1
	Number of consumers	40
	Locations of consumers	See Figure 5-1
	Number of gas stations	8
	Number of retailers	30
	Storage capacity of each retailer	8 crates
	Initial capital of each retailer	$3000
	Initial stocks given to each retailer	0
	Initial number of trucks	400
	Initial capital of each truck	$600
	Initial stocks given to each truck	0
Adaptation parameters	Adaptation interval	4000 time units
	Truck's reproduction rate	0.2
	Truck's mutation rate	0.05
	Retailer's mutation rate	0.04
	Does a new child truck deduct the cost of initial gas from its capital?	Yes
Other	Pay rent interval of retailers	1000 time units
	The rent a retailer pays every pay rent interval	$500
	Average cost of making a phone call	$2
	Estimated factor of a round trip (μ)	3

parameters	Estimated proportion of the total time spent on traveling (λ)	0.6
	Economy type	Supply-driven / demand-driven
	Simulation time	400,000 time units (100 generations)

Figure 5-2 Summary of parameters to be inputted

Two measures are adopted to improve the performance of the simulation. One is to set a maximum congestion time. The maximum number of trucks at a same place at the same time is set to 5 in the simulation. When the number of trucks increases, congestion occurs frequently. The worst scenario is that too many congested trucks on the roads will lead to the result that no trucks can move and finally the system will crash. To prevent this situation from happening, a maximum congestion time is used to limit the length of time a truck being congested. If the length of time a truck being congested reaches the maximum limit, this truck will be cleared out of traffic and be removed from the simulation. The maximum congestion time is set to 2000 simulation time units. Another measure is that a truck or a retailer will be removed from the simulation if it has been unprofitable for consecutive 3 generations.

6 SIMULATION RESULTS AND DISCUSSION

To validate the algorithms described in Section 4, the simulation was carried out using a program developed in C++ on Windows platform. To achieve better performance, various parameter values were experimented. The simulation results talked in the remaining part of this section were got by running the simulation using the parameter values of Figure 5-2.

We will first examine the results in a supply-driven economy and then the results in a demand-driven economy. Four kinds of information were recorded at each generation to reflect the evolution of trucks: number of trucks (NT), total profit of trucks (TPT), average profit of trucks (APT) and healthy rate of trucks (HRT). NT is recorded at the beginning of each generation. TPT, APT and HRT are got at the end of the generation.

- NT is the total number of active trucks at the beginning of the generation, which is the number of trucks that survived the previous generation plus the number of new trucks created in the previous adaptation. These trucks will compete to survive and reproduce in the current generation.
- TPT is the sum of each truck's profit earned in the current generation.
- APT is TPT divided by NT.
- HRT is the ratio of the number of healthy trucks to NT.

Correspondingly, four kinds of information were recorded at each generation to reflect the evolution of retailers: number of retailers (NR), total profit of retailers (TPR), average profit of retailers (APR) and healthy rate of retailers (HRR).

6.1 Results of Experiment 1: Supply-driven Economy

We first examine the results of trucks in a run of supply-driven economy. Figure 6-1, 6-2 and 6-3 show the graphs of NT, TPT, APT and HRT.

Let's look at NT first. Its graph is given in all three figures. The number of trucks dropped at first due to the randomly created initial genomes. With the improvement of the overall fitness, more and more trucks could survive and more and more new trucks were born at the same time, pushing the population size going up. It was at a peak at the 35th generation, with 577 trucks. That was very crowded in a country with only 100 intersections. It caused severe traffic congestions. Since trucks that have been congested for longer than 2000 time units need to be cleared out, part of the trucks was removed for this reason after a period of time. Moreover, congestions brought poor performance, thus further decreased the number of surviving trucks and newborn trucks. That is a kind of over population. So the number of trucks fell straightly. After adjustment, NT went up again, and down due to the congestions, and oscillated during the remaining part of the simulation.

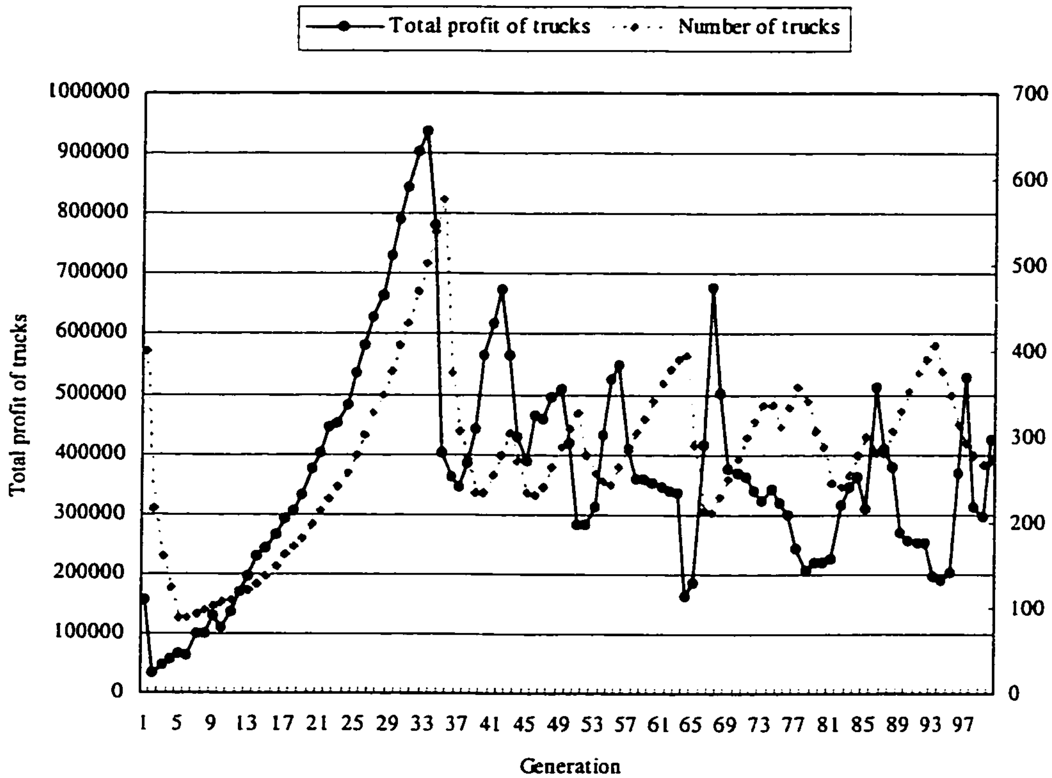

Figure 6-1 TPT and NT in a supply-driven economy

The graph of TPT is shown in Figure 6-1. Like NT, the total profit fell at first due to the randomly created initial genomes. After that it went up quickly following the rise of NT and reached a peak at the 33rd generation. But TPT dropped sharply when the number of trucks was approaching its highest point. As analyzed in the previous paragraph, a large number of trucks caused heavy traffic congestions and then brought the poor performance. That caused the drop. After that, TPT swung following the changes of NT. When observing carefully, we found that most upward curves of TPT were corresponding to downward curves of NT; most downward curves of TPT were corresponding to upward curves of NT. That means when the number of trucks increases the total profit of trucks is likely to decrease. On the other hand, when the number of trucks decreases the total profit is likely to increase. From the analysis of NT, the increase of NT brings poor performance of trucks, which causes the shrink of TPT; with the decrease of NT, the effect of traffic congestions lessens, and trucks can fully apply their strategies to make a profit, so TPT is likely to increase.

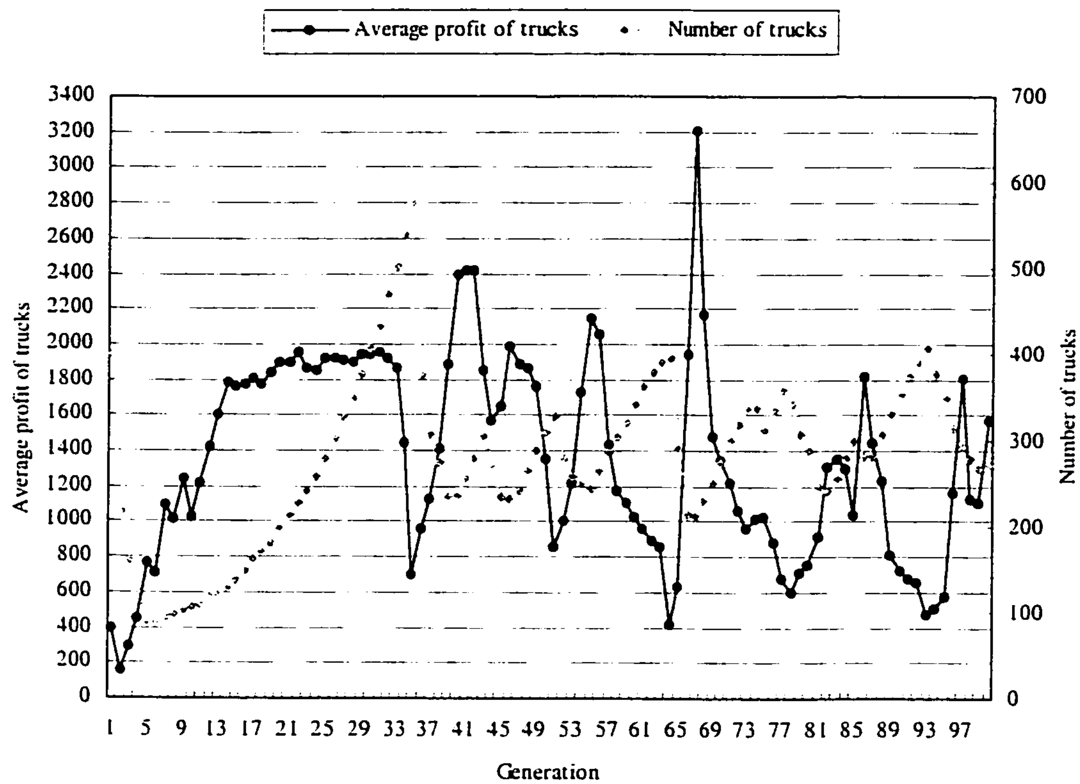

Figure 6-2 APT and NT in a supply-driven economy

Figure 6-2 shows the graph of APT. Its trend is similar with the TPT's. The average profit of trucks also decreased at first due to the randomly created initial genomes. It went up from the third generation and continued to rise following the increasing of NT. When there were too many trucks in the country, APT fell as the result of poor performance caused by traffic congestions. After that, like TPT, most upward-downward curves of APT were opposite to those of NT. If NT increases, both the competition between trucks and the chances of trucks being congested increase, thus the average profit trucks can make is likely to fall. If NT decreases, it lessens the competition and the chances of being congested, thus trucks are likely to be able to make more money.

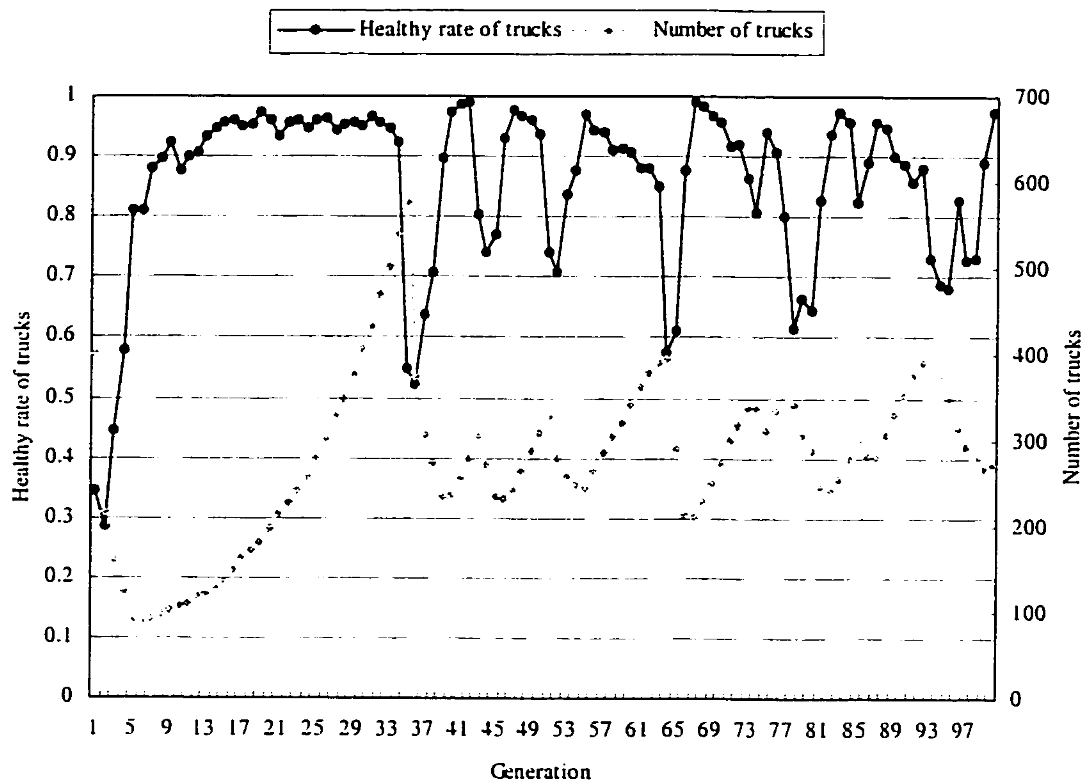

Figure 6-3 HRT and NT in a supply-driven economy

The graph of HRT is shown in Figure 6-3. As we discussed above, with the improvement of the overall fitness of trucks during the first 33 generations, the healthy rate of trucks improved too, from the lowest value of 0.29 up to the highest value of 0.97. More than 90 per cent of trucks successfully made profit from the 11th generation to the 34th generation. But HRT dropped every time when NT was at a peak. Obviously, poor performance made many of the trucks unprofitable. HRT oscillated after the 36th generation. When the number of trucks increased, the healthy rate decreased; the healthy rate increased when the number of trucks decreased.

From the three figures above, we know that trucks were performing very well in the first 33 generations. TPT rose nearly constantly. Although there were some downward sloping curves, the overall trends of APT and HRT were going up too. Trucks were becoming more and more profitable. The genetic algorithm really worked during this period. However, the performance of trucks turned unsteady after the over population. The profitability varied following the changes of the population size of trucks. This trend will probably last out the remaining of the simulation.

In addition, there are other factors affecting the performance of trucks: locations and prices of retailers. With some retailers turning into bankruptcy and some others being born, the population of retailers may change. A truck could earn a surprisingly high profit in a generation with many unsmart retailers whose buying prices are very high and whose selling prices are very low. However, the same truck may be unprofitable or even get a negative profit due to the changed environment in another generation. For example, truck t112 made \$2579.15 in the 1^{st} generation, but only got \$1.89 in the 4^{th} generation, and even worse it lost \$164.92 in the 5^{th} generation according to the result data.

Next we examine the results of retailers. The graphs of NR, TPR, APR and HRR in a supply-driven economy are shown in Figure 6-4, 6-5, 6-6 and 6-7.

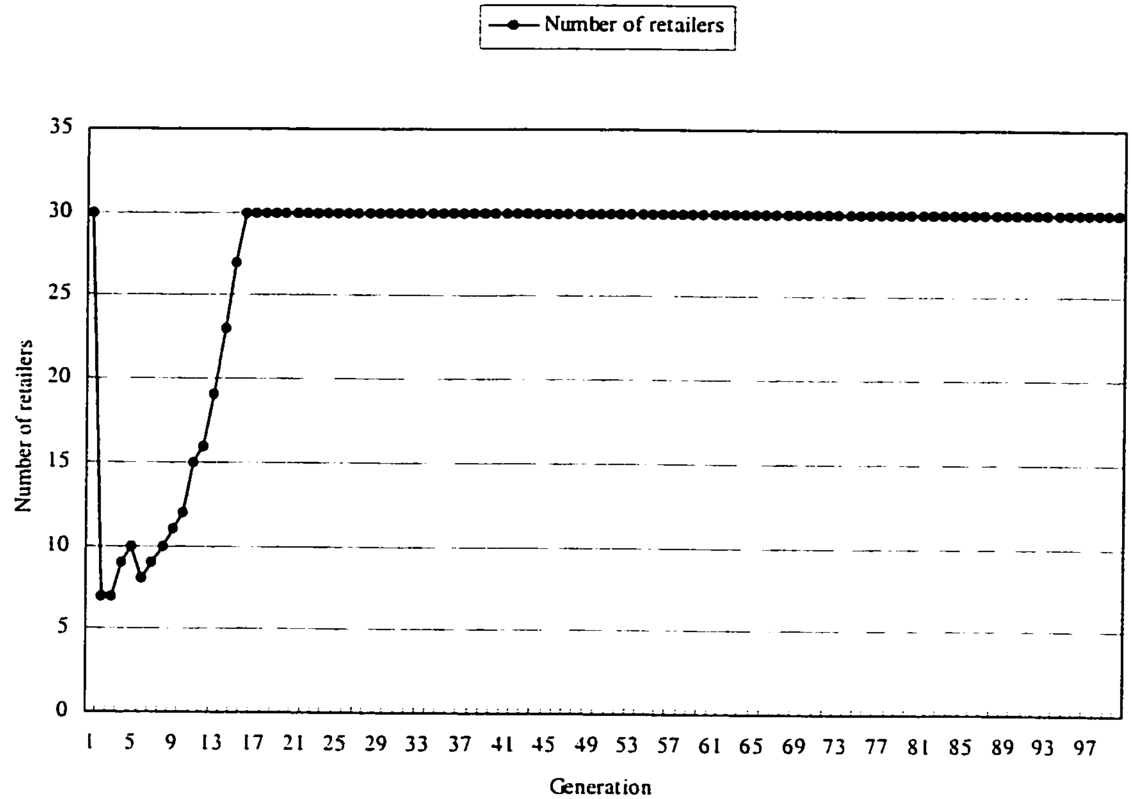

Figure 6-4 NR in a supply-driven economy

NR was 30 initially. Most of the retailers went into bankruptcy at the first generation due to the randomly created initial genomes. The population size came back to 30 eventually at the 16th generation. The number of retailers remained at 30 thereafter, since we set an upper limit to the population size of retailers.

TPR and APR developed slowly in the first 17 generations. After the retailer's population size jumped back to 30 and kept at this level, they went up quickly. That is because with the increase of trucks, the amount of deals that retailers got increased too (A condition that the flow of crates and items had not reached the maximum limit applied). Upon over population, most of the trucks were congested on the roads. That led to the starving of the retailers for not being able to trade to gain profit. Therefore TPR and APR fell sharply. After that, the development of TPR and APR were affected by the changes of NT and thus became unstable. Their figures swung up and down following the changes of the number of trucks.

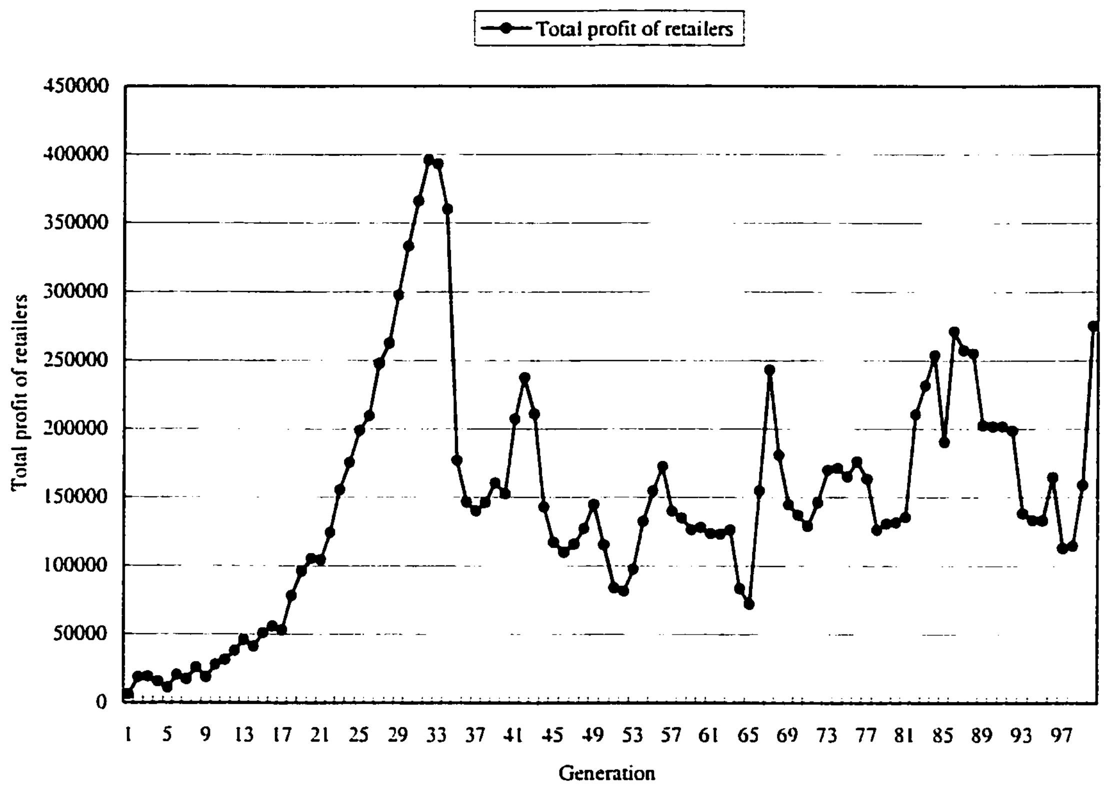

Figure 6-5 TPR in a supply-driven economy

During the first 34 generations, retailers made more and more money. That suggested that retailers were getting healthier and healthier. Their healthy rate reached 1.0 at the 27th generation and from the 29th to the 34th generation. However, the over population of trucks made many retailers starving and caused the deep drop of HRR. Like TPR and APR, HRR oscillated in the remaining part of the simulation.

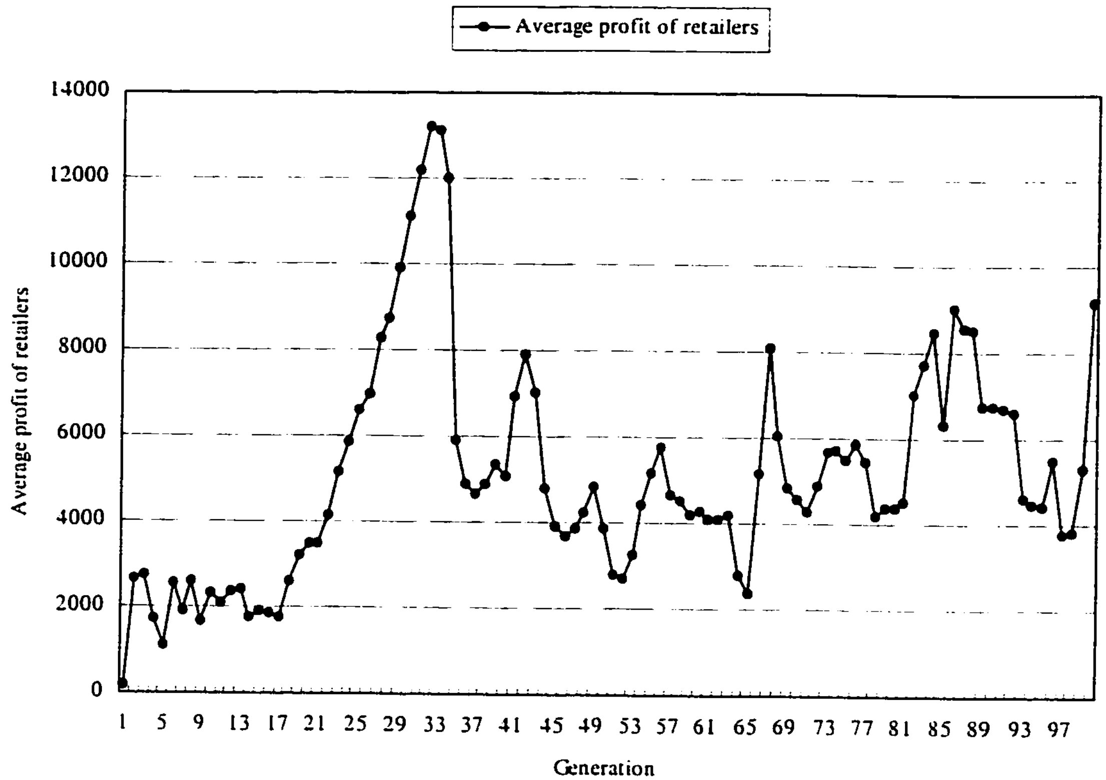

Figure 6-6 APR in a supply-driven economy

By comparing the graphs of total profit, average profit and healthy rate of trucks and retailers, one thing was found that most of the times when a graph of trucks arrived a peak (or a lull), the corresponding graph of retailers also arrived a peak (or a lull) exactly at that generation or at a very close generation. Take for example the total profit. The peak points were at:

TPT: 33, 42, 49, 56, 67, 84, 86, 97, 100

TPR: 32, 42, 49, 56, 67, 84, 86, 96, 100

The lull points were at:

TPT: 37, 45, 52, 64, 78, 85, 94, 99

TPR: 37, 46, 52, 65, 78, 85, 95, 97

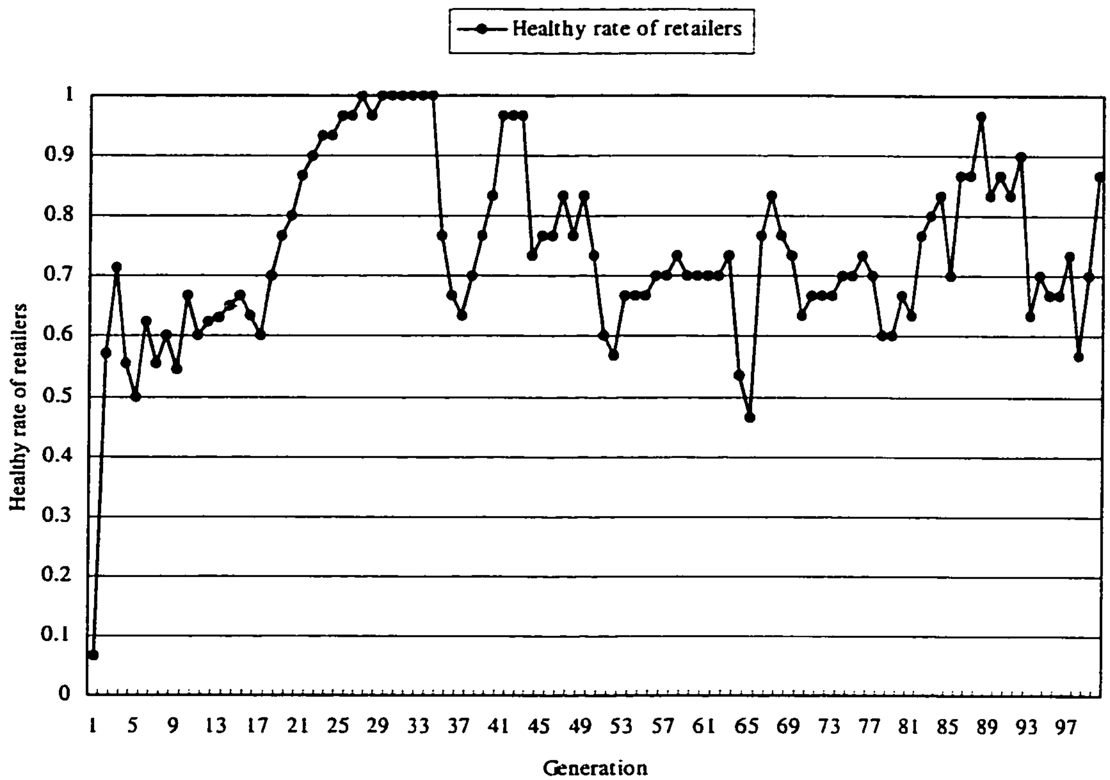

Figure 6-7 HRR in a supply-driven economy

Finally, to discover the gene structures of the most successful individuals, capital gains in the 100 generations of all remaining trucks and retailers are calculated by subtracting their final capitals by the capitals they got when they were created. The genes of the top 10 capital-gaining trucks are listed in Figure 6-8. The genes of the top 5 capital-gaining retailers are listed in Figure 6-9.

From Figure 6-8, G_{badbuy} had eight values of '1'. Seven out of ten values of $G_{badsell}$, $G_{reserve}$ and G_{corner} were the same, which were '1', '2' and '0' respectively. Value of '1' was excluded from $G_{congestion}$. Genes that have a large value range had more differences. However, they did have common if compared carefully. For example, nine out of ten values of G_{tank} were between 50 and 60. Even the last one, 47, was close to 50. Seven of G_s's values were between 73 and 85. Eight of G_b's values were over 220. Eight out of ten values of G_α were less than 10.

Genes	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
G_g(0~20)	11	13	17	18	14	12	8	11	8	18
G_s(0~255)	73	85	73	73	218	17	75	34	85	73
G_b(0~255)	229	224	230	247	252	230	247	229	206	185
G_α(0~100)	8	1	3	3	13	5	5	8	1	19
$G_{capacity}$(1~10)	10	7	5	8	4	8	8	2	7	4
G_{tank}(1~60)	57	55	58	58	47	51	50	58	50	53
$G_{reserve}$(0~2)	2	2	0	2	2	0	2	2	1	2
G_{corner}(0~1)	1	0	0	1	0	0	0	1	0	0
G_{scan}(0~1)	0	1	0	0	0	1	1	0	1	0
G_{deal}(0~2)	0	1	1	2	1	0	0	0	0	0
$G_{priority}$(0~2)	1	2	2	1	1	2	0	1	2	1
G_{best}(0~2)	0	1	2	0	2	2	2	1	1	1
G_{phone}(0~2)	2	0	0	2	0	1	1	2	0	1
$G_{badsell}$(0~1)	0	1	1	1	1	1	0	1	1	0
G_{badbuy}(0~1)	1	0	0	1	1	1	1	1	1	1
$G_{congestion}$(0~2)	0	0	2	2	0	0	2	2	0	2

Figure 6-8 Genes of the top 10 capital-gaining trucks in a supply-driven economy

The genes of the top 5 capital-gaining retailers have the similar situation. Among all five of G_b's values, four were between 80 and 95, with the remaining one just a little less than 80. Four out of five values of G_s were close to each other.

Genes	1	2	3	4	5
G_b(0~255)	90	79	90	82	94
G_s(0~255)	242	246	242	203	251
G_α(0~255)	21	175	20	16	175

Figure 6-9 Genes of the top 5 capital-gaining retailers in a supply-driven economy

6.2 Results of Experiment 2: Demand-driven Economy

Figure 6-10, 6-11 and 6-12 give the graphs of NT, TPT, APT and HRT in a run of demand-driven economy.

Compared with the graphs in the supply-driven run, the varying range of the truck's population size in the demand-driven run was larger, which was from 106 to 664. The NT graphs also indicate that on average DDE can support more trucks running in the country

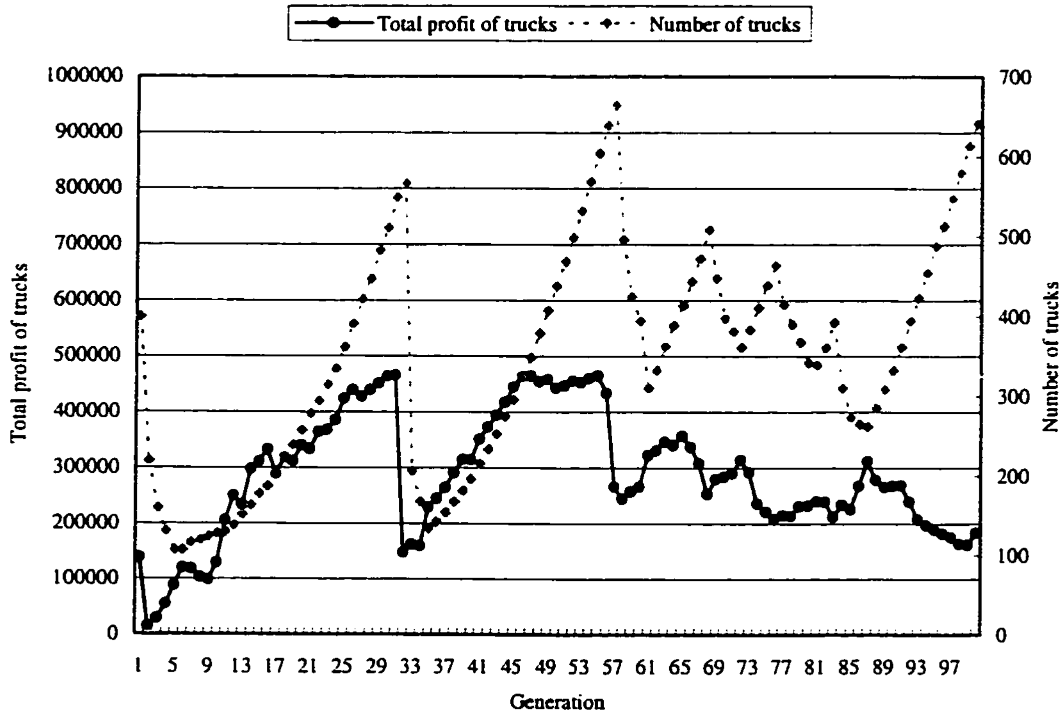

Figure 6-10 TPT and NT in a demand-driven economy

than SDE. In a supply-driven economy, the number of crates each time a truck can buy from a producer is limited by the manufacture rate. A truck can sell all of the items it carries to a consumer quickly without worrying about the consumer's capability. So trucks in a supply-driven economy likely need to go to producers often. In a demand-driven economy, since the number of items a consumer can consume is limited by the consumption capability, it is usually difficult for a truck to find a consumer who can accept all the items the truck carries at one time. Sometimes a truck needs to visit several consumers in order to sell its items out. On the other hand, a truck can buy whatever amount of crates it can afford from a producer. So trucks in a demand-driven economy likely need to travel a lot among consumers. In our experiments, there were only 4 producers, whereas the number of consumers was 40. That means trucks likely moved towards the 4 producers often in the supply-driven economy, while they likely moved around the 40 consumers often in the demand-driven economy. The chance of causing traffic congestions is much higher in the former situation than in the latter. This is the reason why the highest value of NT in the SDE was only 577 and it reached 664 in the DDE.

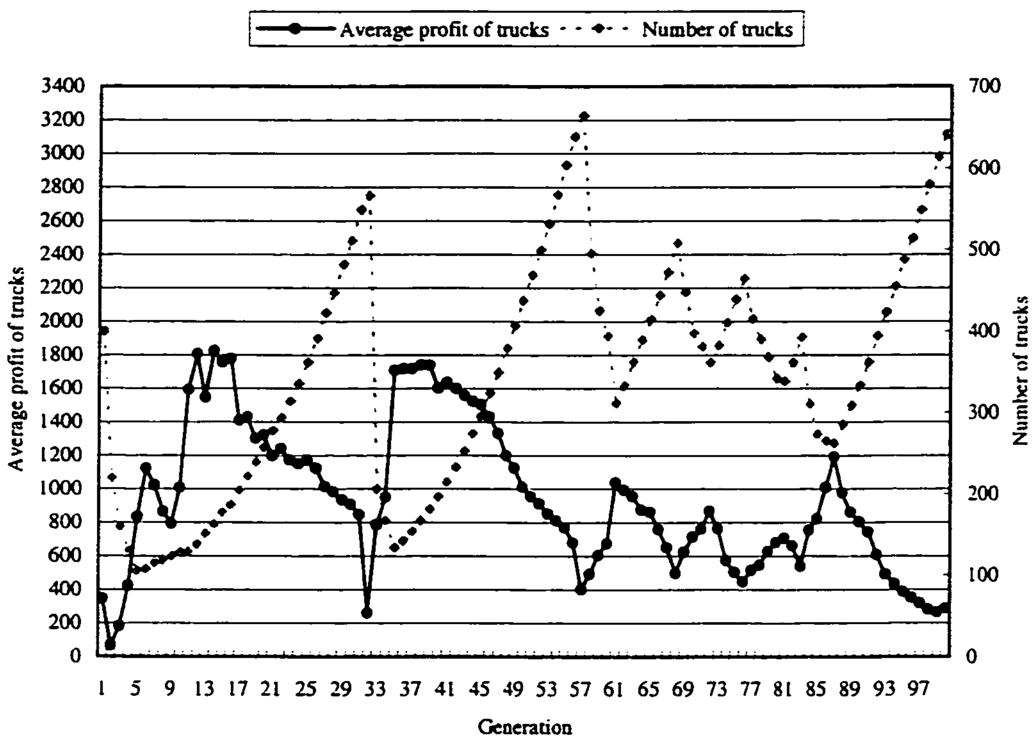

Figure 6-11 APT and NT in a demand-driven economy

More trucks mean more competition. With limited resources and more competitors, trucks in the demand-driven economy did not perform as well as in the supply-driven economy, as seen in Figure 6-10 and 6-11. The highest total profit was only \$465,684 at the 31st generation, compared with \$935,819 at the 33rd generation in the supply-driven economy. APT was not able to fully develop following the upward trend of NT, but began to decline from the 17th generation.

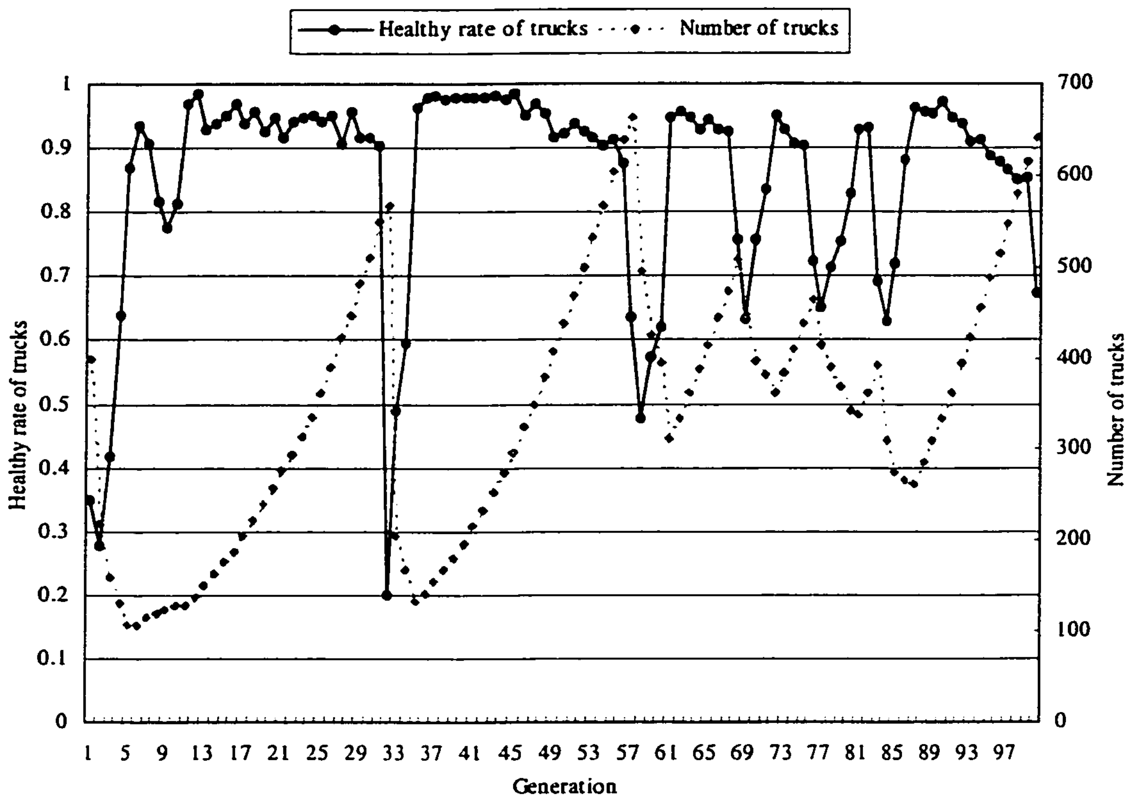

Figure 6-12 HRT and NT in a demand-driven economy

The results of retailers are shown in Figure 6-13, 6-14, 6-15 and 6-16.

Like in the supply-driven economy, NR was 30 initially. With the bankruptcy of the weak retailers composed of randomly created initial genes, the population size dropped at first. It came back to 30 at the 13th generation.

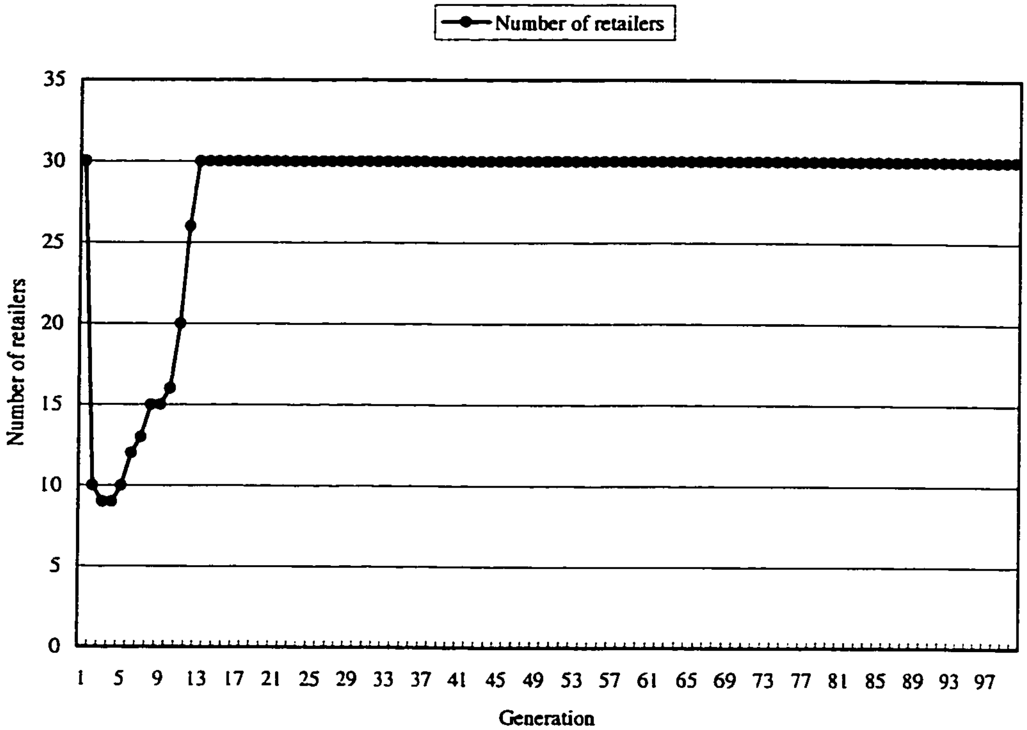

Figure 6-13 NR in a demand-driven economy

Although the average amount of deals each truck could make reduced because of the harder competition, the amount of deals that retailers could get did not decrease due to the increased number of trucks. From the figures, we can see that, on average, the profits retailers earned in the demand-driven economy were not less than the profits they earned in the supply-driven economy.

All of the TPR, APR and HRR look better than those in the supply-driven economy. Both of TPR and APR arrived a peak at the 31st generation. HRR remained at 1.0 from the 22nd to 31st generation. Retailers were doing well in the first 31 generations. Although TPR and APR swung up and down repeatedly during the later part of the simulation, their

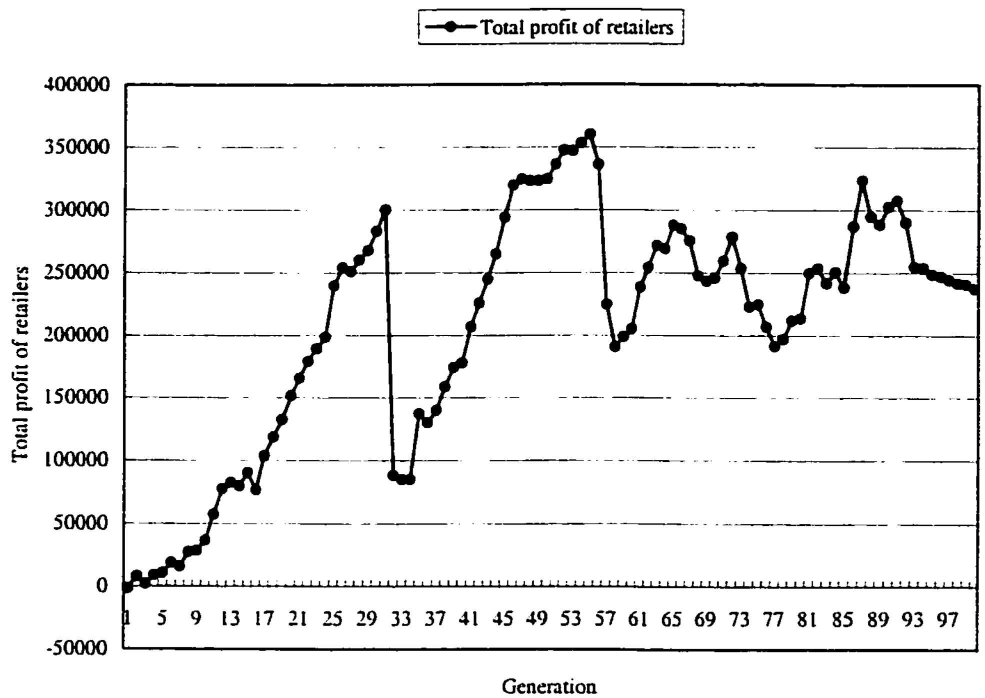

Figure 6-14 TPR in a demand-driven economy

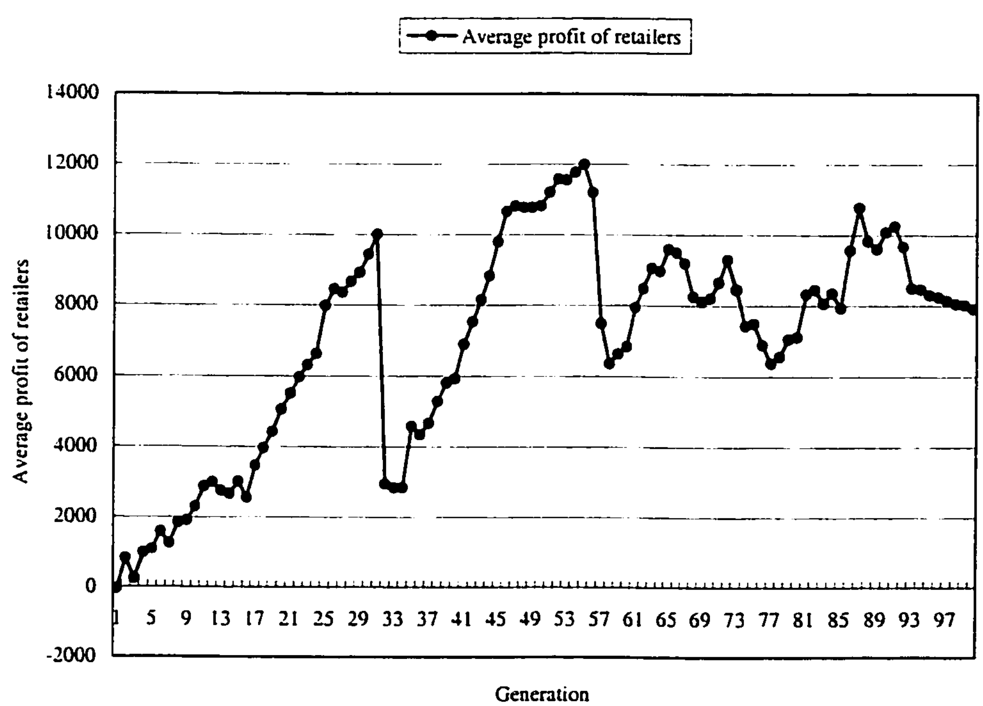

Figure 6-15 APR in a demand-driven economy

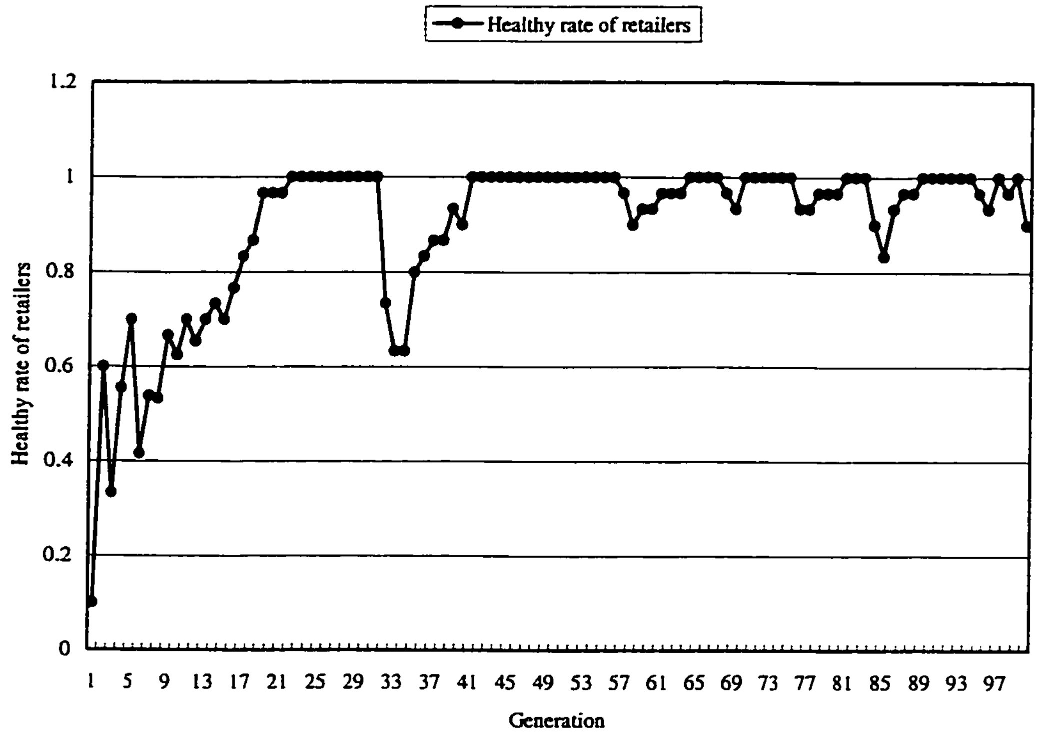

Figure 6-16 HRR in a demand-driven economy

overall trends were still going up. Over 80 per cent of retailers were healthy after the 35^{th} generation.

Just like in the supply-driven economy, the genes of the top 10 capital-gaining trucks are given in Figure 6-17, and the genes of the top 5 capital-gaining retailers in Figure 6-18.

Genes	1	2	3	4	5	6	7	8	9	10
G_g(0~20)	10	14	10	11	19	17	14	19	8	16
G_s(0~255)	57	44	5	73	20	69	57	20	55	13
G_b(0~255)	229	179	133	179	171	213	243	213	113	179

G_α(0~100)	19	4	3	21	3	17	18	4	18	61
$G_{capacity}$(1~10)	6	8	8	10	6	10	10	5	9	9
G_{tank}(1~60)	48	60	48	50	35	60	60	54	36	60
$G_{reserve}$(0~2)	2	2	2	2	2	2	2	2	0	2
G_{corner}(0~1)	0	1	1	0	1	0	0	0	0	1
G_{scan}(0~1)	1	1	1	1	0	1	1	1	1	1
G_{deal}(0~2)	0	0	0	0	0	1	0	0	1	0
$G_{priority}$(0~2)	1	1	2	2	2	2	0	0	2	2
G_{best}(0~2)	2	1	0	0	0	0	0	0	0	2
G_{phone}(0~2)	0	0	1	0	0	0	2	2	0	0
$G_{badsell}$(0~1)	1	1	0	1	0	1	0	1	1	1
G_{badbuy}(0~1)	0	1	0	1	0	0	0	0	1	0
$G_{congestion}$(0~2)	0	2	0	0	1	0	0	0	0	2

Figure 6-17 Genes of the top 10 capital-gaining trucks in a demand-driven economy

Genes	**1**	**2**	**3**	**4**	**5**
G_b(0~255)	120	104	158	104	96
G_s(0~255)	228	197	226	228	228
G_α(0~255)	44	44	177	44	177

Figure 6-18 Genes of the top 5 capital-gaining retailers in a demand-driven economy

The most frequent occurring gene values here are not the same as those observed in the supply-driven economy. Take for example, $G_{priority}$. Among the ten values in the supply-driven economy, '1' occurred five times, and then '2' had four occurrences. While in the demand-driven economy, six out of ten values were '2' and '1' only occurred twice. Look at G_b of retailers. Its values were mainly between 80 and 90 in the supply-driven economy. However, none of the G_b's values in the demand-driven economy was in this range. This suggests that in different economy types trucks and retailers need to use different strategies to survive and to make the most profit.

Since there are two kinds of agents (trucks and retailers) evolving at the same time and they interact with each other, their evolutions can be complicated. Whether the algorithms used in this thesis can converge has not been proved. Subsequent work will need to be done in this area.

7 CONCLUSION AND FUTURE WORK

In this thesis two genetic algorithms were used to evolve trucks and retailers in a small country. Two economy types were simulated as different forms of limitation to resources. The results indicate that in a supply-driven economy both trucks and retailers were evolving towards better solutions before over population of trucks. While the results of trucks in a demand-driven economy are not as optimistic as in a supply-driven economy, they did have upward slope curves indicating trucks were getting fitter during some periods. The results of retailers in the demand-driven economy reflect the algorithm works too. The results also disclose the significant influence of truck's population size on the performance of trucks and retailers during the evolution process.

In this project, a real-valued encoding method is used to encode the genome of trucks, and a binary encoding method is used to encode the genome of retailers. All of the three genes of retailers have values ranging from 0 to 255. Each gene can easily be represented as an 8-bit binary string. So binary encoding is suitable for retailers. On the other hand, among the 16 genes of trucks, some have values ranging from 0 to 255, some have values from 1 to 60, and some only have two values. The irregularity makes the real-valued encoding a more suitable method for trucks. Based on these two encoding methods, the selection, crossover, and mutation of trucks and retailers performed well in our simulation. As a result, the genes of trucks and retailers were improving, and individuals became more profitable when the population size of trucks was under control. This indicates that genetic algorithms are suitable for the Trucking problem.

As discussed in the Section 6, over population of trucks inhibited further evolving of both trucks and retailers. So solving this problem will be a major part of future work. A possible solution is to make the "birth rate" match the "death rate" by reducing the number of new trucks born in each generation and/or by increasing the number of trucks that die in each generation. To reduce the number of newborn trucks, we may set the reproduction rate lower. Another way is to add extra requirements to make it more difficult for a truck to be healthy. For example, trucks may be required to pay tax in each generation. We may even improve the standard of being a healthy truck by requiring that a healthy truck must be a truck that has earned at least a certain amount of money in the evaluated generation. Less healthy trucks mean less reproducing parents, so the number of newborn trucks will be less. A way of increasing the number of trucks that die in each generation is to let the trucks that did not make enough money die. Another approach to solve the over population problem is to set an upper limit to the number of active trucks in each generation. Another possible improvement to this problem may be to let trucks smarter by checking the road conditions beforehand.

To improve the results, other improvements to the project include developing better strategies for trucks, developing new diverse strategies for retailers, and varying the adaptation control parameters to find out their most suitable values. Some ideas of retailer's strategies are: making the size of the warehouse a gene of retailers; a retailer may choose whether it needs to reserve money for paying rent and the amount of the reservation.

REFERENCES

[1] Baker, J. E., "Adaptive selection methods for genetic algorithms," in J. J. Grefenstette, ed., Proceedings of the First International Conference on Genetic Algorithms and Their Applications, Erlbaum, 1985.

[2] Baker, J. E., "Reducing bias and inefficiency in the selection algorithm," in J. J. Grefenstette, ed., Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms, Erlbaum, 1987.

[3] De Jong, K. A., "Analysis of the behavior of a class of genetic adaptive systems," Ph.D. Dissertation, The University of Michigan, Ann Arbor, 1975.

[4] Edelstein, J. "Truckin': The Genetic Algorithm Way," Master thesis, Concordia University, 2001.

[5] Forrest, S., "Scaling fitnesses in the genetic algorithm," in Documentation for PRISONERS DILEMMA and NORMS Programs That Use the Genetic Algorithm, unpublished manuscript, 1985.

[6] Goldberg, D. E., Genetic Algorithms in Search, Optimizatoin, and Machine Learning, Addison-Wesley, 1989 .

[7] Goldberg, D. E., and Deb, K., "A comparative analysis of selection schemes used in genetic algorithms," in G. Rawlins, ed., Foundations of Genetic Algorithms, Morgan Kaufmann, 1991.

[8] Grogono, P. "Trucking," 2001.

[9] Haupt, R. L., and Haupt, S. E. Practical Genetic Algorithms, Wiley-Interscience, 1998.

[10] Holland, J. H. Adaptation in Natural and Artificial Systems, Ann Arbor, MI: University of Michigan Press, 1975.

[11] Janikow, C. Z., and Michalewicz, Z. "An experimental comparison of binary and floating point representations in genetic algorithms," Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann, 1991.

[12] Kirkpatrick, S., Gelatt, C. D., Jr., and Vechhi, M. P. "Optimization by simulated annealing," Science, volume 220, 1983, pp. 671-680.

[13] Mitchell, M. An Introduction To Genetic Algorithms, The MIT Press, 1996.

[14] Papoulis, D. "The truckin' project : experimenting with genetic algorithms," Master thesis, Concordia University, 2001.

[15] Rudolph, G. Convergence properties of evolutionary algorithms, Verlag Kovac, 1997.

[16] Stefik, M., Bobrow, D., and Mittal, S., "Truckin' and the Knowledge Competition," retrieved March 23, 2003 from http://www2.parc.com/istl/members/stefik/truckin.htm.

[17] Wright, A. H. "Genetic algorithms for real parameter optimization," in: Rawlins, G. E. (ed.), Foundations of Genetic Algorithms, Morgan Kaufmann, 1991.

APPENDIX A: RESULTS OF TRUCKS IN A RUN UNDER SUPPLY-DRIVEN ECONOMY

The following data are the results of trucks collected from the experiment discussed in Section 6.1.

Description of columns:

A: Generation sequence

B: Average profit of trucks

C: Number of trucks

D: Number of surviving trucks

E: Number of healthy trucks

F: Number of newborn trucks

G: Healthy rate of trucks

H: Total profit of trucks

Results:

A	B	C	D	E	F	G	H
1	391.309	400	203	138	13	0.345	156524
2	156.47	216	154	62	6	0.287037	33797.5

3	298.103	160	116	71	7	0.44375	47696.4
4	456.804	123	82	71	7	0.577236	56186.9
5	765.162	89	82	72	7	0.808989	68099.4
6	717.162	89	85	72	7	0.808989	63827.4
7	1096.58	92	89	81	8	0.880435	100885
8	1011.02	97	95	87	8	0.896907	98068.8
9	1244.59	103	98	95	9	0.92233	128193
10	1028.95	107	101	94	9	0.878505	110098
11	1224.02	110	109	99	9	0.9	134642
12	1422.74	118	112	107	10	0.90678	167883
13	1607.12	122	118	114	11	0.934426	196069
14	1779.33	129	126	122	12	0.945736	229534
15	1766.82	138	137	132	13	0.956522	243821
16	1778.41	150	148	144	14	0.96	266762
17	1807.61	162	157	154	15	0.950617	292833
18	1768.07	172	165	164	16	0.953488	304108
19	1841.38	181	181	176	17	0.972376	333290
20	1896.34	198	194	190	19	0.959596	375476
21	1893.77	213	208	199	19	0.934272	403373
22	1957.18	227	220	217	21	0.955947	444280
23	1868.99	241	236	231	23	0.958506	450426
24	1856.74	259	255	245	24	0.945946	480895
25	1919.28	279	276	268	26	0.960573	535480

26	1924.76	302	299	291	29	0.963576	581276
27	1913.94	328	320	309	30	0.942073	627773
28	1895.38	350	343	334	33	0.954286	663382
29	1946.9	376	371	360	36	0.957447	732035
30	1940.34	407	395	387	38	0.95086	789717
31	1948.71	433	429	418	41	0.965358	843790
32	1924.7	470	459	449	44	0.955319	904608
33	1860.47	503	493	477	47	0.94831	935819
34	1442.94	540	528	499	49	0.924074	779187
35	697.764	577	344	317	31	0.549393	402610
36	965.248	375	289	196	19	0.522667	361968
37	1126.58	308	255	196	19	0.636364	346985
38	1410.24	274	216	194	19	0.708029	386405
39	1882.5	235	215	211	21	0.897872	442388
40	2391.77	236	233	230	23	0.974576	564458
41	2414.13	256	255	253	25	0.988281	618018
42	2412.94	280	278	277	27	0.989286	675622
43	1853.65	305	249	245	24	0.803279	565362
44	1571.09	273	215	202	20	0.739927	428909
45	1649.99	235	215	181	18	0.770213	387747
46	1992.95	233	222	217	21	0.93133	464356
47	1882.71	243	242	237	23	0.975309	457499
48	1868.03	265	264	256	25	0.966038	495029

49	1762.63	289	282	277	27	0.958478	509399
50	1351.8	309	300	290	29	0.938511	417707
51	858.162	329	255	244	24	0.741641	282335
52	1010.76	279	239	197	19	0.706093	282002
53	1214.44	258	229	216	21	0.837209	313325
54	1727.69	250	224	219	21	0.876	431922
55	2141.75	245	243	238	23	0.971429	524730
56	2056.4	266	260	251	25	0.943609	547002
57	1430.54	285	278	268	26	0.940351	407705
58	1179.19	304	295	277	27	0.911184	358474
59	1110.47	322	313	294	29	0.913043	357573
60	1032.28	342	331	310	31	0.906433	353041
61	955.299	362	347	319	31	0.881215	345818
62	897.583	378	358	333	33	0.880952	339286
63	860.386	391	362	332	33	0.849105	336411
64	412.804	395	268	227	22	0.574684	163058
65	636.201	290	197	177	17	0.610345	184498
66	1947.74	214	193	188	18	0.878505	416817
67	3208.08	211	211	209	20	0.990521	676905
68	2171	231	230	227	22	0.982684	501501
69	1483.92	252	251	244	24	0.968254	373948
70	1342.45	275	273	263	26	0.956364	369175
71	1214.9	299	291	274	27	0.916388	363254

72	1067.19	318	308	293	29	0.921384	339367
73	960.379	337	309	291	29	0.863501	323648
74	1016.49	338	284	273	27	0.807692	343574
75	1030.03	311	307	292	29	0.938907	320340
76	885.308	336	328	305	30	0.907738	297464
77	675.72	358	315	287	28	0.801676	241908
78	596.493	343	287	211	21	0.61516	204597
79	708.886	308	268	205	20	0.665584	218337
80	756.688	288	229	186	18	0.645833	217926
81	912.69	247	222	204	20	0.825911	225434
82	1308.12	242	233	227	22	0.938017	316565
83	1351.52	255	254	248	24	0.972549	344637
84	1302.82	278	273	266	26	0.956835	362183
85	1035.81	299	257	246	24	0.822742	309707
86	1822.49	281	257	250	25	0.88968	512121
87	1451.1	282	280	270	27	0.957447	409211
88	1229.05	307	302	291	29	0.947883	377319
89	813.652	331	324	298	29	0.900302	269319
90	725.265	353	343	313	31	0.886686	256019
91	675.687	374	358	321	32	0.858289	252707
92	650.202	390	373	344	34	0.882051	253579
93	477.997	407	347	297	29	0.72973	194545
94	507.82	376	324	258	25	0.68617	190940

95	577.093	349	292	238	23	0.681948	201405
96	1166.58	315	266	261	26	0.828571	367471
97	1807.3	292	257	212	21	0.726027	527733
98	1125.54	278	247	203	20	0.730216	312900
99	1104.79	267	248	238	23	0.891386	294979
100	1567.43	271	268	264	26	0.97417	424774

APPENDIX B: RESULTS OF RETAILERS IN A RUN UNDER SUPPLY-DRIVEN ECONOMY

The following data are the results of retailers collected from the experiment discussed in Section 6.1.

Description of columns:

A: Generation sequence

B: Average profit of retailers

C: Number of retailers

D: Number of surviving retailers

E: Number of healthy retailers

F: Number of newborn retailers

G: Healthy rate of retailers

H: Total profit of retailers

Results:

A	B	C	D	E	F	G	H
1	204.41	30	6	2	1	0.0666667	6132.31
2	2674.03	7	5	4	2	0.571429	18718.2

3	2730.01	7	7	5	2	0.714286	19110.1
4	1698.78	9	8	5	2	0.555556	15289
5	1097.79	10	6	5	2	0.5	10977.9
6	2547.51	8	7	5	2	0.625	20380.1
7	1914.24	9	8	5	2	0.555556	17228.2
8	2593.04	10	8	6	3	0.6	25930.4
9	1688.89	11	9	6	3	0.545455	18577.8
10	2310.36	12	11	8	4	0.666667	27724.4
11	2080.96	15	12	9	4	0.6	31214.4
12	2369.71	16	14	10	5	0.625	37915.4
13	2421.3	19	17	12	6	0.631579	46004.6
14	1785.59	23	20	15	7	0.652174	41068.6
15	1885.97	27	23	18	7	0.666667	50921.2
16	1859.93	30	28	19	2	0.633333	55798
17	1765	30	28	18	2	0.6	52950
18	2609.56	30	24	21	6	0.7	78286.9
19	3201.55	30	26	23	4	0.766667	96046.4
20	3504.95	30	30	24	0	0.8	105148
21	3480	30	28	26	2	0.866667	104400
22	4137.96	30	28	27	2	0.9	124139
23	5179.57	30	30	28	0	0.933333	155387
24	5848.36	30	29	28	1	0.933333	175451
25	6627.02	30	29	29	1	0.966667	198811

26	6986.79	30	30	29	0	0.966667	209604
27	8264.14	30	30	30	0	1	247924
28	8753.56	30	30	29	0	0.966667	262607
29	9916.46	30	30	30	0	1	297494
30	11109.3	30	30	30	0	1	333278
31	12200.3	30	30	30	0	1	366008
32	13195.6	30	30	30	0	1	395868
33	13104.9	30	30	30	0	1	393147
34	11993.7	30	30	30	0	1	359810
35	5898.87	30	30	23	0	0.766667	176966
36	4876.13	30	30	20	0	0.666667	146284
37	4669.96	30	24	19	6	0.633333	140099
38	4879.38	30	28	21	2	0.7	146381
39	5351.72	30	28	23	2	0.766667	160552
40	5091.48	30	27	25	3	0.833333	152744
41	6908.19	30	29	29	1	0.966667	207246
42	7917.66	30	30	29	0	0.966667	237530
43	7029.92	30	30	29	0	0.966667	210898
44	4770.2	30	29	22	1	0.733333	143106
45	3904.94	30	30	23	0	0.766667	117148
46	3667.58	30	27	23	3	0.766667	110027
47	3864.23	30	30	25	0	0.833333	115927
48	4240.64	30	27	23	3	0.766667	127219

49	4822.9	30	29	25	1	0.833333	144687
50	3844.76	30	29	22	1	0.733333	115343
51	2793.96	30	27	18	3	0.6	83818.8
52	2711.61	30	26	17	4	0.566667	81348.2
53	3255.19	30	27	20	3	0.666667	97655.8
54	4407.47	30	27	20	3	0.666667	132224
55	5158.13	30	26	20	4	0.666667	154744
56	5745.81	30	28	21	2	0.7	172374
57	4659.15	30	28	21	2	0.7	139774
58	4495.51	30	26	22	4	0.733333	134865
59	4205.34	30	28	21	2	0.7	126160
60	4270.85	30	28	21	2	0.7	128125
61	4111.94	30	26	21	4	0.7	123358
62	4110.71	30	27	21	3	0.7	123321
63	4206.02	30	28	22	2	0.733333	126180
64	2780.71	30	26	16	4	0.533333	83421.4
65	2392.98	30	28	14	2	0.466667	71789.3
66	5158.46	30	27	23	3	0.766667	154754
67	8109.29	30	28	25	2	0.833333	243279
68	6032.08	30	29	23	1	0.766667	180962
69	4826.69	30	28	22	2	0.733333	144801
70	4565.87	30	26	19	4	0.633333	136976
71	4292.83	30	28	20	2	0.666667	128785

72	4867.93	30	25	20	5	0.666667	146038
73	5660.77	30	27	20	3	0.666667	169823
74	5714.22	30	29	21	1	0.7	171427
75	5504.04	30	25	21	5	0.7	165121
76	5873.74	30	28	22	2	0.733333	176212
77	5448.71	30	29	21	1	0.7	163461
78	4204.75	30	25	18	5	0.6	126142
79	4357.42	30	28	18	2	0.6	130723
80	4388.2	30	26	20	4	0.666667	131646
81	4522.18	30	26	19	4	0.633333	135665
82	7019.09	30	28	23	2	0.766667	210573
83	7726.72	30	27	24	3	0.8	231802
84	8465.51	30	27	25	3	0.833333	253965
85	6347.48	30	30	21	0	0.7	190425
86	9038.32	30	28	26	2	0.866667	271150
87	8581.19	30	28	26	2	0.866667	257436
88	8502.64	30	30	29	0	0.966667	255079
89	6750.76	30	30	25	0	0.833333	202523
90	6724.2	30	29	26	1	0.866667	201726
91	6720.56	30	28	25	2	0.833333	201617
92	6620.68	30	30	27	0	0.9	198620
93	4606.49	30	27	19	3	0.633333	138195
94	4442.07	30	30	21	0	0.7	133262

95	4434.2	30	24	20	6	0.666667	133026
96	5488.38	30	28	20	2	0.666667	164652
97	3755.24	30	29	22	1	0.733333	112657
98	3817.34	30	24	17	6	0.566667	114520
99	5301.11	30	29	21	1	0.7	159033
100	9172.85	30	28	26	2	0.866667	275186

APPENDIX C: RESULTS OF TRUCKS IN A RUN UNDER DEMAND-DRIVEN ECONOMY

The following data are the results of trucks collected from the experiment discussed in Section 6.2.

Description of columns:

A: Generation sequence

B: Average profit of trucks

C: Number of trucks

D: Number of surviving trucks

E: Number of healthy trucks

F: Number of newborn trucks

G: Healthy rate of trucks

H: Total profit of trucks

Results:

A	B	C	D	E	F	G	H
1	346.889	400	206	139	13	0.3475	138756
2	68.07	219	154	61	6	0.278539	14907.3

3	180.942	160	124	67	6	0.41875	28950.7
4	424.623	130	98	83	8	0.638462	55200.9
5	833.673	106	98	92	9	0.867925	88369.4
6	1123.12	107	106	100	10	0.934579	120174
7	1023.5	116	109	105	10	0.905172	118726
8	864.109	119	115	97	9	0.815126	102829
9	792.194	124	119	96	9	0.774194	98232
10	1008.62	128	119	104	10	0.8125	129104
11	1592.88	129	126	125	12	0.968992	205481
12	1808.22	138	138	136	13	0.985507	249535
13	1547.07	151	149	140	14	0.927152	233608
14	1825.19	163	162	153	15	0.93865	297505
15	1756.61	177	171	168	16	0.949153	310919
16	1778.07	187	187	181	18	0.967914	332500
17	1410.12	205	203	192	19	0.936585	289074
18	1430.71	222	218	212	21	0.954955	317617
19	1301.72	239	235	221	22	0.924686	311110
20	1322.57	257	254	243	24	0.945525	339899
21	1197.45	278	269	255	25	0.917266	332891
22	1241.17	294	287	277	27	0.942177	364903
23	1173.06	314	306	297	29	0.94586	368342
24	1151.29	335	331	318	31	0.949254	385683
25	1171.81	362	357	341	34	0.941989	424197

26	1124.16	391	385	371	37	0.948849	439546
27	1013.82	422	409	383	38	0.907583	427830
28	983.598	447	440	427	42	0.955257	439668
29	937.422	482	467	441	44	0.914938	451837
30	908.778	511	503	468	46	0.915851	464386
31	848.24	549	517	496	49	0.903461	465684
32	260.5	566	195	112	11	0.19788	147443
33	787.888	206	158	101	10	0.490291	162305
34	951.854	168	124	100	10	0.595238	159912
35	1709.35	134	130	129	12	0.962687	229053
36	1720.57	142	141	139	13	0.978873	244321
37	1718.56	154	152	151	15	0.980519	264658
38	1740.08	167	165	163	16	0.976048	290594
39	1739.5	181	179	177	17	0.977901	314849
40	1602.1	196	196	192	19	0.979592	314011
41	1635.68	215	212	210	21	0.976744	351670
42	1601.41	233	231	228	22	0.978541	373130
43	1559.39	253	250	248	24	0.980237	394526
44	1525.76	274	270	267	26	0.974453	418059
45	1504.92	296	295	291	29	0.983108	445457
46	1430.32	324	319	308	30	0.950617	463424
47	1331.13	349	346	338	33	0.968481	464564
48	1198.98	379	371	361	36	0.952507	454414

49	1125.86	407	401	373	37	0.916462	458225
50	1011.83	438	429	404	40	0.922374	443182
51	954.891	469	455	440	44	0.938166	447844
52	913.754	499	486	461	46	0.923848	455963
53	852.274	532	520	487	48	0.915414	453410
54	811.511	568	553	514	51	0.90493	460938
55	770.924	604	584	552	55	0.913907	465638
56	678.707	639	609	559	55	0.874804	433694
57	400.913	664	454	421	42	0.634036	266206
58	493.317	496	402	237	23	0.477823	244685
59	606.124	425	370	243	24	0.571765	257603
60	675.114	394	287	244	24	0.619289	265995
61	1038.81	311	304	295	29	0.948553	323069
62	994.539	333	331	318	31	0.954955	331181
63	957.501	362	355	343	34	0.947514	346615
64	875.754	389	378	361	36	0.928021	340668
65	862.443	414	405	391	39	0.944444	357051
66	759.231	444	432	412	41	0.927928	337099
67	651.141	473	465	438	43	0.926004	307990
68	497.787	508	410	384	38	0.755906	252876
69	624.707	448	369	284	28	0.633929	279869
70	714.444	397	351	300	30	0.755668	283634
71	761.201	381	331	318	31	0.834646	290018

72	867.526	362	349	344	34	0.950276	314044
73	762.042	383	376	356	35	0.929504	291862
74	573.885	411	402	372	37	0.905109	235867
75	503.839	439	425	396	39	0.90205	221185
76	447.458	464	382	335	33	0.721983	207621
77	517.172	415	363	270	27	0.650602	214626
78	546.911	390	341	278	27	0.712821	213295
79	628.858	368	315	277	27	0.752717	231420
80	680.329	342	311	283	28	0.827485	232672
81	708.012	339	330	315	31	0.929204	240016
82	662.667	361	359	336	33	0.930748	239223
83	539.688	392	283	271	27	0.691327	211558
84	755.023	310	255	195	19	0.629032	234057
85	823.896	274	246	197	19	0.718978	225747
86	1009.85	265	239	234	23	0.883019	267611
87	1191.42	262	260	252	25	0.961832	312153
88	975.667	285	282	273	27	0.957895	278065
89	860.146	309	304	295	29	0.954693	265785
90	804.752	333	330	324	32	0.972973	267982
91	743.495	362	360	343	34	0.947514	269145
92	607.926	394	386	370	37	0.939086	239523
93	491.288	423	417	385	38	0.910165	207815
94	432.39	455	447	416	41	0.914286	196737

95	388.524	488	471	434	43	0.889344	189600
96	354.181	514	503	451	45	0.877432	182049
97	320.334	548	533	475	47	0.866788	175543
98	282.284	580	565	494	49	0.851724	163725
99	264.098	614	589	524	52	0.85342	162156
100	286.811	641	517	431	43	0.672387	183846

APPENDIX D: RESULTS OF RETAILERS IN A RUN UNDER DEMAND-DRIVEN ECONOMY

The following data are the results of retailers collected from the experiment discussed in Section 6.2.

Description of columns:

A: Generation sequence

B: Average profit of retailers

C: Number of retailers

D: Number of surviving retailers

E: Number of healthy retailers

F: Number of newborn retailers

G: Healthy rate of retailers

H: Total profit of retailers

Results:

A	B	C	D	E	F	G	H
1	-48.0627	30	9	3	1	0.1	-1441.88
2	829.685	10	6	6	3	0.6	8296.85

3	240.395	9	8	3	1	0.333333	2163.56
4	990.331	9	8	5	2	0.555556	8912.97
5	1086.81	10	9	7	3	0.7	10868.1
6	1588.21	12	11	5	2	0.416667	19058.6
7	1251.31	13	12	7	3	0.538462	16267
8	1841.31	15	11	8	4	0.533333	27619.7
9	1912.89	15	11	10	5	0.666667	28693.3
10	2309.8	16	15	10	5	0.625	36956.9
11	2866.17	20	19	14	7	0.7	57323.3
12	2986.95	26	22	17	8	0.653846	77660.6
13	2750.65	30	28	21	2	0.7	82519.6
14	2659.71	30	30	22	0	0.733333	79791.4
15	3005.93	30	24	21	6	0.7	90177.9
16	2548.01	30	29	23	1	0.766667	76440.2
17	3458.45	30	30	25	0	0.833333	103754
18	3958.92	30	26	26	4	0.866667	118768
19	4423.7	30	30	29	0	0.966667	132711
20	5062.45	30	30	29	0	0.966667	151874
21	5518.26	30	29	29	1	0.966667	165548
22	5972.5	30	30	30	0	1	179175
23	6314.68	30	30	30	0	1	189440
24	6631.07	30	30	30	0	1	198932
25	7994.04	30	30	30	0	1	239821

26	8467.31	30	30	30	0	1	254019
27	8371.26	30	30	30	0	1	251138
28	8679.86	30	30	30	0	1	260396
29	8928.23	30	30	30	0	1	267847
30	9444.43	30	30	30	0	1	283333
31	10014.2	30	30	30	0	1	300425
32	2942.76	30	30	22	0	0.733333	88282.7
33	2834.09	30	30	19	0	0.633333	85022.8
34	2842.7	30	23	19	7	0.633333	85281
35	4577.35	30	30	24	0	0.8	137321
36	4342.37	30	29	25	1	0.833333	130271
37	4670.37	30	27	26	3	0.866667	140111
38	5288.27	30	30	26	0	0.866667	158648
39	5808.87	30	30	28	0	0.933333	174266
40	5927.76	30	28	27	2	0.9	177833
41	6894.34	30	30	30	0	1	206830
42	7528.21	30	30	30	0	1	225846
43	8158.85	30	30	30	0	1	244765
44	8836.74	30	30	30	0	1	265102
45	9804.44	30	30	30	0	1	294133
46	10662.6	30	30	30	0	1	319877
47	10818.7	30	30	30	0	1	324561
48	10770.1	30	30	30	0	1	323104

49	10778.4	30	30	30	0	1	323353
50	10823.9	30	30	30	0	1	324717
51	11210.2	30	30	30	0	1	336306
52	11586.9	30	30	30	0	1	347608
53	11559.5	30	30	30	0	1	346785
54	11774.3	30	30	30	0	1	353228
55	12002.4	30	30	30	0	1	360073
56	11201.1	30	30	30	0	1	336033
57	7486.48	30	30	29	0	0.966667	224594
58	6354.62	30	30	27	0	0.9	190639
59	6629.09	30	29	28	1	0.933333	198873
60	6843.46	30	29	28	1	0.933333	205304
61	7946.36	30	30	29	0	0.966667	238391
62	8483.25	30	30	29	0	0.966667	254497
63	9063.51	30	29	29	1	0.966667	271905
64	8977.63	30	30	30	0	1	269329
65	9598.26	30	30	30	0	1	287948
66	9505	30	30	30	0	1	285150
67	9190.17	30	30	30	0	1	275705
68	8255.14	30	30	29	0	0.966667	247654
69	8109.46	30	30	28	0	0.933333	243284
70	8209.05	30	30	30	0	1	246271
71	8658.43	30	30	30	0	1	259753

72	9298.43	30	30	30	0	1	278953
73	8461.21	30	30	30	0	1	253836
74	7426.65	30	30	30	0	1	222800
75	7490.98	30	30	30	0	1	224729
76	6893.89	30	30	28	0	0.933333	206817
77	6379.9	30	30	28	0	0.933333	191397
78	6568.05	30	29	29	1	0.966667	197042
79	7060.48	30	30	29	0	0.966667	211815
80	7125.94	30	30	29	0	0.966667	213778
81	8328.7	30	30	30	0	1	249861
82	8455.04	30	30	30	0	1	253651
83	8063.85	30	30	30	0	1	241915
84	8358.04	30	30	27	0	0.9	250741
85	7938.22	30	30	25	0	0.833333	238147
86	9571.37	30	29	28	1	0.933333	287141
87	10787.6	30	30	29	0	0.966667	323629
88	9827.26	30	30	29	0	0.966667	294818
89	9615.93	30	30	30	0	1	288478
90	10097.1	30	30	30	0	1	302914
91	10259.2	30	30	30	0	1	307776
92	9670.89	30	30	30	0	1	290127
93	8495.72	30	30	30	0	1	254872
94	8465.88	30	30	30	0	1	253976

95	8301.12	30	30	29	0	0.966667	249034
96	8244.39	30	30	28	0	0.933333	247332
97	8147.05	30	30	30	0	1	244411
98	8058.54	30	30	29	0	0.966667	241756
99	8023.53	30	30	30	0	1	240706
100	7914.73	30	30	27	0	0.9	237442