CHAPTER II

THEORY AND LITERATURE REVIEW

This chapter considers the supply chain and logistics (SCL), Genetic Algorithm (GA) and linear programming (LP). Finally, the statistical background including factorial designs, the 2^k factorial design, full factorial designs, fractional factorial Designs and analysis of variance (ANOVA) is described.

Supply Chain and Logistics (SCL)

The definitions of supply chain and logistics are found favourite in literature (Beamon, 1998) (Cox & Blckstone, 1998) (Harland, 1996). The distinction between supply chain and logistics is often questioned. The answer to this question however depends on who is addressing this issue. Supply chain consists of all parties involved directly and indirectly in fulfilling a customer request (Chopra & Meindl, 2004).

Logistics is often defined as the art of bringing the right amount of the right product to the right place at the right time and usually refers to supply chain problems (Tilanus, 1997). Logistics system is the planning and coordination of the physical movement aspects of a firm's operations such that a flow of raw materials, parts, and finished goods is achieved in a manner that minimizes total costs for the levels of service desired (Cox & Blackstone, 1998).

The supply chain and logistics need to reduce system-wide costs, in order to meet customer's requirement. Therefore, the objective of supply chain and logistics is the effective supply chain management. Supply chain management is a set of all parties involved, directly or indirectly, to efficiently integrate suppliers, manufacturers, warehouses, and stores, so that merchandise is produced and distributed at the right quantities, to the right locations, and at the right time, to minimize system-wide costs while in order to meet customer's requirement (Simchi-Levi, et al., 2003).

Supply chain management is defined as the global network, used to deliver products and services from raw material to end customers, raw materials are procured from suppliers items are then produced at one or more factories, finished items are shipped to warehouses for intermediate storage, and then shipped to retailers or customers. The supply chain consists of suppliers, manufacturing centers, warehouses, distribution centers, and retail outlets, as well as raw materials, work-in-process inventory, and finished products that flow between the facilities (see Figure 1). Consequently, to reduce cost and improve service levels, effective supply chain strategies must take into account the interactions at the various levels in the supply chain.

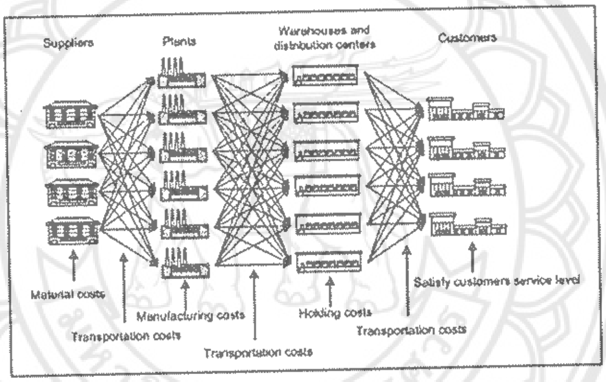

Figure 1 Typical logistics chain network.

The problem of logistics chain network (LCN) is usually concerned with the allocation strategy from suppliers via plants and warehouses/distribution centers to customers with finite capacity constrains in order to meet customer's requirement at minimum cost or maximum profit (Syarif, et al., 2002), (Simchi-Levi, et al., 2003). The LCN problem may therefore be referred to a multiple stages capacitated transportation/allocation problem known to be NP-hard (Gen & Cheng, 1997). The performance of the logistics chain network system may be measured in terms of total transportation costs, which comprise of moving cost of raw material from suppliers to plants, finished goods from plants to distribution centers and then delivering to customers.

Main issue in logistics system is to find the network strategy that can give the least cost of the physical distribution flow as written in the study on multi-stage logistics chain network: a spanning tree-based genetic algorithm approach by Admi Syarif, YoungSu Yun and Mitsuo Gen (2002). In this paper they consider the logistics chain network problem formulated by 0-1 mixed integer linear programming model and design tasks involve the choice of the facilities (plants and distribution centers) to be opened and the distribution network design to satisfy the demand with minimum cost. As the solution method, this paper proposes the spanning tree-based genetic algorithm by using Prüfer number representation. They design the feasibility criteria and develop the repairing procedure for the infeasible Prüfer number, so that it can work for relatively large size problems. The efficacy and the efficiency of this method are shown by comparing its numerical experiment results between those of traditional matrix-based genetic algorithm and professional software package LINDO. Conclusion of this paper, they proposed a st-GA approach to find the best production/distribution design in multi-stage logistics system. They utilize the Prüfer number that is known to be an efficient way to represent various network problems. Even though the structure of the proposed methods is very simple. Experimental results showed that those algorithms can not only give better heuristic solutions in almost all of the time, but also has better performance in the sense of computational time and required memory for computation than those of m-GA. For relatively small size problems, it is proofed that these proposed methods can search the optimal solution in almost all of the time. So they believe this method will be an efficient and robust method to solve this kind of multi-stage logistics chain design problems.

Nozick and Turnquist (2000) present a modeling approach that provides an integrated view of inventory costs, transportation costs, and service levels when making DC location decisions. At first, an approximate inventory cost function is proposed that can be embedded in a fixed-charge facility location model. This allows the decisions on the optimal number and location of DCs to directly tie to inventory cost implications. Second, the fixed-charge location model is extended to incorporate multiple objectives (minimizing cost and maximizing service coverage). The third step shows how this integrated model can be used to explore important trade-offs in the DC location decisions. Conclusions of this paper, has developed an analysis procedure of the location of DCs that integrates facility costs, inventory costs, transportation costs and service responsiveness. That procedure integrates ideas in queuing theory, discrete choice location analysis and multi-objective decision-making. Using this procedure, decision makers can easily understand the service-cost trade-offs that are available, so that optimal location decisions can be reached.

The balanced allocation of customers to multiple distribution centers in the supply chain network: a genetic algorithm approach was written (Zhou, et al., 2002). This paper present a new model based on naïve balanced star spanning forest formulation. This model goes beyond traditional mathematical programming by incorporating a genetic algorithm that is proven to be effective in dealing with the *NP-hard* problem. For the genetic algorithm development, the Prüfer number is represented such a kind of star-spanning forest solutions. Conclusions of this paper, the balanced allocation of customers to distribution centers increases the chances of minimizing stock-outs and late deliveries, while maximizing the order fill rate and utilization rate of distribution centers and developed a new balanced star-spanning forest formulation and a GA to solve the balanced allocation problem which is known to be *NP-hard*.

A genetic algorithm approach to the bi-criteria allocation of customers to warehouses was written (Zhou, et al., 2003). This paper presents a mathematical model and an efficient solution procedure for the bi-criteria allocation problem involving designed to find Pareto optimal solutions for this problem in a short period of time. Conclusion of this paper was presented a novel genetic algorithm approach to BMWAP where the assumption that customers are served by warehouses with equal capacity was relaxed. This paper also took into account two conflicting objectives (transit time versus shipping cost) involving the warehouse allocation problem. The experiments of this paper showed the tradeoff between total transit time and total shipping cost. As expected, total transit time was increased with a reduction in total shipping cost or vice versa. It is intriguing to note that a scenario where warehouses have equal capacity tends to produce solutions with slightly faster transit time and slightly lower shipping costs than the scenario where warehouses have varying capacities. This result indicates that the presence of various sizes of warehouses may force some of he customers to be allocated to warehouses located farther away from their locations, due to insufficient capacities of warehouses near them. This, consequently, increases the total shipping cost. Furthermore, given the successful application of a genetic algorithm to the real-world bi-objective warehouse allocation problem, the proposed algorithm can be modified to obtain non-dominated solutions for warehouse allocation problems with more than two objectives.

Design options for a distribution network

There are six distinct distribution network designs that may be used to move products from factory to customer (Chopra & Meindl, 2004). These are classified as follows:

1. Manufacturer storage with direct shipping
2. Manufacturer storage with direct shipping and in-transit merge
3. Distributor storage with package carrier delivery
4. Distributor storage with last mile delivery
5. Manufacturer/distributor storage with consumer pickup
6. Retail storage with customer pickup

Manufacturer Storage with Direct Shipping

In this option, product is shipped directly from the manufacturer to the end customer, bypassing the retailer (who takes the order and initiates the delivery request). It is also referred to as drop-shipping with product delivered directly from the manufacturer to the customer location. The retailer, if they exist independent of the manufacturer, carries no inventories with all inventories stored at the manufacturer, whereas product is shipped directly from the manufacturer to customers as shown in Figure 2.

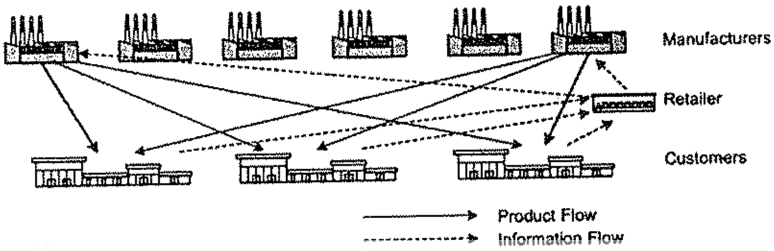

Figure 2 Manufacturer Storage with Direct Shipping (Chopra & Meindl, 2004).

Manufacturer Storage with Direct Shipping and In-Transit Merge

Unlike pure drop-shipping where each product in the order is sent directly from its manufacturer to the end customer, in-transit merge combines pieces of the order coming from different locations so that the customer gets a single delivery. Information and product flows for the in-transit merge network are as shown in Figure 3.

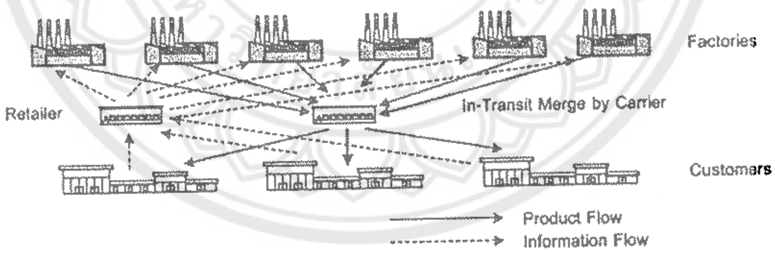

Figure 3 In-Transit Merge Network (Chopra & Meindl, 2004).

Distributor Storage with Carrier Delivery

In this option, inventory is not held by manufacturers at the factories but is held by distributors/retailers in intermediate warehouses and package carriers are used to transport products from the intermediate location to the final customer. Information and product flows when using distributor storage with delivery by a package carrier are shown in Figure 4.

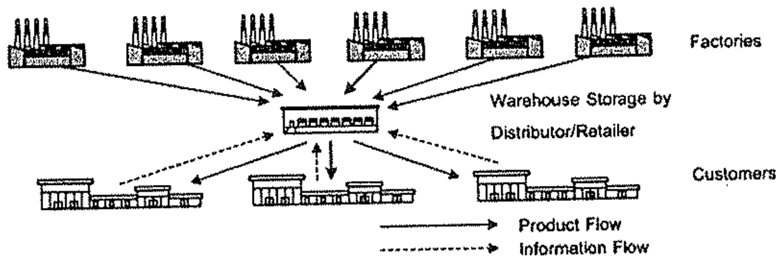

Figure 4 Distributor Storage with Carrier Delivery (Chopra & Meindl, 2004).

Distributor Storage with Last Mile Delivery

Last mile delivery refers to the distributor/retailer delivering the product to the customer's home instead of using a package carrier. Unlike package carrier delivery, last mile delivery requires the distributor warehouse to be much closer to the customer. Given the limited radius that can be served with last mile delivery, more warehouses are required compared to the case when package delivery is used. The warehouse storage with last mile delivery network is as shown in Figure 5.

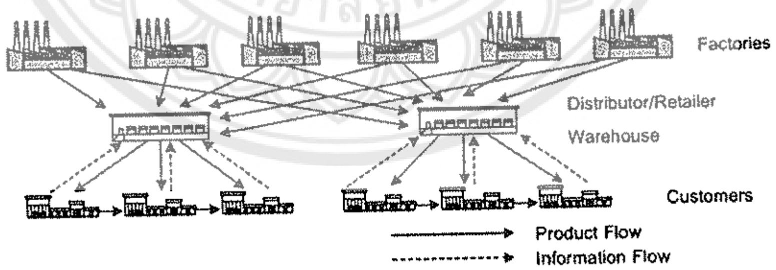

Figure 5 Distributor Storage with Last Mile Delivery (Chopra & Meindl, 2004).

Manufacturer or Distributor Storage with Consumer Pickup

In this option, inventory is stored at the manufacturer or distributor warehouse but customers place their orders online or on the phone and then come to designated pickup points as needed. Its allow customers to pick up online orders at a designated store. The information and product flows in the network as shown in Figure 6.

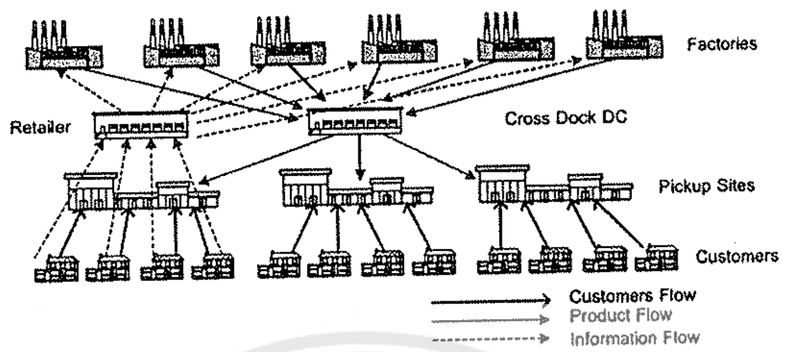

Figure 6 Manufacturers or Distributor Warehouse Storage with Consumer Pickup (Chopra & Meindl, 2004).

Retail Storage with Customer Pickup

In this option, inventory is stored locally at retail stores. Customers walk into the retail store or place an order online or on the phone and pick it up at the retail store. The companies that offer multiple options of order placement always uses part of the facility as a grocery store and part of the facility as an online fulfillment center (Chopra & Meindl, 2004).

Selecting a Distribution Network Design

A network designer needs to consider product characteristics as well as network requirements when deciding on the appropriate delivery network. The various networks considered earlier have different strengths and weaknesses. In Table 1, the various delivery networks are ranked relative to each other along different performance dimensions. A ranking of 1 indicates the best performance along a given dimension; as the relative performance worsens, the ranking gets higher (Chopra & Meindl, 2004).

Table 1 Comparative Performance of Delivery Network Designs (Chopra & Meindl, 2004).

	Retail Storage with Customer Pickup	Manufacturer Storage with Direct Shipping	Manufacturer Storage with In-Transit Merge	Distributor Storage with Package Carrier Delivery	Distributor storage with last mile delivery	Manufacturer storage with pickup
Response Time	1	4	4	3	2	4
Product Variety	4	1	1	2	3	1
Product Availability	4	1	1	2	3	1
Customer Experience	5	4	3	2	1	5
Order Visibility	1	5	4	3	2	6
Returnability	1	5	5	4	3	2
Inventory	4	1	1	2	3	1
Transportation	1	4	3	2	5	1
Facility & Handling	6	1	2	3	4	5
Information	1	4	4	3	2	5

Only niche companies will end up using a single distribution network. Most companies are best served by a combination of delivery networks. The combination used will depend on product characteristics as well as the strategic position that the firm is targeting. The suitability of different delivery designs (from a supply chain perspective) in various situations is shown in Table 2.

Table 2 Performance of Delivery Networks for Different Product/Customer Characteristics (Chopra & Meindl, 2004).

	Retail Storage With Customer Pickup	Manufacturer Storage with Direct Shipping	Manufacturer Storage with In-Transit Merge	Distributor Storage with Package Carrier Delivery	Distributor Storage with last mile delivery	Manufacturer storage with pickup
High demand product	+2	-2	-1	0	+1	-1
Medium demand product	+1	-1	0	+1	0	0
Low demand product	-1	+1	0	+1	-1	+1
Very low demand product	-2	+2	+1	0	-2	+1
Many product sources	+1	-1	-1	+2	+1	0
High product value	-1	+2	+1	+1	0	-2
Quick desired response	+2	-2	-2	-1	+1	-2
High product variety	-1	+2	0	+1	0	+2
Low customer effort	-2	+1	+2	+2	+2	-1

+2: Very suitable; +1: Somewhat suitable; 0: Neutral; -1: Somewhat unsuitable; -2: Very unsuitable.

Genetic Algorithm (GA)

Genetic algorithms (GA) are stochastic search techniques based upon the mechanics of natural selection and natural genetics (Gen & Gheng, 1997) (Goldberg, 1989). The basic idea came from an analogy with biological evolution, in which the fitness of individual determines its ability to survive and reproduce. To compare the advantage of GA with other conventional methods, GA iteratively performs a multiple directional search by maintaining a population of potential solutions whilst single directional search is normally used in other conventional methods (Gen & Gheng, 1997). GA has therefore been widely applied to solve production and operation management (POM) problems. However, transportation problem in supply chain have not been received much attention in GA-applied research (Aytug, et al., 2003).

The simple GA mechanism starts by encoding the problem to produce a list of genes. The genes are represented by either numeric (binary or real), or alphanumeric characters. The genes are randomly combined to produce a population of chromosome, each of which represents a candidate solution. Genetic operations including crossover and mutation are next performed on chromosomes, which are randomly selected from the population as parents, for producing offspring. The fitness function is then used to measure the chromosomes' fitness value from which the probability of their survival is determined. The most famous selection mechanism called roulette wheel is performed to maintain number of chromosomes for next generation. The GA process is repeated until a termination condition is satisfied. See Figure 7 for the general structure of genetic algorithms.

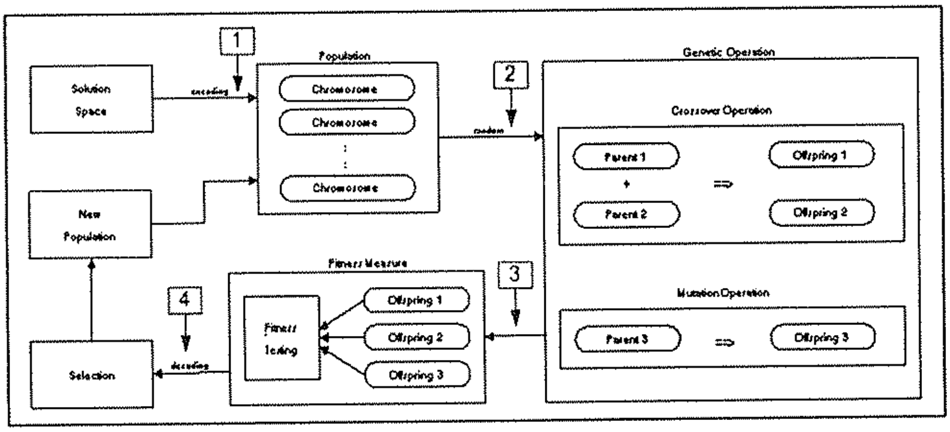

Figure 7 The general structure of genetic algorithms (Pongcharoen, 2001).

The general structure of genetic algorithms included procedures for gene encoding, chromosome initialization, genetic operations (crossover and mutation), fitness evaluation and chromosome selection (Pongcharoen, 2001).

1. Gene Encoding

The first step of genetic algorithms is the gene encoding. The gene encoding start with an initial set of random solutions called population. Each individual in the population is called a chromosome, representing a solution to the problem at hand. A chromosome is a string of symbols; it is usually, but not necessarily, a binary bit string (Gen & Cheng, 1997). A chromosome should in some way contain information about solution that it represents. The most used way of encoding is a binary string. A chromosome can be show in Figure 8. In some way of encoding is permutation encoding, every chromosome is a string of numbers that represent a position in a sequence (see Figure 9).

Chromosome 1	1	1	0	1	1	0	0	1
Chromosome 2	1	0	0	1	1	1	1	0

Figure 8 Example of chromosomes with binary string for eight genes.

Chromosome 1	1	2	3	4	5	6	7	8
Chromosome 2	8	7	6	5	4	3	2	1

Figure 9 Example of chromosomes with permutations for eight genes.

2. **Genetic Operations**

Genetic operations consist of two processes, crossover and mutation.

2.1 Crossover

Crossover is the main genetic operator. It operates on two chromosomes at a time and generates offspring by combining both chromosomes, features. A simple way to achieve crossover would be to choose a random cut-point and generate the offspring by combining the segment of one parent to the left of the cut-point with the segment of the other parent to the right of the cut-point (Gen & Cheng, 1997). Crossover can be illustrated in Figure 10.

Chromosome 1	1	1	0	1	1	0	0	1	→	Offspring 1	1	1	0	1	1	1	1	0
Chromosome 2	1	0	0	1	1	1	1	0	→	Offspring 2	1	0	0	1	1	0	0	1

Figure 10 Crossover Operation for chromosomes with binary string.

There are other ways how to make crossover, for example we can choose more crossover points. Crossover can be quite complicated and depends mainly on the encoding of chromosomes. Specific crossover made for a specific problem can improve performance of the genetic algorithm.

The crossover rate (denoted by %C) is defined as the ratio of the number of offspring produced in each generation to the population size (usually denoted by *pop_size*). This ratio controls the expected number %C x *pop_size* of chromosomes to undergo the crossover operation (Gen & Cheng, 1997). If crossover probability is

100%, then all offspring are made by crossover. If it is 0%, whole new generation is made from exact copies of chromosomes from old population.

2.2 Mutation

Mutation is a background operator which produces spontaneous random changes in various chromosomes. A simple way to achieve mutation would be to alter one or more genes (Gen & Cheng, 1997). After a crossover is performed, mutation takes place. Mutation is intended to prevent falling of all solutions in the population into a local optimum of the solved problem. Mutation operation randomly changes the offspring resulted from crossover. In case of binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1. Mutation can be then illustrated in Figure 11.

Offspring 1	1	1	0	1	1	1	1	0	→	Mutated offspring 1	1	0	0	1	1	1	1	0
Offspring 2	1	0	0	1	1	0	0	1	→	Mutated offspring 2	1	0	0	1	1	1	0	1

Figure 11 Mutation Operation for chromosomes with binary string.

The mutation rate (denoted by %M) is defined as the percentage of the total number of genes in the population. The mutation rate controls the rate at which new genes are introduced into the population for trial (Gen & Cheng, 1997). If mutation probability is 100%, whole chromosome is changed, if it is 0%, nothing is changed.

3. Fitness Evaluation

Fitness Evaluation is usually applied to measure the performance (fitness value) of a candidate solution (individual) by determining an objective (fitness) function. The objective function is used to provide a measure of how individuals have performed in the problem domain. In the case of a minimization problem, the most fit individuals will have the lowest numerical value of the associated objective function. This raw measure of fitness is usually only used as an intermediated stage in determining the relative performance of individuals in a GA. Another function, the fitness function, is normally used to transform the objective function value into a measure of relative fitness thus:

$$F(x) = g(f(x))$$

Where f is the objective function, g transforms the value of the objective function to a non-negative number and F is the resulting relative fitness. This mapping is always necessary when the objective function is to be minimized as the lower objective function values correspond to fitter individuals. In many cases, the fitness function value corresponds to the number of offspring that an individual can expect to produce in the next generation. A commonly used transformation is that of proportional fitness assignment (Chipperfield, et al., n.d.).

4. Chromosome Selection

Selection is the process of determining the number of times, or trials, a particular individual is chosen for reproduction and, thus, the number of offspring that an individual will produce (Chipperfield, et al., n.d.). The selection of individuals can be viewed as two separate processes, determination of the number of trails an individual can expect to receive, and conversion of the expected number of trials into a discrete number of offspring. There are many methods in selecting the best chromosomes. Examples are roulette wheel selection (RWS), stochastic universal sampling (SUS), Boltzman selection, tournament selection, rank selection, steady state selection and some others. For this work, the chromosome selection is RWS and SUS.

4.1 Roulette Wheel Selection

The basic roulette wheel selection method is stochastic sampling with replacement (SSR). The segment size and selection probability remain the same throughout the selection phase and individuals are selected according to the procedure outlined above. SSR gives zero bias but a potentially unlimited spread. Any individual with a segment size > 0 could entirely fill the next population.

The size of each individual interval corresponds to the fitness value of the associated individual. For example, in Fig. 12 the circumference of the roulette wheel is the sum of all six individual's fitness values. Individual 5 is the most fit individual and occupies the largest interval, whereas individuals 6 and 4 are the least fit and have correspondingly smaller intervals within the roulette wheel. To select an individual, a random number is generated in the interval [0, Sum] and the individual whose segment spans the random number is selected. This process is repeated until the desired number of individuals have been selected (Chipperfield, et al., n.d.).

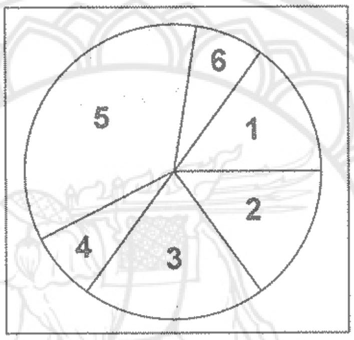

Figure 12 Roulette Wheel Selection (Chipperfield, et al., n.d.).

4.2 Stochastic Universal Sampling

Stochastic universal sampling (SUS) is a single-phase sampling algorithm with minimum spread and zero bias. Instead of the single selection pointer employed in roulette wheel methods, SUS uses *N* equally spaced pointers, where *N* is the number of selections required. The population is shuffled randomly and a single random number is the range [0 *sum/N*] is generated, *ptr*. The *N* individuals are then chosen by generating the N pointers space by 1, [*ptr*, *ptr+1*, ..., *ptr+N-1*], and selecting the individuals whose fitnesses span the positions of the pointers. An individual is thus guaranteed to be selected a minimum of $\lfloor et(i) \rfloor$ times and no more than $\lceil et(i) \rceil$, thus achieving minimum spread. In addition, as individuals are selected entirely on their position in the population, SUS has zero bias (Chipperfield, et al., n.d.).

Linear programming (LP)

Linear programming (LP) is a tool for solving optimization problems. In 1947, George Dantzig developed an efficient method, the simplex algorithm, for solving linear programming problems (also called LP). Since the development of the simplex algorithm, LP has been used to solve optimization problems in industries as diverse as banking, education, forestry, petroleum, and trucking (Winston, 1991).

Before formally defining a linear programming problem, we define the concepts of linear function and linear inequality.

A function $f(x_1, x_2, ..., x_n)$ of $x_1, x_2, ... x_n$ is a linear function if and only if for some set of constants $c_1, c_2, ..., c_n, f(x_1, x_2, ... x_n) = c_1x_1 + c_2x_2 + ... + c_nx_n$. For example, $f(x_1, x_2) = 2x_1 + x_2$ is a linear function of x_1 and x_2, but $f(x_1, x_2) = x_1^2x_2$ is not a linear function of x_1 and x_2.

For any linear function $f(x_1, x_2, ..., x_{n)}$ and any number b, the inequalities $f(x_1, x_2, ..., x_{n)} \leq b$ and $f(x_1, x_2, ..., x_{n)} \geq b$ are linear inequalities. Thus, $2x_1 + 3x_2 \leq 3$ and $2x_1 + 3x_2 \geq 3$ are linear inequalities, but $x_1^2x_2 \geq 3$ is not a linear inequality.

The optimization problem in linear programming following:

1. Attempt to maximize (or minimize) a linear function of the decision variables. The function that is to be maximized or minimized is called the objective function.

2. The values of the decision variables must satisfy a set of constraints. Each constraint must be a linear equation or linear inequality.

3. A sign restriction is associated with each variable. For any variable x_i, the sign restriction specifies either that x_i must be nonnegative $(x_i \geq 0)$.

The simplex method is an algebraic procedure, where each iteration involves solving a system of equations. The simplex algorithm can be used to solve LPs in which the goal is to maximize or minimize objective functions.

The simplex algorithm proceeds as follows:

Step 1 Convert the LP to standard form

Step 2 Obtain a basic feasible solution (if possible) from the standard form.

Step 3 Determine whether the current basic feasible solution (bfs) is optimal.

Step 4 If the current bfs is not optimal; determine which non-basic variable should become a basic variable and which basic variable should become a non-basic variable in order to find a new bfs with a better objective function value.

Step 5 Use ero's to find the new bfs with the better objective function value and go to Step 3.

The transportation model is basically a linear program that can be solved by the regular simplex method. However, its special structure allows the development of a solution procedure, called the transportation technique that is computationally more efficient. The basic assumption of the model is that the transportation cost on a given route is directly proportional to the number of units transported. The definition of "unit of transportation" will vary depending on the "commodity" transported.

Figure 13 depicts the transportation model as a network with m sources and n destinations. A source or a destination is represented by a node. The arc joining a source and a destination represents the route through which the commodity is transported. The amount of supply at source i is a_i and the demand at destination j is b_j. The unit transportation cost between source i and destination j is c_{ij}

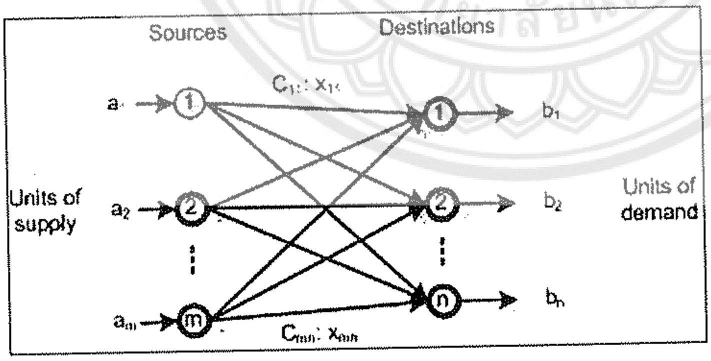

Figure 13 The transportation model as a network.

Statistical Background

In engineering field, every practical engineering design must take account of the effects of the variation inherent in parameters, environments, and processes (Patrick, 1991). For all of experimental designs in this thesis were brought some courses with the design and analysis of experiments form textbook of Montgomery (1997). These courses are factorial designs, the 2^k factorial design, two-level fractional factorial design and analysis of variance (ANOVA).

1. Factorial designs

Montgomery (1997) argued "factorial designs have several advantages. They are more efficient than one factor at a time experiments. Furthermore, a factorial design is often necessary when interactions may be present to avoid misleading conclusions. Finally, factorial experiments allow the effects of a factor to be estimated at several levels of order factors, yielding conclusions that are valid over a range of experimental conditions" (Pongcharoen, 2001).

2. The 2^k factorial design

The most important of these special cases is that of k factors, each at only two levels. These levels may be qualitative, such as two machines, two operators, the "high" and "low" levels of a factor, or perhaps the presence and absence of a factor. A complete replicate of such a design require $2 \times 2 \times \ldots \times 2 = 2^k$ factorial design.

3. Full factorial designs

Pongcharoen (2001) described a full factorial experiment is a set of experimental runs, which are performed at all combinations of factor levels. For example, if there are k factors, each of which has 2 levels, a full factorial experimental design would require 2^k runs.

As the number of factors and levels in a full factorial design rises the number of runs increases rapidly. For example, a design with five factors, three at two levels and remaining two factors at eight levels, would result in 2x2x2x2x8x8 = 512 runs per replication of the experiment.

The effect of a factor is defined as the change in response produced by a change in the level of the factor. It is called a main effect because it refers to the primary factors in the study. In some experiments, the difference in response between the levels of one factor is not the same at all levels of the other factors. This is an interaction effect between the factors (Montgomery & Runger, 1999).

4. **Fractional Factorial Designs**

The fractional factorial designs were described by Pongcharoen (2001). A major use of fractional factorials is in screening experiments, in which many factors are considered, with the purpose of identifying those factors (if any) that have large effects (Montgomery & Runger, 1999). The successful use of fractional factorial experiments is based upon three ideas (Montgomery, 1997). The first is the scarcity of effects principle. When there are several variables, the system or process is likely to be driven primarily by some of the main effects and low order interactions. The second is the projective property. Fractional factorial designs can be projected into stronger (larger) designs in the subset of significant factors. Finally, through sequential experimentation it is possible to combine the runs of two or more fractional factorials to assemble sequentially a larger design to estimate the factor effects and interactions of interest.

Sequential experimentation is particularly useful in computational experiments because many of the issues relating to physical experimentation, such as changes in experimental conditions over time, or background trends or cycles, do not apply. A strategy of running only those experiments that are needed to determine that suspected factor effects are significant can be safely adopted, leading to great savings in time. Box and Liu (1999) described this approach.

A half fraction of a 2^k experiment is called a 2^{k-1} design, similarly a quarter faction would be The 2^{k-2} , A design is of resolution K if no p-factor effect is aliased with another effect containing less than K-p factors where the design is denoted The 2^{K-p} .

5. **Analysis of Variance (ANOVA)**

This topic was described by Pongcharoen (2001). The Analysis of Variance (ANOVA) is a commonly used approach for analyzing the results from factorial experiments. In general, a two factor factorial experiment is obtained as shown in Table 3

Where y_{ijk} is the observed value (response) obtained by using factor A at the i^{th} level (I = 1, 2, ..., a) and factor B at the j^{th} level (j = 1, 2, ..., b) for the k^{th} replicate (k = 1, 2, ..., n).

Table 3 Observed response arrangement for Two-Factor Factorial design (Montgomery, 1997)

		Factor B			
		1	2	...	b
Factor A	1	$Y_{111}, Y_{112}, \ldots, Y_{11n}$	$Y_{121}, Y_{122}, \ldots, Y_{12n}$	...	$Y_{1b1}, Y_{1b2}, \ldots, Y_{1bn}$
	2	$Y_{211}, Y_{212}, \ldots, Y_{21n}$	$Y_{221}, Y_{222}, \ldots, Y_{22n}$	...	$Y_{2b1}, Y_{2b2}, \ldots, Y_{2bn}$
	⋮	⋮	⋮	⋮	⋮
	a	$Y_{a11}, Y_{a12}, \ldots, Y_{a1n}$	$Y_{a21}, Y_{a22}, \ldots, Y_{a2n}$	...	$Y_{ab1}, Y_{ab2}, \ldots, Y_{abn}$

The purpose of the ANOVA test is to establish whether a factor has a statistically significant effect on the variable being measured. ANOVA partitions the total variation within the results into its component parts, that is the variability due to each factor or interaction of interest and background uncertainty or error. There are a number of assumption behind ANOVA. Firstly, the results are independent of one another, that is, the result of one trial is not affected by another. Secondly, that differences in repeat trial would follow a normal distribution and finally that the error is normally distributed and is approximately equal over the whole experimental region (Kvanli et al., 1995). In general, the ANOVA table contains source of variation, sum of squares (SS), degrees of freedom (DF), mean squares (MS) and F value as summarized in Table 4.

Table 4 General ANOVA table for the Two-Factor Factorial design (Kvanli et al., 1995).

Source of Variation	Sum of Square	Degree of Freedom	Mean Square	F Value
A treatments	$SS_A = \frac{1}{bn}\sum_{i=1}^{a} y_{i..}^2 - \bar{y}^2$	a-1	$MS_A = \frac{SS_A}{a-1}$	$\frac{MS_A}{MS_E}$
B treatments	$SS_B = \frac{1}{an}\sum_{j=1}^{b} y_{.j.}^2 - \bar{y}^2$	b-1	$MS_B = \frac{SS_B}{a-1}$	$\frac{MS_B}{MS_E}$
Interaction	$SS_{AB} = \frac{1}{n}\sum_{i=1}^{a}\sum_{j=1}^{b} y_{ij.}^2 - \bar{y}^2 - SS_A - SS_B$	(a-1) (b-1)	$MS_{AB} = \frac{SS_{AB}}{(a-1)(b-1)}$	$\frac{MS_{AB}}{MS_E}$
Error	$SS_E = SS_T - SS_A - SS_B - SS_{AB}$	Ab(n-1)	$MS_E = \frac{SS_E}{ab(n-1)}$	
Total	$SS_T = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n} y_{ijk}^2 - \bar{y}^2$	Abn-1		

The F test is used for comparing population variances. The F value is the ration of the mean square of the factor divided by the mean square of the error. It is therefore a ratio of two independent estimates of the population variance. The F value indicates the p value, which is the probability that a good model is falsely rejected. The p value is compared with a pre-specified significance level (α). It would lead to rejection of the null hypothesis (that the variances are the same) if p > α for example if a 95% confidence limit is used, rejection would occur if p > 0.05. The ANOVA can be conveniently performed using a statistical analysis packages such as Minitab and SPSS.