EVOLUTIONARY TECHNIQUE FOR LOGISTICS NETWORK DESIGN: STATE-OF-THE-ART SURVEY

Mitsuo Gen
Graduate School of Information, Production & Systems, Waseda University, Japan.
gen@waseda.jp

ABSTRACT

The use of evolutionary techniques in the logistics networks design has been growing the last decades due to the fact that the logistics networks design problem is an NP hard problem. This paper examines recent developments in the field of evolutionary optimization for logistics. A number of papers in various areas are highlighted that give good points of evolutionary techniques. A wide range of strategies to approach the problem is covered as follows: first, we apply the hybrid Genetic Algorithm (hGA) approach for solving Fixed Charge Transportation Problem (fcTP). We have done several numerical experiments and compared the results with those of a simple GA. The proposed approach is more effective in larger size than benchmark test problems. Second, we give the recent GA approach for solving Multistage Logistic Network Problems. Third, we introduce Vehicle Routing Problem (VRP) and variants of VRP. We apply the priority-based Genetic Algorithm (pGA) approach for solving Multi-depot vehicle routing problem with time windows (mdVRP-tw). Fourth, we discuss the distribution centre location problem of a distribution system which consists of customers and a number of distribution centres to be located. We adopt a hybrid genetic algorithm (hGA) method to find the global or near global optimal solution for the location-allocation problem. Fifth, as a case study model, practical logistics applications to find the optimal routing will be introduced. Last, we model an automated guided vehicles (AGV) system by using network structure. This network model of an AGV dispatching system has simplex decision variables; considering most of the AGV problem's constraints. Furthermore, we apply an evolutionary approach for solving this problem with minimizing the time required to complete all jobs (i.e., makespan). The aims of this paper are to illustrate state-of-the-art survey in the evolutionary technique for logistics network design.

Key Words: *Evolutionary technique, Genetic Algorithm, Fuzzy Logic Controller, Logistics Network Design, Transportation Problem, Multistage Logistic Network, Vehicle Routing Problem, Location-Allocation Problem, Automated Guided Vehicles.*

1. INTRODUCTION

Logistics optimization is currently the biggest opportunity for most companies to significantly reduce costs in the supply chain. Companies have made tremendous strides in automating transaction processing and data capture related to logistics operations in the last few decades. While these innovations have reduced cost in their own right by reducing manual effort, their greatest impact is yet to come, as they pave the way for optimizing logistics decisions with computer-based technology. Logistics optimization is neither easy nor cheap, but for most logistics operations there is an opportunity to reduce cost by making optimized decisions.

A large number of combinatorial problems are associated with logistics optimization. Most of them are NP complete, i.e. there is no polynomial-time algorithm that can possibly solve them, unless it is proved that P = NP (Garey & Johnson, 1979). Heuristic methods are normally employed for the solution of these problems. A growing number of researchers have adopted the use of meta-heuristic techniques ("smart heuristics") for large combinatorial problems. Evolutionary techniques are meta-heuristics that are able to search large regions of the solution's space without being trapped in local optima.

The aim of this paper is to illustrate recent developments in the field of evolutionary techniques for logistics optimization. A wide range of optimization problems are considered, from the basic transportation models to assembly multistage logistic networks and reverse logistics networks.

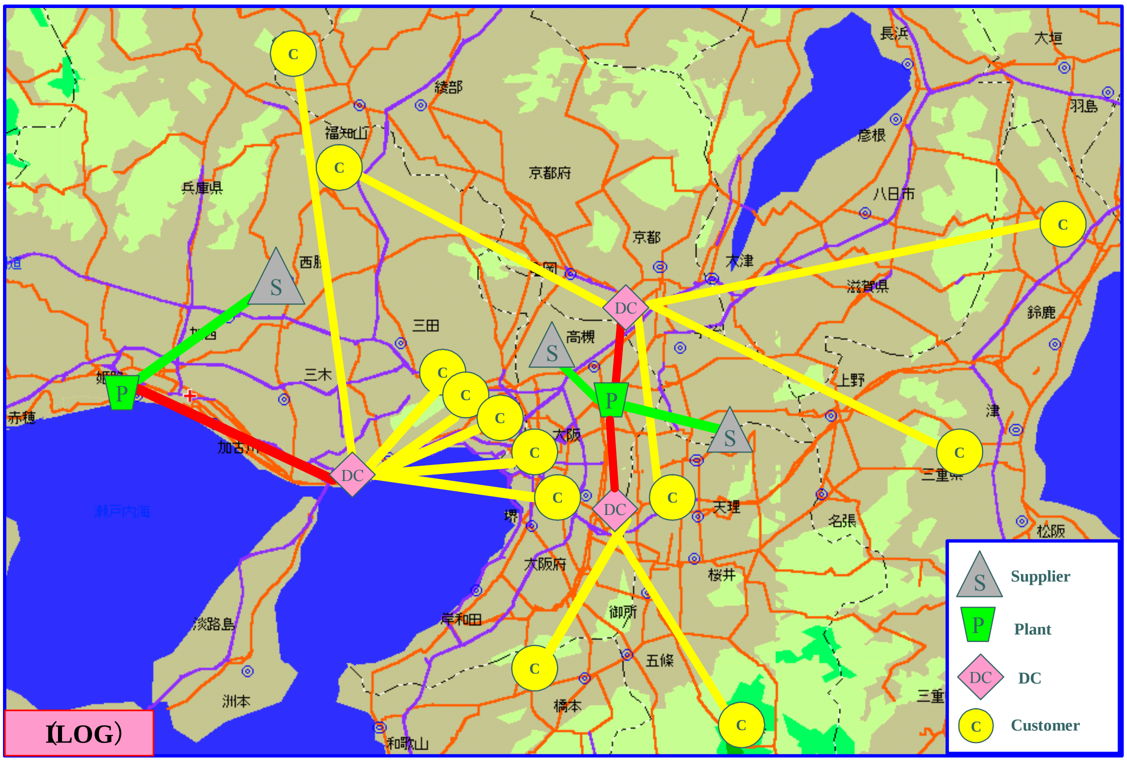


Figure 1. The illustration of the logistics system

Basic Transportation Models: Distribution occurs between every pair of stages in the supply chain. Raw materials and components are moved from suppliers to manufacturers. Distribution is a key driver for the overall profitability of a firm because it directly impacts both the supply chain cost and the customer experience (Chopra & Meindl, 2004). In real life, transportation has the following applications: how to find the reasonable assignment strategy to satisfy the source and destination requirement without shipping goods from any pairs of prohibited sources simultaneously to the same destination so that the total cost can be minimized.

Multistage Logistic Network: In many logistic environments managers must make decisions such as (1) location for factories/ warehouses/ distribution centres (DC), (2) allocation of customers to each service area, and (3) transportation plans connecting customers, raw materials, plants, warehouses and channel members. These decisions are important in the sense that they greatly affect the level of service for customers and the total logistic system cost. For these applications, the transportation problem can be extended to have decision in multistage (Tragantalerngsak *et al.*, 1997; Syarif, Yun & Gen, 2002).

Vehicle Routing Problem (VRP) Models: VRP is a generic name given to a whole class of problems in which a set of routes for a fleet of vehicles based at one or several depots must be determined for a number of geographically dispersed cities or customers. The objective of the

VRP is to deliver a set of customers with known demands on minimum-cost vehicle routes with minimum number of vehicles originating and terminating at a depot (Vignaux & Michalewicz, 1997).

Location-Allocation Models: Location-allocation problems concern the optimal number and location of DCs needed to provide some service to a set of customers. The optimal solution must balance two types of costs the cost of establishing a DC at a particular location and the total cost of providing service to each of the customers from one of the opened DCs. In its simplest form, if each opened DC can serve only a limited number of customers the problem is called capacitated.

Practical Logistics Application: We consider optimal routing problem among 6 DCs, it mainly aims at cost reduction by optimal routing for DC from supplier. To find the optimal routing in the logistics network, we presented genetic algorithm approach. Compared with the conventional delivery model, a result of about a 4.8% cut in logistics cost was obtained.

Automated guided vehicles (AGVs) Dispatching: It is the state–of–the–art, and is often used to facilitate automatic storage and retrieval systems (AS/RS). In this paper, we focus on the dispatching of AGVs in a flexible manufacturing system (FMS). A FMS environment requires a flexible and adaptable material handling system. An AGV system is modelled by using a network structure, and effective evolutionary approach is proposed for solving a kind of AGV problem in which the aim is to minimize time required to complete all jobs (i.e. makespan). Numerical analyses for case studies show the effectiveness of the proposed approach.

The rest of the paper is organized as follows. Section 2 introduces evolutionary technique examined, and Section 3 examines a recent hybrid Genetic Algorithm (hGA) approach for solving Fixed Charge Transportation Problem (fcTP). Section 4 gives the several resent GA approach for solving Multistage Logistic Network Problems. The priority-based Genetic Algorithm (pGA) for solving Vehicle Routing Problem (VRP) and variants of VRP is applied in Section 5. In Section 6, a distribution centre location problem of distribution system which consists of customers and a number of distribution centres to be located is discussed. In Section 7, practical logistics applications to find the optimal routing will be introduced. In Section 8, an evolutionary approach for solving AGV dispatching problems with minimizing time required to complete all jobs (i.e. makespan) is applied. Section 9 draws the conclusion of this paper.

2. EVOLUTIONARY TECHNIQUE

Evolutionary technique is a keyword in Information Technologies, and refers to a synthesis of methodologies from *Neural Networks* (NNs), *Genetic Algorithms* (GAs) and other *Evolutionary Algorithms* (EAs). In the last decade, these methodologies have jointly provided valuable control tools for systems presenting strong combinatorial optimization problems. By contrast, in most cases plants cannot be handled by traditional control strategies.

2.1. Genetic Algorithms

The general form of GAs was described by Goldberg (Goldberg, 1989). GA is one of the stochastic search algorithms based on the mechanism of natural selection and natural genetics. GAs, differing from conventional algorithms, starts with an initial set of random solutions called population $P(t)$. Each individual in the population is called an individuals (or chromosome), representing a potential solution to the problem. The individuals evolve through successive iterations, called generations.

During each generation, the individuals are evaluated by using some measures of fitness. To create the next generation, new chromosomes called offspring $C(t)$ are formed by either merging two individuals from current generation using crossover operator and/or modifying an individual using mutation operator. A new generation is formed by the selection of good individuals according to their fitness values. After several generations, the algorithm converges to the best individual, which hopefully represents the optimal solution or near-optimal solution for the problem. Figure 3.1 shows a general structure of a GA. In general, a GA has five basic components, as summarized by Michalewicz (Michalewicz, 1996):

(1) A genetic representation of solutions to the problem.
(2) A way to create an initial set of potential solutions.
(3) An evaluation function rating solutions in terms of their fitness.
(4) Genetic operators that alter the genetic composition of offspring (crossover, mutation, selection, etc.).
(5) Values for the parameters of genetic algorithms (population size, probabilities of genetic operators, etc.).

Let $P(t)$ and $C(t)$ be parents and offspring in current generation t, the general structure of GA is described as follows (Gen & Cheng, 1997).

```
procedure: Standard GA
input: GA parameters
output: best solution
begin
    t ← 0;                                       // t: generation number
    initialize P(t) by encoding routine;         // P(t): population of chromosomes
    fitness eval(P) by decoding routine;
    while (not termination condition) do
        crossover P(t) to yield C(t);            // C(t): offspring
        mutation P(t) to yield C(t);
        fitness eval(C) by decoding routine;
        select P(t+1) from P(t) and C(t);
        t ← t+1;
    end
    output  best solution;
end
```

GAs have received considerable attention regarding their potential as a novel optimization technique. There are three major advantages when applying GAs to optimization problems:

(1) *Adaptability*: GA does not have much mathematical requirements about the optimization problems. Due to the evolutionary nature, GAs will search for solutions without regard to the specific inner workings of the problem. GAs can handle any kind of objective functions and any kind of constraints, i.e. linear or nonlinear, defined on discrete, continuous or mixed search spaces.

(2) *Robustness*: The use of evolution operators makes GAs very effective in performing global search (in probability), while most of conventional heuristics usually perform local search. It has been proved by many studies that GA is more efficient and more robust in locating optimal solution and reducing computational effort than other conventional heuristics.

(3) *Flexibility*: GA provides us a great flexibility to hybridize with domain-dependent heuristics to make an efficient implementation for a specific problem.

2.2. Auto-tuning of GA Strategy Parameter by Fuzzy Logic Controller

Generally, the behaviour of GAs depend on many uncertain factors, and only incomplete knowledge and imprecise information for assigning several parameters are available for identification of the relationship between the strategy parameters and the behaviour of GAs. Fuzzy logic controller (FLC) provides an algorithm that can convert linguistic control strategy based on expert knowledge into an automatic control strategy. In particular, FLC appears very useful when the processes are too complex for analysis using conventional techniques or when the available sources of information are interpreted qualitatively, inexactly, or are ambiguous. Therefore, it is acceptable for FLC to adjust strategy parameters of GAs dynamically.

Extending the fuzzy logic technique to dynamic control of the strategy, GA parameters were first attempted by (Xu & Vukovich, 1994; Lee & Takagi 1995; Zeng & Rabenasolo, 1997). The main idea is to use a FLC to compute new strategy parameter values of the GA with any combination of the performance measures (changes in the average fitness of the population) and current parameters as the inputs to the controller. According to Lee and Takagi (1985), a FLC is comprised of four principal components:

(1) A knowledge base
(2) A fuzzification interface
(3) An inference system
(4) A defuzzification interface.

The experts' knowledge is stored in the knowledge base in the form of linguistic control rules. The fuzzification interface is used to transform crisp data into fuzzy data. The inference system, the heart of the controller, provides approximate reasoning based on the knowledge base. The defuzzification interface translates fuzzy control action to nonfuzzy control action. The generic structure of a FLC is shown in Figure 2.

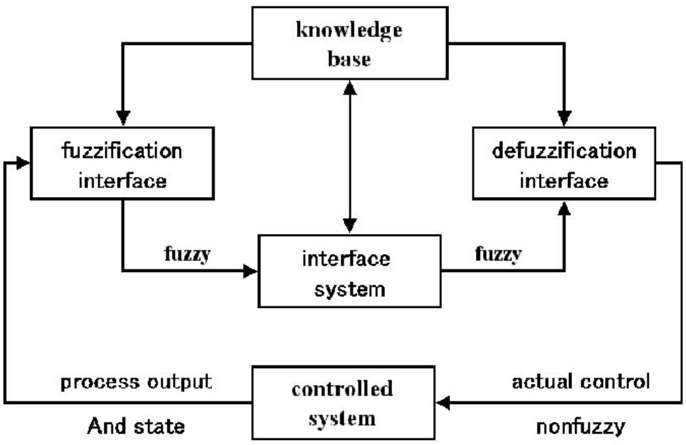


Figure 2. Generic structure of a FLC

Adapting such parameters automatically does not only improve the searching ability of the GA in finding the global optimum but also saves much time for fine-tuning them. The main idea is to use a FLC to compute new strategy parameter values of the GA with any combination of the performance measures (changes in the average fitness of the population *eval*(v)) and current parameters as the inputs to the controller. Here, Wang *et al.*'s (1997) concept is introduced for adjusting the crossover probability p_C and mutation probability p_M of GAs.

3. BASIC TRANSPORTATION MODELS

The transportation problem (TP) was formulated and proposed by Hitchcock (1941). Although this problem might seen almost too simple to have much applicability, the TP is very important in our real life applications.

3.1. Basic version of the TP

The basic version of the TP is a linear, single objective, balanced, and planar problem. Because the problem possesses a special structure in its constraints, an efficient optimization algorithm has been proposed for it, which is a variation of the simplex method adapted to the particular structure (Bazaraa *et al.*, 1993).

Nonlinear side constrained Transportation Problem: The transportation problem with nonlinear side constraints (nsc-TP) has many applications in our real world (Cao & Uebe, 1995). As one of examples, the posing of nscTP is as follows: In a container terminal which is divided into several areas (indexed by i) and arriving containers are classified into several categories (index by j and k) according to certain criteria. The problem of assignment of the storage positions for arriving container is to find a reasonable assignment strategy so that the costs of operations (searching for and/or loading containers) can be minimized. The side constraint represents some of necessary conditions (e.g. the limitation of the space in the storage so that some pair of different categories of containers can not be stacked in the same areas. In other word, the source i cannot serve two destinations j and k simultaneously). Without loss of generality, we can assume that the supply and demand in this problem perform a balance condition since we can convert a problem which is unbalanced into a balanced one by introducing a dummy source or destination.

Exclusionary side constrained Transportation Problem: In this model, the TP is extended to satisfy the additional constraint in which the simultaneous shipment from some pairs of source centres is prohibited. With this additional side constraint, the problem becomes enormously more difficult, yet the relevance for the real world applications also increases significantly. Moreover, since the side constraint is nonlinear, it is impossible to solve this problem using a traditional linear programming software package (such as LINDO) (Syarif & Gen, 2003).

Fixed Charge Transportation Problem: Linear TP is well-known as the simplest model of distribution problem. But fcTP is much more difficult to solve, due to the presence of fixed charges, which cause discontinuities in the objective function.

The fixed-charge transportation problem (fcTP) has a wide variety of classic applications that have been documented in the scheduling and facility location literature. Two of the most common of these arise: (1) in making warehouse or plant location decisions, where there is a charge for opening the facility, and (2) in transportation problems, where there are fixed charges for transporting goods between demand and supply points (Adlakha & Kowalski, 2003). In the fcTP, two types of costs are considered simultaneously when the best course of action is selection: (1) Variable costs proportional to the activity level, and (2) Fixed costs.

Indices

i index of plant (i=1,2,...,m)
j index of warehouse (j=1,2,...n)

Parameters

a_i number of units available at plant i
b_j number of units demanded at warehouse j

c_{ij} the cost of shipping one unit from plants i to warehouse j
d_{ij} fixed cost associated with route (i, j)

Decision variables

x_{ij} the unknown quantity to be transported on route (i, j)
$f_{ij}(\mathbf{x})$ total transportation cost for shipping per unit from plant i to warehouse j in which $f_{ij}(\mathbf{x})=c_{ij}x_{ij}$ will be a cost function if it is linear.

The usual objective function is to minimize the total variable cost and fixed costs from the allocation. It is one of the combinatorial problems involving constraints. This fcTP with m plants and n warehouses can be formulated as follows:

$$\min \quad f(x)=\sum_{i=1}^{m}\sum_{j=1}^{n}\left[f_{ij}(x)+d_{ij}g_{ij}(x)\right] \qquad (3.1)$$

$$\text{s.t.} \quad \sum_{j=1}^{n}x_{ij} \le a_i, \qquad i=1,2,\ldots,m \qquad (3.2)$$

$$\sum_{i=1}^{m}x_{ij} \ge b_j, \qquad j=1,2,\ldots,n \qquad (3.3)$$

$$x_{ij} \ge 0, \qquad \forall i,j \qquad (3.4)$$

$$\text{with} \quad g_{ij}(x)=\begin{cases}1, & \text{if } x_{ij}>0\\ 0, & \text{otherwise}\end{cases}$$

While equation (3.2) and (3.3) ensure the satisfaction of the plant's capacity and warehouse's demand, equation (3.4) enforces the non-negativity restriction on the decision variable.

3.2. Genetic Algorithms Approach

Representation: The priority-based encoding method that was adopted to escape the repair mechanisms in the search process of GA had been developed by (Gen & Cheng, 2000). A gene contains two kinds of information: the locus, the position of a gene located within the structure of a chromosome; and the allele, the value taken by the gene (Gen & Cheng, 1997). The position of a gene is used to represent a node, and the value is used to denote the priority of the node for constructing a tree among candidates.

For solving the fcTP, a chromosome $v_k(l)$ (l=1, 2,..., L, k=1,2,…, *popSize*, where *popSize* is total number of chromosomes in each generation) consists of priorities of plants and warehouses to obtain transportation tree, and its length is equal to total number of plants (m) and warehouses (n). Only one arc is added to tree for selecting a plant (warehouse) with the highest priority and connecting it to a warehouse (plant) which considers minimum unit cost.

Figure 3 shows the representation of fcTP with 3 plants and 7 warehouses. From first to third gene represents 3 plants and the others represent 7 warehouses.

node ID (l)	1	2	3	4	5	6	7	8	9	10
priority $v_k(l)$	5	1	7	10	4	9	2	6	3	8

Figure 3. Sample representation by priority-based encoding

Genetic Operators: Crossover and Mutation, we use genetic operators as follows: Partial-Mapped Crossover (PMX) and the Swap mutation are used. PMX uses a special repairing procedure to resolve the illegitimacy caused by the simple two-point crossover. Thus the essentials of PMX are a simple two-point crossover plus a repairing procedure. Swap mutation is used, which simply selects two positions at random and swaps their contents (Gen & Cheng, 2000).

Evaluation and selection: Evaluation function used for the GA is based on total transportation cost for shipping per unit and the fixed cost from plant *i* to warehouse *j* in this problem. The evaluation function is related to the objective function. Therefore, the evaluation function using total cost is defined as follows:

$$eval(v_k) = 1/f(x) = 1\Big/\sum_{i=1}^{m}\sum_{j=1}^{n}\left[f_{ij}(x) + d_{ij}g_{ij}(x)\right]$$

For the selection methods, we use elitist method that enforces the best chromosomes into the next generation. Because in elitism ensures that at least one copy of the best individual in the population is always passed onto the next generation, the convergence is guaranteed.

Local Search Techniques: The idea of combining GAs with local search (LS) techniques for solving optimization problems has been investigated extensively during the past decade, and various methods of hybridization have been proposed.

Since hybrid approach can combine the merits of GAs with those of LS technique, the hybrid approach with GA is less likely to be trapped in a local optimum than LS technique alone.

GAs are used for global exploration among the population of the GA, while LS techniques perform local exploitation around the convergence area of the GA. Because of the complementary properties between GAs and LS techniques, the hybrid approach often outperforms either the former or the latter alone.

One of the most common forms of the hybrid GA is to incorporate a LS technique into a conventional GA loop. With the hybrid GA, the LS technique is applied to each newly generated offspring to move it to a local optimum before injecting it into the population of the GA (Gen & Cheng, 2000). In this study, we adopt a LS technique which is applied to each new generation of the GA, select the best individual, and use insertion mutation until the offspring which the fitness is better than the best individual in offspring v_c is generated and inserts it into the population (Gen & Lin, 2004).

3.3. Numerical Experiments and Conclusions

We tested 4 problems taken from fcTP benchmark problems (Gamsworld [Online]). A comparison between our proposed algorithm and the best known results is described in this section. All experiments were realized using JAVA language under Pentium IV PC with 2.6 GHz CPU and 1GB RAM. Each simulation was run 30 times. GA parameter settings were taken as follows:

Population size: *popSize* =100
Maximum generation: *maxGen* =1000
Crossover probability, p_C = 0.70; Mutation probability, p_M = 0.50
Terminating condition, *T*=200 generations with the best solution not improved.

Table 1 shows the computational results of simple GA (sGA) and hybrid GA with Local Search (ls-hGA) to each test problem. By using ls-hGA, we can get same solutions and better solutions compared to s-GA in all test problems. The proposed ls-hGA can find the same solution in ran 10×10 (b), and near-best solution in ran 10×10 (c), ran 13×13.

Table 1. The computational results of each test problem

Test problems (# of plants× # of DC)	Best Known[12]	s-GA			ls-hGA		
		Worst	Average	Best	Worst	Average	Best
ran 10× 10 (a)	1499	1626	1613.2	1589	1626	1610.9	1589
ran 10× 10 (b)	3073	3264	3177.6	**3073**	3311	3157.8	**3073**
ran 10× 10 (c)	13007	13090	13027.0	**13020**	13090	13034.0	**13020**
ran 13× 13	3252	3522	3498.5	3424	3538	3482.3	**3286**

As explained above, we can find best solution and near-best solution by proposed ls-hGA approach. For more realistic problem, we generated 3 problems randomly larger size than fcTP benchmark problems.

Table 2. The computational results of three large-size problems

Test problems (# of plants× # of DC)	s-GA			ls-hGA		
	Worst	Average	Best	Worst	Average	Best
ran 20× 50	10305	10156.1	9921	10183	9759.1	**9306**
ran 30× 70	13955	13685.1	13212	12911	12592.8	**12123**
ran 40× 100	25333	25053.2	24198	24081	23221.7	**22478**

We simulated three problems 30 times ran 20×50, ran 30×70 and ran 40×100 respectively. GA parameter settings were as same as described above. The computational results are show in Table 2. Comparing s-GA with ls-hGA, we can get better solutions in all large-size problems. The proposed approach is effective to solve not only benchmark problems but large-size problems.

Four transportation models for logistics network have been introduced. Priority-based encoding methods and minimum cost-based decoding methods have been applied for solving the fcTP. To increase the performance of the proposed algorithm, we hybridize the proposed method with LS technique. We used LS technique which is applied to each newly generation, select the best individual and adopt Insertion Mutation until the offspring in which the fitness is better than best individual in offspring is generated, injecting it into the population.

4. MULTISTAGE LOGISTIC NETOWRKS

Multistage logistic network design is to provide an optimal platform for efficient and effective logistic systems. The problem is often an important and strategic operations management problem in logistic systems. The design task involves the choice of facilities (plants or DCs) to be opened and the distribution network design to satisfy the customer demand with minimum cost. This problem and its different versions have been studied in literature (Pirkul & Jayaraman, 1998; Azevedo & Sousa, 2000; Syam, 2002; Syarif, Yun & Gen, 2002; Yan *et al*., 2003; Jayaraman & Ross, 2003; Gen & Syarif, 2005, Gen *et al.*, 2006).

4.1. Two-stage Logistic Networks

The efficiency of the logistic system is influenced by many factors; one of them is to decide the number of DCs, and find the good location to be opened, in such a way that the customer demand can be satisfied at minimum DCs' opening cost and minimum shipping cost. In this paper, we consider an extension of two-stage logistic network problem (tsLNP). The problem aims to determine the transportation network to satisfy the customer demand at minimum cost subject to the plant and DCs capacity and also the maximum number of DCs to be opened.

Most companies have only limited resources to open and operate DCs. So, limiting the number of DCs that can be located is important when a manager has limited available capital. For this reason, the maximum number of DCs to be opened is considered as constraint in this study. We developed a priority-based Genetic Algorithm (pb-GA) with new decoding and encoding procedures considering the characteristic of tsLNP, and proposed a new crossover operator called as Weight Mapping Crossover (WMX). We carried out an experimental study into two-stages. While the effect of WMX on the performance of pb-GA was investigated in the first stage, pb-GA and another GA approach based on different representation method were compared according to solution quality and solution time in the second stage.

The tsLNP considered in the study aims to determine the distribution network to satisfy the customer demand at minimum cost subject to the plant and DCs capacity and also the minimum number of DCs to be opened. We assumed that the customer locations and their demand were known in advance. The numbers of potential DC locations as well as their maximum capacities were also known. The mathematical model of the problem is:

$$\min \quad Z=\sum_{i=1}^{I}\sum_{j=1}^{J}t_{ij}x_{ij}+\sum_{j=1}^{J}\sum_{k=1}^{K}c_{jk}y_{jk}+\sum_{j=1}^{J}g_j z_j \qquad (4.1)$$

$$\text{s.t.} \quad \sum_{j=1}^{J}x_{ij}\le a_i, \qquad \forall i \qquad (4.2)$$

$$\sum_{k=1}^{K}y_{jk}\le b_j z_j, \qquad \forall j \qquad (4.3)$$

$$\sum_{j=1}^{J}z_j\le W, \qquad (4.4)$$

$$\sum_{j=1}^{J}y_{jk}\ge d_k, \qquad \forall k \qquad (4.5)$$

$$\sum_{i=1}^{I}\sum_{j=1}^{J}x_{ij}=\sum_{j=1}^{J}\sum_{k=1}^{K}y_{jk} \qquad (4.6)$$

$$x_{ij}, y_{jk}\ge 0, \qquad \forall i,j,k \qquad (4.7)$$

$$z_j=\{0,1\} \qquad \forall j \qquad (4.8)$$

where:

I: number of plants (i = 1,2,…,I), J : number of distribution centres (j = 1,2,…,J), K : number of customers (k=1,2,…,K), a_i : capacity of plant i, b_j : capacity of distribution centre j, d_k : demand of customer k, t_{ij} : unit cost of transportation from plant i to distribution centre j, c_{jk} : unit cost of transportation from distribution centre j to customer k, g_j : fixed cost for operating distribution centre j, W : an upper limit on total number of DCs that can be opened, x_{ij} : the amount of shipment from plant i to distribution centre j, y_{jk} : the amount of shipment from distribution centre j to customer k, z_j : 0-1 variable that takes on the value 1 if DC j is opened.

While constraints (4.2) and (4.3) ensure that the plant-capacity constraints and the distribution centre-capacity constraints, respectively, constraint (4.4) satisfies the opened DCs do not exceed their upper limit. This constraint is very important when a manager has limited available capital. Constraint (4.5) ensure that all demand of customers are satisfied by opened DCs; Constraints (4.6) and (4.7) enforce the non-negativity restriction on the decision variables and the binary nature of the decision variables used in this model. Without loss of generality, we assume that this model satisfies the balanced condition, since the unbalanced problem can be changed balanced one by introducing dummy suppliers or dummy customers.

4.1.1. Priority-Based Genetic Algorithms

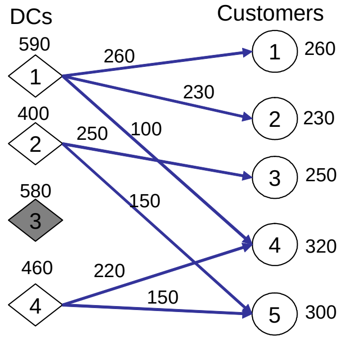
Representation: Michalewicz *et al* (1991) were the first researchers who used GA for solving linear and non-linear transportation/distribution problems. In their approach, matrix-based representation had been used. When m and n are the number of sources and depots respectively, the dimension of matrix is $m \times n$. Although representation is very simple, there is need to special crossover and mutation operators for obtaining feasible solutions.

The use of spanning tree GA (st-GA) for solving some network problems was introduced by Gen and Cheng (1997, 2000). They employed Prüfer number in order to represent a candidate solution to the problems and developed feasibility criteria for Prüfer number to be decoded into a spanning tree. They noted that the use of Prüfer number is very suitable for encoding a spanning tree, especially in some research fields, such as transportation problems, minimum spanning tree problems, and so on.

In this study, to escape from these repair mechanisms in the search process of a GA, we propose a new encoding method based on priority-based encoding developed by (Gen & Cheng, 1997). This encoding had been successfully applied on shortest path problem and project scheduling problem (Gen & Cheng, 2000).

For the problem, a chromosome consists of priorities of sources and depots to obtain transportation tree and its length is equal to total number of sources (m) and depots (n), i.e. $m+n$. The transportation tree corresponding with a given chromosome is generated by sequential arc appending between sources and depots. At each step, only one arc is added to tree selecting a source (depot) with the highest priority and connecting it to a depot (source) considering minimum cost. Figure 4 represents a transportation tree with 4 sources and 5 depots, its cost matrix and priority based encoding.

$$c_{jk} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 15 & 18 & 16 & 19 & 21 \\ 17 & 20 & 12 & 18 & 15 \\ 25 & 23 & 15 & 21 & 19 \\ 20 & 16 & 13 & 14 & 18 \end{bmatrix} \end{array}$$



node ID l :	1	2	3	4	1	2	3	4	5
priority $v(l)$:	7	5	2	4	6	3	9	1	8

Figure 4. A sample of transportation tree and its encoding

Genetic operators: In this study, we propose a new crossover operator called as weight mapping crossover (WMX) and investigate the effects of four different crossover operators on the performance of a GA. WMX can be viewed as an extension of one-point crossover for permutation encoding. As in one-point crossover, after determining a random cut-point, the offspring are generated by using left segment of the cut-point and caring out remapping on the right segment of own parent. In the remapping process, after obtaining an increasing order of digits on the right segments of parents and mapping digits on the ordered parts, new right segment of the first offspring is obtained using original sequence of right segment of the second parent and its mapped digits on the first parent. When obtaining new right segment of second parent, original sequence of right segment of the first parent and its mapped digits on the second parent are used. Figure 5 shows an example of WMX. As it is seen in the Figure 5, left segments of parents have been copied to offspring based on the cut-point selected as 4 and an increasing orders of right segments are mapped as 3 ↔ 1, 5 ↔ 4 and 6 ↔ 5. A new right segment of first offspring is obtained as 6, 3 and 5 using original sequence of second parent as 5, 1 and 4 and its mapped digits on the first parent. Using same procedure, it is possible to obtain the second offspring.

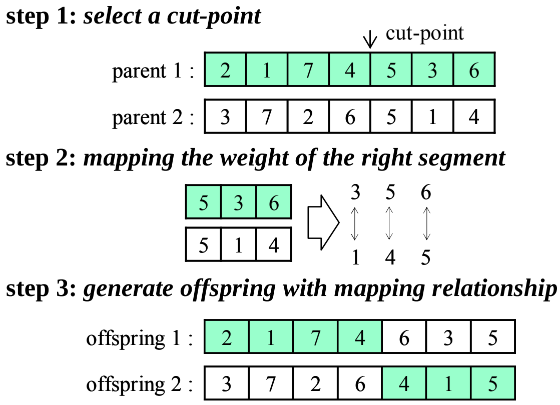


Figure 5. Illustration of the WMX

Similar to crossover, mutation is done to prevent the premature convergence and explore new solution space. However, unlike crossover, mutation is usually done by modifying gene within a chromosome. We also investigate the effects of two different mutation operators on the performance of GA. Insert and swap mutations are used for this purpose.

4.1.2. Numerical Examples

To investigate the effectiveness of the developed GAs with new encoding method (pb-GA), we used spanning tree-based GA (st-GA) using Prüfer number proposed by Syarif and Gen (2003). Seven different test problems were considered.

Table 3 gives computational results for the st-GA and pb-GA based on Prüfer number encoding and priority-based encoding methods, respectively, on seven test problems. In st-GA, one-cutpoint crossover and insertion mutation operators were used as genetic operators and its rates were taken as 0.5. Each test problem is run by 10 times using GA approaches. To make comparison between st-GA and pb-GA according to solution quality and computational burden, we consider again best, average and worst costs and also ACT. In addition, each test problem is divided into three numerical experiments to investigate the effects of population size and number of generations on the performance of st-GA and pb-GA. When we compare columns of the best cost of the st-GA and pb-GA, it is possible to see that pb-GA developed in this study reaches optimum solutions for the first four test problems, while st-GA finds optimum solution for only the first problem. In addition, average percent deviation from optimum solution on st-GA changes between 2.31% and 30% except to the first problem. For big size problems, i.e. last three problems, the best costs of pb-GA are always smaller than found with st-GA.

Table 3. Computational Results for st-GA and pb-GA

Parameters			st-GA				pb-GA			
Problem	Popsize	maxgen	Best	Average	Worst	ACT	Best	Average	Worst	ACT
1	10	300	1089	1175.4	1339	0.07	1089	1089.0	1089	0.12
	15	500	1089	1091.8	1099	0.16	1089	1089.0	1089	0.23
	20	1000	1089	1089.0	1089	0.35	1089	1089.0	1089	0.57
2	20	1000	2341	2402.5	2455	0.48	2283	2283.2	2285	0.78
	30	1500	2291	2375.2	2426	1.06	2283	2283.0	2283	1.76
	50	2000	2303	2335.8	2373	2.42	2283	2283.0	2283	4.10
3	30	1500	2781	2874.4	2942	1.25	2527	2527.0	2527	2.04
	50	2500	2719	2787.1	2874	3.43	2527	2527.0	2527	5.91
	100	4000	2623	2742.2	2796	11.85	2527	2527.0	2527	21.32
4	75	2000	3680	3873.8	4030	7.78	2886	2891.2	2899	12.99
	100	3000	3643	3780.4	3954	15.93	2886	2892.6	2899	26.85
	150	5000	3582	3712.5	3841	41.41	2886	2890.0	2893	71.76
5	75	2000	5738	5949.1	6115	18.29	2971	2985.3	3000	29.07
	100	3000	5676	5786.1	5889	36.88	2967	2980.6	2994	59.13
	150	5000	5461	5669.4	5835	94.33	2952	2973.2	2989	153.02
6	100	2000	7393	7705.6	8067	36.27	2975	2999.0	3025	56.32
	150	3000	7415	7563.8	7756	76.23	2963	2994.3	3005	130.29
	200	5000	7068	7428.5	7578	188.37	2962	2984.9	3000	295.28
7	100	2000	10474	11083.1	11306	177.03	3192	3204.2	3224	241.74
	150	3000	10715	10954.7	11146	395.52	3148	3184.3	3207	548.30
	200	5000	10716	10889.4	11023	875.03	3136	3179.6	3202	1213.65

4.2. Multiobjective Three-stage Logistic Networks

The design task of three-stage logistic networks, involves the choice of facilities (plants and distribution centres) to be opened and the distribution network design to satisfy the customer demand with minimum cost. Syarif, Yun and Gen (2002) propose a spanning tree-based genetic algorithm approach. Altiparmak, Gen and Lin (2004) and Altiparmak *et al.* (2006) propose a genetic algorithm with priority-based encoding for three-stage logistic network problem.

This study proposes a new solution procedure based on genetic algorithms to find a set of Pareto-optimal solutions for multi-objective SCN design problem. To deal with multi-objective and enable the decision maker for evaluating a greater number of alternative solutions, two different weight approaches are implemented in the proposed solution procedure.

The mathematical notation and formulation are as follows:

Objectives: f_1 is the total cost of SCN. It includes the fixed costs of operating and opening plants and DCs, the variable costs of transportation raw material from suppliers to plants and the transportation the product from plants to customers through DCs. f_2 is the total customer demand (in %) that can be delivered within the stipulated access time τ. f_3 is the equity of the capacity utilization ratio for plants and DCs, and it is measured by mean square error (MSE) of capacity utilization ratios. The smaller the value is, the closer the capacity utilization ratio for every plant and DC is, thus ensuring the demands are fairly distributed among the opened DCs and plants, and so it maximizes the capacity utilization balance.

$$\min \quad f_1 = \sum_k g_k p_k + \sum_j v_j z_j + \sum_s \sum_k t_{sk} b_{sk} + \sum_k \sum_j a_{kj} f_{kj} + \sum_j \sum_i c_{ji} q_{ji} \quad (4.9)$$

$$\max \quad f_2 = \left(\sum_{j \in o_D} \sum_{i \in C(j)} q_{ji} \right) \Big/ \left(\sum_i d_i \right) \quad (4.10)$$

$$\min \quad f_3 = r_1 \left[\sum_{k \in o_P} \left[(\sum_{j \in o_D} f_{kj} / D_k) - (\sum_{k \in o_P} \sum_{j \in o_D} f_{kj} / \sum_{k \in o_P} D_k) \right]^2 / | o_P | \right]^{1/2} \quad (4.11)$$

$$+ r_2 \left[\sum_{j \in o_D} \left[(\sum_i q_{ji} / W_j) - (\sum_{j \in o_D} \sum_i q_{ji} / \sum_{j \in o_D} W_j) \right]^2 / | o_D | \right]^{1/2}$$

$$\text{s.t.} \quad \sum_j y_{ji} = 1, \quad \forall i \quad (4.12)$$

$$\sum_i d_i y_{ji} \le W_j z_j, \quad \forall j \quad (4.13)$$

$$\sum_j z_j \le W \quad (4.14)$$

$$q_{ji} = d_i y_{ji}, \quad \forall i, j \quad (4.15)$$

$$\sum_k f_{kj} = \sum_i q_{ji}, \quad \forall j \quad (4.16)$$

$$\sum_k b_{sk} \le \sup_s, \quad \forall s \quad (4.17)$$

$$u \sum_j f_{kj} \le \sum_s b_{sk}, \quad \forall k \quad (4.18)$$

$$u \sum_j f_{kj} \le D_k p_k, \quad \forall k \quad (4.19)$$

$$\sum_k p_k \le P \quad (4.20)$$

$$z_j = \{0,1\}, \quad \forall j \quad (4.21)$$

$$p_k = \{0,1\}, \quad \forall k \quad (4.22)$$

$$y_{ji} = \{0,1\}, \quad \forall i, j \quad (4.23)$$

$$b_{sk} \ge 0, \quad \forall s, k \quad (4.24)$$

$$f_{kj} \ge 0, \quad \forall j, k \quad (4.25)$$

$$q_{ji} \ge 0, \quad \forall i, j \quad (4.26)$$

where:

Indices: i is an index for customers ($i \in I$). j is an index for DCs ($j \in J$). k is an index for manufacturing plants ($k \in K$). s is an index for suppliers ($s \in S$).

Model variables: b_{sk} is the quantity of raw material shipped from supplier s to plant k. f_{kj} is the quantity of the product shipped from plant k to DC j. q_{ji} is the quantity of the product shipped from DC j to customer i.

$$z_j = \begin{cases} 1 & \text{if DC } j \text{ is open} \\ 0 & \text{otherwise} \end{cases}$$

$$p_k = \begin{cases} 1 & \text{if plant } k \text{ is open} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ji} = \begin{cases} 1 & \text{if DC } j \text{ serves customer } i \\ 0 & \text{otherwise} \end{cases}$$

Model Parameters: D_k is the capacity of plant *k*. W_j is the annual throughput at DC *j*. sup_s is the capacity of supplier *s* for raw material. d_i is the demand for the product at customer *i*. *W* is the maximum number of DCs. *P* is the maximum number of plants. v_j is the annual fixed cost for operating a DC *j*. g_k is the annual fixed cost for operating a plant *k*. c_{ji} is the unit transportation cost for the product from DC *j* to customer *i*. a_{kj} is the unit transportation cost for the product from plant *k* to DC *j*. t_{sk} is the unit transportation and purchasing cost for raw material from supplier *s* to plant *k*. *u* is the utilization rate of raw material per unit of the product. h_{ji} is the delivery time (in hours) from DC *j* to customer *i*. τ is the maximum allowable delivery time (hours) from warehouses to customers. *C*(*j*) is the set of customers that can be reached from DC *j* in τ hours, or $C(j) = \{ i \mid h_{ji} \le \tau \}$. o_D is the set of opened DCs, o_P is the set of opened plants. r_1 and r_2 are the weights of plants and DCs, respectively.

4.2.1. Priority-Based Genetic Algorithms

Representation: In this study, to escape from these repair mechanisms in the search process of the GA, priority-based encoding developed by Gen & Cheng (2000) was used. They had successfully applied this encoding to the shortest path problem and the project scheduling problem. The first application of this encoding structure to a single product transportation problem was carried out by Gen *et al.* (2006), and its extension to design of multi-product, multi-stage SCN had been made by Altiparmak *et al*. (2006). As it is known, a gene in a chromosome is characterized by two factors: locus, the position of the gene within the structure of chromosome, and allele, the value the gene takes. In priority-based encoding, the position of a gene is used to represent a node (source/depot in transportation network), and the value is used to represent the priority of corresponding node for constructing a tree among candidates.

For a transportation problem, a chromosome consists of priorities of sources and depots to obtain transportation tree and its length is equal to total number of sources (|*K*|) and depots (|*J*|), i.e. |*K*|+|*J*|. The transportation tree corresponding with a given chromosome is generated by sequential arc appending between sources and depots. At each step, only one arc is added to tree selecting a source (depot) with the highest priority and connecting it to a depot (source) considering minimum cost. The decoding algorithm is same with Subsection 4.1.1.

Evaluation: An important issue in multi-objective optimization is how to determine the fitness value of the chromosome for survival. The fitness value of each individual reflects how good it is based upon its achievement of objectives. In literature, there are different techniques to define fitness function (Gen & Cheng, 2000). One of them, also simplest approach, is weight-sum technique. Given *b* objective functions, fitness function is obtained by combining the objective functions

$$Eval(f) = \sum_{i=1}^{b} w_i f_i$$

where w_i is constant representing weight for f_i, and $\sum_{i=1}^{b} w_i = 1$

To determine the weight values, two approaches proposed by Murata *et al.* (1996) and Zhou & Gen (1999) were adopted. Approach I is based on random weight approach in which weights are randomly determined for each step of evolutionary process (Murata *et al.*, 1996). This approach explores the entire solution space in order to avoid local optima and thus gives a uniform chance to search all possible Pareto solutions along the Pareto frontier. In Approach II, weights are determined based on the ideal point generated in each evolutionary process (Gen & Cheng, 2000).

4.2.2. Numerical Experiments

Table 4 gives information about suppliers' capacities, and capacity and fixed costs for plants and DCs. As it is seen from Table 4, fixed costs of plants are different from each other, although their capacities are equal. Fixed cost of plants consists of expenditures such as hiring costs of buildings and facilities; amortization of machines and tools; salaries of managers and guardians; and insurance premiums. Although amortizations, fixed man-power and insurance cost are approximately equal in Turkey, land and building costs depend on the developing and industrialization level of cities. Thus, differences between fixed costs of plants come from this fact.

Table 4. Capacities and fixed costs for suppliers, plants, and DCs

Suppliers	Capacity (ton/year)	Plants	Capacity (package /year)	Fixed Cost (USD/year)	DCs	Capacity (package /year)	Fixed Cost (USD/year)
USA	10000	Konya	640000	440000	Konya	200000	70000
Belgium	10000	Istanbul	640000	1100000	Istanbul	160000	60000
France	10000	Izmir	640000	720000	Izmir	80000	40000
Japan	10000				Ankara	120000	50000
Petkim	7200				Trabzon	80000	40000
					Adana	120000	50000

The problems and their objective functions are listed below:

Problem 1: min f_1 and max f_2
Problem 2: min f_1 and min f_3
Problem 3: min f_1, max f_2 and min f_3

In the rest of the paper, the proposed GA with Approach 1 and Approach 2 will be called as GA_A1 and GA_A2, respectively.

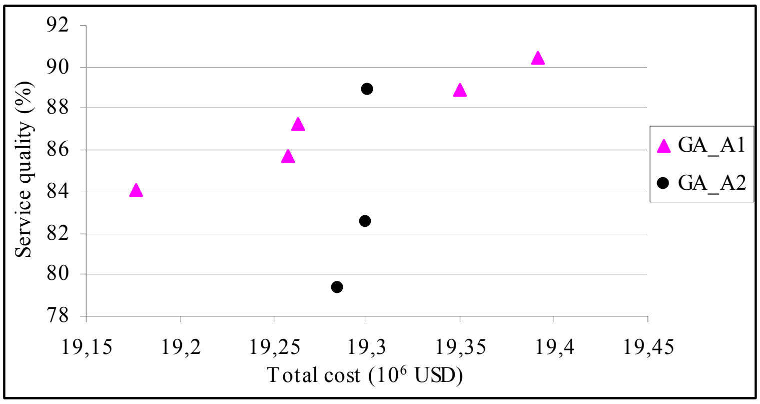


Figure 6. Pareto-optimal solutions of GA_A1 and GA_A2 for Problem 1

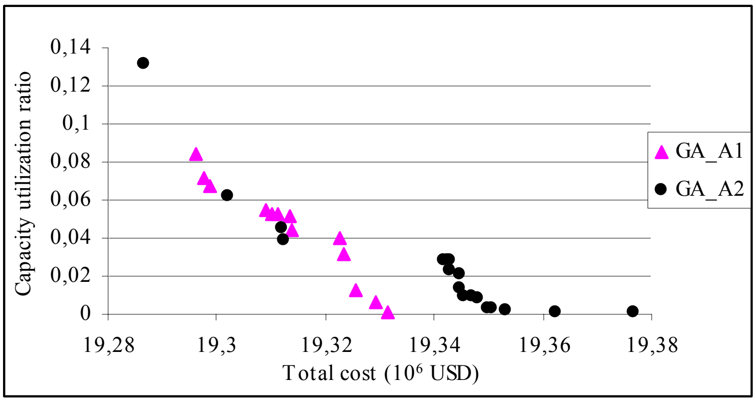


Figure 7. Pareto-optimal solutions of GA_A1 and GA_A2 for Problem 2

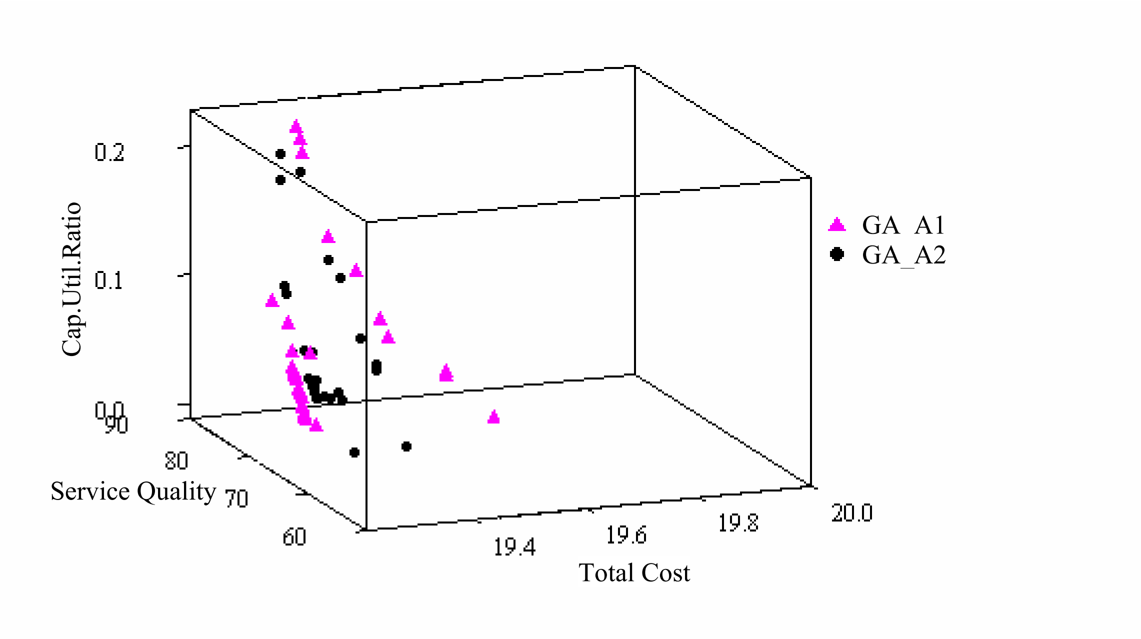


Figure 8. Pareto-optimal solutions of GA_A1 and GA_A2 for Problem 3

5. VEHICLE ROUTING PROBLEM MODELS

Vehicle routing problem (VRP) is a generic name given to a whole class of problems in which a set of routes for a fleet of vehicles based at one or several depots must be determined for a number of geographically dispersed cities or customers. The objective of the VRP is to deliver a set of customers with known demands on minimum-cost vehicle routes with minimum number of vehicles originating and terminating at a depot.

VRP is a well known integer programming problem which falls into the category of NP-hard problems, meaning that the computational effort required solving this problem increase exponentially with the problem size. For such problems it is often desirable to obtain approximate solutions, so they can be found fast enough and are sufficiently accurate for the purpose. Usually this task is accomplished by using various heuristic methods, which rely on some insight into the problem nature (VRP Web [Online]).

Capacitated VRP (cVRP): cVRP is a VRP in which a fixed fleet of delivery vehicles of uniform capacity must service known customer demands for a single commodity at minimum transit cost.

VRP with time windows (VRP-tw): The time window constant is denoted by a predefined time interval, given an earliest arrival time and latest time. Each customer also imposes a service time to the route, taking consideration of the service time of goods.

VRP with Pick-up and Delivery (VRP-pd): VRP-pd is a VRP in which the possibility that customers return some commodities is contemplated. So in VRP-pd it's needed to take into account that the goods that customers return to the deliver vehicle must fit into it.

VRP with simultaneous Pick-up and Delivery (VRP-sPD): The problem dealing with a single depot distribution/collection system servicing a set of customers by means of a homogeneous fleet of vehicles. Each customer requires two types of service, a pickup and a delivery. The critical feature of the problem is that both activities have to be carried out simultaneously by the same vehicle (each customer is visited exactly once). Products to be delivered are loaded at the depot and products picked up are transported back to the depot. The objective is to find the set of routes servicing all the customers at the minimum cost.

VRP with Backhauls (VRP-b): VRP-b is a VRP in which customers can demand or return some commodities. So in VRP-pd it's needed to take into account that the goods that customers return to the deliver vehicle must fit into it. The critical assumption in that all deliveries must be made on each route before any pickups can be made. This arises from the fact that the vehicles are rear-loaded, and rearrangement of the loads on the tracks at the delivery points is not deemed economical or feasible. The quantities to be delivered and picked-up are fixed and known in advance.

Multiple Depot VRP (mdVRP): A company may have several depots from which it can serve its customers. The mdVRP can be solved in two stages: first, customers must be assigned to depots; then routes must be built that link customers assigned to the same depot.

Split Delivery VRP (sdVRP): sdVRP is a relaxation of the VRP wherein it is allowed that the same customer can be served by different vehicles if it reduces overall costs. This relaxation is very important if the sizes of the customer orders are as big as the capacity of a vehicle.

5.1. Problem description (mdVRP-tw)

To solve multi-depot VRP-tw (mdVRP-tw), when the number of customers is usually much larger than that of DC, the cluster approach can be adopted first, and then route ones. mdVRP-tw become more complex as it involves servicing customers with time windows using multiple vehicles that vary in number with respect to the problem. Therefore, mdVRP-tw should be designed as follows:

(1) All distances are represented by Euclidean distance.
(2) Each customer is serviced by one of depots.
(3) Each route starts a depot and then returns the depot.
(4) Each customer can be visited only once by a vehicle.
(5) The vehicle capacity of each route is equal.
(6) Total customer demand for each route does not exceed the vehicle capacity.
(7) Each customer is associated with a time window period for its service time.
(8) Each vehicle has maximum travel time.

The objective for solving mdVRP-tw is to determine depot and vehicle routing system in order to achieve the minimal cost without violating the DC capacity and time window constraints. mdVRP-tw is an NP-hard problem due to an NP-hard of VRP-tw. The mdVRP-tw is to determine the set of vehicle routing that can satisfy the customer demand within its time-window constraints, thus, it is divided into two phases. First phase is to cluster customers and then vehicle routing phase is considered.

5.1.1. Clustering customers (Phase 1)

In this phase, sets of customers are divided into regionally bounded sets that satisfy restrictions to ensure within the customers. Each customer should be serviced by one of DCs shown in Figure 9.

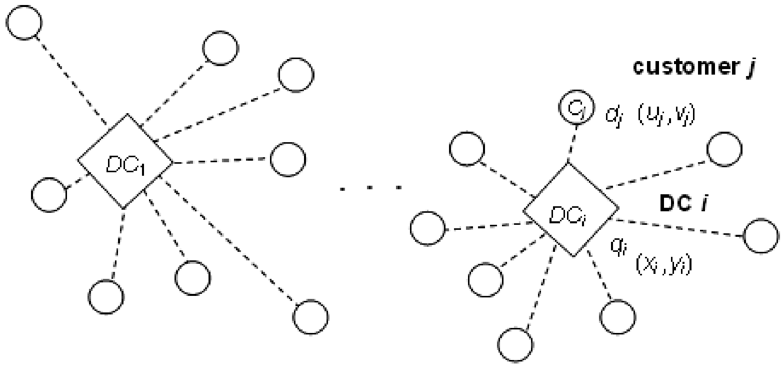


Figure 9. Assigning each customer to DC

The objective here is to determine the DC to satisfy the customer demand so that the total distance is minimized. The mathematical model is formulated as follows:

Indices:

i index of DC (i=1,2,...,m)
j index of customer (j=1,2,...,n)

Parameters:

q_i maximum capacity of DC i
d_j demand of customer j
p_{ij} distance from customer j to DC i

Decision Variable:

x_{ij}=1 if customer j is assigned to DC i. Otherwise 0

The mathematical model for this phase is given as follows:

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} \tag{5.1}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} d_j x_{ij} \le q_i, \quad i = 1,2,\dots,m \tag{5.2}$$

$$\sum_{i=1}^{m} x_{ij} = 1, \quad j = 1,2,\dots,n \tag{5.3}$$

$$x_{ij} \in \{0,1\}, \quad \forall i,j \tag{5.4}$$

The constraint (5.2) shows that the total customer demand assigned by a specific DC i does not exceed the capacity of the DC. The constraint (5.3) shows that each customer should be served by only one of the DC.

5.1.2. Vehicle routing (Phase 2)

Due to the output from the previous phase, the set customers assigned to each DC is determined. At this phase, it is favourable to make the vehicle routing to satisfy all constraints, at the same time minimizing the total travel distance.

The time window constraint is denoted by a predefined time interval, given an earliest arrival time and latest arrival time. The vehicles must arrive at the customers within the latest arrival time, while arrive earlier than the earliest arrival time, waiting occurs.

Index:

e index of vehicle

Parameters:

n total number of customers
g_i earliest arrival time at customer j
l total number of vehicles
h_j latest arrival time at customer j
Y_e capacity of vehicle e
s_{ij} service time at customer j in DC i
c_{jk} distance from customer j to customer k
t_{jk} travel time from customer j to customer k
r_{ei} maximum time of a route allowed for vehicle e in DC i

Decision Variables:

a_{ij} arrival time at customer j in DC i
w_{ij} waiting time at customer j in DC i
$z_{eijk}=1$ if the vehicle e travels from customer j to k in DC i. Otherwise $z_{eijk}=0$.

Here, it is beneficial to determine the set of vehicle routes to satisfy the customer demand within its time window periods. The mathematical model is formulated as follows:

$$\min \quad \sum_{i=1} \sum_{e \in E_i} \sum_{j \in C_i} \sum_{k \in C_i} c_{jk} z_{eijk} \tag{5.5}$$

$$\text{s.t.} \quad \sum_{e \in E_i} \sum_{j \in C_i} z_{eijk} = 1, \quad \forall i,\ k \in C_i - \{0\} \tag{5.6}$$

$$\sum_{j \in C_i - \{0\}} \sum_{k \in C_i} d_j z_{eijk} \le Y_e, \quad \forall i = 1,\ e \in E_i \tag{5.7}$$

$$\sum_{k \in C_i} z_{ei0k} = 1, \quad \forall e \in E_i,\ i \tag{5.8}$$

$$\sum_{j \in C_i - \{0\}} z_{eijk} - \sum_{k \in C_i - \{0\}} z_{eijk} = 0, \quad \forall e \in E_i,\ i \tag{5.9}$$

$$\sum_{j \in C_i} z_{eij0} = -1, \quad \forall e \in E_i,\ i \tag{5.10}$$

$$a_{ik} + M(1 - z_{eijk}) \ge (a_{ij} + s_{ij}) + t_{jk}, \quad \forall i,\ j,\ k \in C_i,\ e \in E_i \tag{5.11}$$

$$\sum_{j=1}^{n_i} \sum_{k=1}^{n_i} (s_{ij} + t_{jk} + w_{ik}) z_{eijk} \le r_{ei}, \ \forall e,\ i \tag{5.12}$$

$$\sum_{e=1}^{E} \sum_{j=1}^{n} (a_{ij} + s_{ij} + t_{jk} + w_{ik}) z_{eijk} \le a_{ik}, \ \forall i,\ k \tag{5.13}$$

$$g_j \le (a_{ij} + w_{ij}) < h_j, \ \forall i,\ j \tag{5.14}$$

$$z_{eijk} = 0 \text{ or } 1 \quad \forall e \in E_i,\ i,\ j,\ k \in C_i \tag{5.15}$$

The constraint (5.6) shows that only one vehicle e directly go from customer j to customer k. The constraint (5.7) shows that the total demand of customer in each vehicle route is less than the capacity of vehicle e. The constraint (5.8), (5.9) and (5.10) ensure that each vehicle leaves the depot 0, after arriving at a customer the vehicle leaves again and finally returns to depot. The inequality (5.11) states that a vehicle e in DC i cannot arrive at customer k before $(a_{ij}+w_{ij})+t_{jk}$ if it travels from customer j to customer k. The constraint (5.12) shows the maximum travel time. Finally, the constraint (5.13) and (5.14) ensure that the time windows are observed.

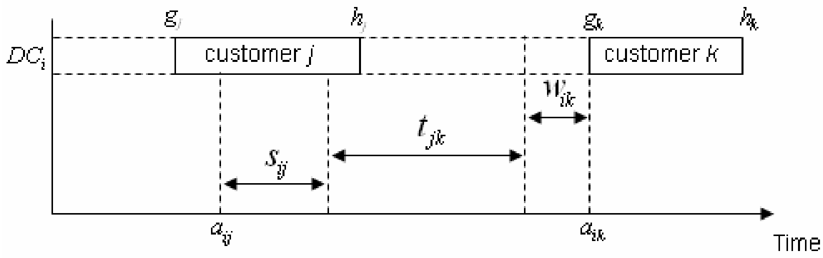


Figure 10. Time window of time constraints

5.2. Genetic Algorithms Approach

Clustering customers (Phase 1): The aim of this phase is to determine the assignment of customers to each DC so that the total distance is minimized.

Parallel assignment: A parallel assignment for clustering customers was adopted. The name parallel is due to the fact that the urgency for each customer is calculated considering all depots at the same time (Tansini, Urquhart and Viera, 1999).

Vehicle routing (Phase 2): The aim of this phase is to develop the vehicle routing from DCs satisfying the time window constraint.

Genetic representation: In this step, GA with priority-based encoding method is proposed to escape the repair mechanisms in the search process of GA. The priority-based encoding method had been developed by Cheng and Gen (1997) and applied to many problems such as shortest path problem, project scheduling problem.

node ID (l)	1	2	3	4	5	6	7	8	9	10	11	12
priority $v(l)$	4	12	1	11	2	9	3	6	7	5	8	10

Figure 11. Sample representation by priority-based encoding

All the customers are sorted in increasing order of earliest arrival time. The sorted customer number by node ID in a chromosome was used. The sample representation by priority-based encoding is represented in Figure 11.

At each step, only one customer is added to set selected by the highest priority and find the next customer considering minimum distance. The sequence of route was considered, first assigned customer form DC is r, the next is u, u+1, and so on.

In time window constraints, start time at customer j t_j^S has to be considered, which is the duration from starting time to the next customer. Finish time at customer j t_j^F means the time of finishing the service at customer j. The customer which is selected by the highest priority is not the only consideration; the left and right gene from it is also considered.

In the encoding procedure, the new priority is divided by the ID No and taken from the original priority. By using this method, more customers in a route can be assigned. The sample representation by new priority-based encoding is represented in Figure 12.

node ID (l)	1	2	3	4	5	6	7	8	9	10	11	12
priority $v(l)$	4.00	6.00	0.33	2.75	0.40	1.50	0.42	0.75	0.77	0.50	0.72	0.83

Figure 12. The sample representation by new priority-based encoding

Crossover and Mutation: Genetic operators are used as follows: Order Crossover (OX) and the Swap mutation are used. It can be viewed as a kind of PMX that uses a different repair procedure. Swap mutation is used, which simply selects two positions at random and swaps their contents.

Evaluation and selection: The evaluation function using total distance is defined as follows:

$$eval(v_k) = \frac{1}{f(x)}$$
$$= \frac{1}{\sum_{i=1} \sum_{e \in E_i} \sum_{j \in C_i} \sum_{k \in C_i} c_{jk} z_{eijk}}$$

For the selection methods, *elitist* methods that enforces the best chromosomes into the next generation are used. Because elitism ensures that at least one copy of the best individual in the population is always passed onto the next generation, the convergence is guaranteed.

5.3. Numerical Experiments

To prove the efficiency of the proposed GA approaches, several problems comparing the result of two approaches were tested. In this study, six test problems were generated and each problem consists of small size (2 DCs / 60 customers) and large size (3DCs /100 customers). The geographical data are randomly generated in each problem. Maximum load of vehicles is 150 in all test problems. Three factors for more realistic vehicle routing problem are also considered:

(1) Capacities of DCs
(2) A mix of short scheduling and a long scheduling in a problem
(3) Different service time for customers

All of problems are represented in Appendix. Six problems are tested by using proposed GA and represents the customer routes and total distances. All experiments were realized using C language under Pentium IV PC with 2.7 GHz CPU and 1GB RAM. GA parameter settings were taken as follows:

Population size: *popSize* =100
Maximum generation: *maxGen* =1500
Crossover probability, p_C = 0.70; Mutation probability, p_M = 0.50
Terminating condition, T=200 generations with the best solution not improved.
Table 5 represents the fleet of vehicles and total distance of each test problem.

Table 5. Computational results of each test problems

Test No.	# of DCs / # of customers	Proposed GA-1	
		NV	TD
1-1	2 / 60	12	982.334
1-2	3 /100	20	1771.903
2-1	2 / 60	12	826.374
2-2	3 /100	17	1472.461
3-1	2 / 60	13	878.753
3-2	3 /100	18	1489.279

Vehicle routing problem (VRP) and variants of VRP were introduced. Multi-depot vehicle routing problem with time windows (mdVRP-tw) have been considered. For implementing the mdVRP-tw, it was divided into two phases. In the first phase, a cluster approach has been adopted, and then the route ones have been used, while satisfying each customer demand. In the second phase, we have determined the vehicle routing from the DCs satisfying time window constraints. Since the mdVRP-tw is very difficult to be solved optimally, we proposed priority-based genetic algorithm (pGA). In numerical experiments, we tested six problems by using proposed GA. All of the customer route and total distances are represented. For near future work, we will also expand our research area to a VRP with pickup and delivery (VRP-pd) and develop a suitable solution based on GA for this problem.

6. LOCATION-ALLOCATION MODELS

Location-allocation problems concern the optimal number and location of DCs needed to provide some service to a set of customers. The optimal solution must balance two types of costs the cost of establishing a DC at a particular location and the total cost of providing service to each of the customers from one of the opened DCs. In its simplest form, if each opened DC can serve only a limited number of customers the problem is called capacitated.

6.1. Capacitated Location Allocation problem (cLAP)

Cooper was the first author of formally recognize and state the multi-Weber problem. He proposed a heuristics called alternative location-allocation, which is the best heuristic available. With the development of nonlinear programming techniques, relaxing integer allocation constraints and treating the location variables and allocation variables simultaneously, some new methods have been developed (Cooper, 1963).

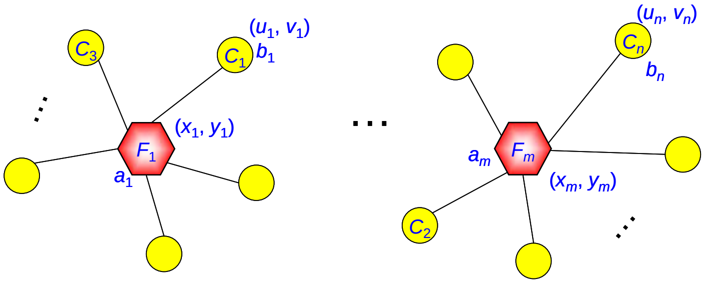


Figure 13. Capacitated Location-Allocation Model

In Cooper's location allocation model, it is assumed that a DC has an infinite service capacity. However, this assumption is not the case in practice. The capacity constraint is an important factor in DC location analysis.

The cLAP is a well-known combinatorial optimization problem with applications in production and distribution system. It is more complex because the allocation subproblem is a general assignment problem known as the NP-hard combinatorial optimization problem.

Although the approaches for the location-allocation problem can be extended to cLAP, there is no method capable of finding a global or near global optimal solution, especially for practical scale problems. It is important and necessary to develop an efficient method to find the global or near global optimal solution for cLAP.

In this study, we are interested in finding the location of m DCs in continuous space in order to serve customers at n fixed points as well as the allocation of each customer to the DCs so that total distance sum are minimized. We assume there is restriction on the capacity of the DCs. Each customer has a demand b_j (j=1,2,…,n), and each DC has a service capacity a_i (i=1,2,…,m). The cLAP can be illustrated in Figure 13

The capacitated location-allocation problem can be modelled as a mixed integer programming model as follows:

Indices

i: index of DC (i=1,2,…,m)
j: index of customer (j=1,2,…,n)

Parameters

m: total number of DCs
n: total number of customers
$F_i=(x_i, y_i)$: ith DC i=1,2,…,m
$C_j=(u_j, v_j)$: jth customer j=1,2,…,n
a_i: capacity of ith DC
b_j: demand of jth customer
$t(Fi,Cj)$: the Euclidean distance from the location of DC i,(x_i, y_i) to the location of a customer at fixed point j, (u_j, v_j)

$$t(F_i, C_j) = \sqrt{(x_i - u_j)^2 + (y_i - v_j)^2}$$

Decision Variables

z_{ij}: 0-1 decision allocation variables
z_{ij}=1, representing the jth customer is served by ith DC. Otherwise 0
$F_i = (x_i, y_i)$: unknown location of the ith DC

$$\min \quad f(F, \mathbf{z}) = \sum_{i=1}^{m} \sum_{j=1}^{n} t(F_i, C_j) \cdot z_{ij} \tag{6.1}$$

$$\text{s.t.} \quad g_i(\mathbf{z}) = \sum_{j=1}^{n} b_j z_{ij} \le a_i, \; i = 1,2,\cdots,m \tag{6.2}$$

$$g_{m+j}(\mathbf{z}) = \sum_{i=1}^{m} z_{ij} = 1, \quad j = 1,2,\cdots,n \tag{6.3}$$

$$z_{ij} = 0 \text{ or } 1, \quad i = 1,2,\cdots,m, \; j = 1,2,\cdots,n \tag{6.4}$$

$t(F_i, C_j)$ is the Euclidean distance from the location (coordinates) of DC i, (x_i, y_i) (the decision variable), to the location of a customer at fixed point j, (u_i, v_i); and z_{ij} is the 0-1 allocation variable, z_{ij}=1 representing that customer j is served by DC i and z_{ij}=0 otherwise (i=1,2,…,m, j=1,2,…n). Constraints (6.2) reflect that the service capacity of each DC should not be exceeded. Constraints (6.3) reflect that every customer should be served by only one DC.

6.2. GA Approach for cLAP

Cooper proposed a heuristic method called alternative location-allocation (ALA) to solve this problem (Cooper, 1963). But this method heavily depends on the selection of initial locations of the DCs and optimality is not guaranteed yet. Recently, evolutionary computing has been shown powerful in global and hard-solving optimization problem such as genetic algorithm (GA) and evolutionary strategy (ES). We choose to employ a genetic algorithm to solve cLAP and find better solutions than heuristic approaches. In this problem, there are two kinds of decision variables. One is continuous location value and another is zero-one allocation variables (Gong *et al.*, 1995). So we only use GA to search the best locations of DCs, otherwise we use construction heuristic to allocate customers to DCs.

Genetic Representation: In continuous location problems, a binary representation may result in locating two DCs which are very close to each other. As in (Gen & Cheng, 1997; Gong *et al.*, 1996) we use a real number representation where a chromosome consists of m (x, y) pairs representing the sites of DCs to be located, and p is the number of DCs. For instance this is represented as follows:

$$\text{chromosome } v = [(x_1, y_1), (x_2, y_2), \cdots, (x_i, y_i), \cdots, (x_m, y_m)]$$

where the coordinate (x_i, y_i) denotes the location of the ith DC, $i = 1, \ldots, m$.

Evaluation: Once a chromosome is given, the locations of DCs of this chromosome are fixed and the Euclidian distance $t(F_i,C_j)$ between location of a DC i, (x_i,y_i) and location of a customer at fixed point j, (u_j,v_j). We define the fitness of this chromosome as the objective function of the optimal allocation of customers to the known DC. This problem is a general assignment problem shown as follows:

$$\begin{aligned} eval(v_k) &= \frac{1}{f(F, \boldsymbol{z})} \\ &= \frac{1}{\sum_{i=1}^{m} \sum_{j=1}^{n} t(F_i, C_j) \cdot z_{ij}} \end{aligned}$$

Construction Heuristic (***Nearest Neighbour Algorithm***) for Allocation Customers: Nearest Neighbour algorithm (NNA) is to allocate customers based on customer demand and minimum distance between customers so that NNA preserves each DC does not exceed its capacity by customer demand.

Crossover operator: Crossover depends on how to choose the parents and how parents produce their children. GA usually selects parents by spinning the weight roulette wheel rule and ES such as ES-(μ+λ) which aims at numerical optimization give every member in population pool with equal probability to produce child and let evolution be done by selection procedure. We adopt the idea of ES-(μ+λ) to select parents to produce child. Suppose two chromosomes:

$$v_1 = [(x_1^1, y_1^1), (x_2^1, y_2^1), \cdots, (x_i^1, y_i^1), \cdots, (x_m^1, y_m^1)]$$
$$v_2 = [(x_1^2, y_1^2), (x_2^2, y_2^2), \cdots, (x_i^2, y_i^2), \cdots, (x_m^2, y_m^2)]$$

We allow them to produce only one child:

$$\bar{v} = \left[\left(\overline{x_1}, \overline{y_1}\right), \left(\overline{x_2}, \overline{y_2}\right), \cdots, \left(\overline{x_i}, \overline{y_i}\right), \cdots, \left(\overline{x_m}, \overline{y_m}\right)\right]$$

where genes in the chromosomes of children are decided by following equations:

$$\bar{x}_i^1 = \alpha_i \cdot x_i^1 + (1-\alpha_i) \cdot x_i^2 \ , \qquad \bar{y}_i^1 = \alpha_i \cdot y_i^1 + (1-\alpha_i) \cdot y_i^2$$
$$\bar{x}_i^2 = (1-\alpha_i) \cdot x_i^1 + \alpha_i \cdot x_i^2 \ , \qquad \bar{y}_i^2 = (1-\alpha_i) \cdot y_i^1 + \alpha_i \cdot y_i^2$$

where α_i is a random numbers in (0, 1) (i=1,2,…,m).

Mutation operator: Mutation is very important to introduce new gene and prevent the premature of genetic process. For any chromosome in a population an associated real value, $0 \le \rho \le 1$, is generated randomly. If ρ less than the predefined mutation threshold, p_M, mutation operator is applied to this chromosome. Considering the characteristics of the original problem, we suggest two kinds of mutation operators. One is *subtle mutation* which only gives a small random disturbance to a chromosome to form a new child chromosome. Another is *violent mutation* which give a new child chromosome totally randomly the same as the initialization. We also use the above two kinds of mutation operators alternatively in the evolutionary process.

Suppose the chromosome to be mutated is as follows:

$$v_k = \left[(x_1^k, y_1^k), (x_2^k, y_2^k), \cdots, (x_i^k, y_i^k), \cdots, (x_m^k, y_m^k)\right]$$

Then the child produced by the *subtle mutation* is as follows:

$$v' = \left[(x_1', y_1'), (x_2', y_2'), \cdots, (x_i', y_i'), \cdots, (x_m', y_m')\right]$$
$$x_i' = x_i^k + \text{random value in } [-\varepsilon, \varepsilon]$$
$$y_i' = y_i^k + \text{random value in } [-\varepsilon, \varepsilon]$$

where ε is a small positive real number.

The child produced by *violent mutation* is as follows:

$$v' = \left[(x_1', y_1'), (x_2', y_2'), \cdots, (x_i', y_i'), \cdots, (x_m', y_m')\right]$$
$$x_i' = \text{random value in } [x_{\min}, x_{\max}]$$
$$y_i' = \text{random value in } [y_{\min}, y_{\max}]$$

Selection: ES-(μ+λ) selection is adopted to select the better individuals among parents and their children to form the next generation. However, the strategy usually leads to degeneration of the genetic process. In order to avoid this degeneration, a new selection strategy called relative prohibition is suggested.

Give two positive parameters α and γ, the neighbourhood for a chromosome v_k is defined as follows:

$$\Omega(v_k, \alpha, \gamma) \cong \{ s \,|\, \|s - s_k\| \le \gamma, D(s_k) - D(s) < \alpha, s \in R^{2m} \}$$

In selection process, once s_k is selected into the next generation, any chromosome falling within its neighbourhood is prohibited from selection. The value of γ defines the neighbourhood of s_k in terms of location, which is used to avoid selecting individuals with very small difference in location. The value of α defines the neighbourhood of s_k in terms of fitness, which is used to avoid selecting individuals with very small difference in fitness.

6.3. Experiments and Discussion

In this Section, in order to test the effectiveness of the proposed method, we use it to solving example. We consider an example consists of 3 DCs and 16 customers. We use same data set in the literature (Gong *et al.*, 1996). The demand of each customer and the coordinates of each customer and demand of customer are shown in Table 6. We assume that the capacity of each DC in Table 7. The result of this experiment is summarized in Table 8, Table 9 and Table 10. The GA parameters for this problem were set as follows:

$popSize = 100$, $maxGgen = 1000$, $p_C = 0.5$, $p_M = 0.5$
$[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \rightarrow [0, 12000]\times[0,10000]$

Table 6. Coordinates and Demands of customers

j	(u_j,v_j)	b_j	j	(u_j,v_j)	b_j
1	(0,0)	100	9	(4000,10000)	100
2	(0,500)	100	10	(5000,1000)	100
3	(1000,4000)	100	11	(7000,6000)	100
4	(1000,9000)	100	12	(8000,1000)	100
5	(2000,2000)	100	13	(8000,10000)	100
6	(2000,6000)	100	14	(10000,7000)	100
7	(4000,4000)	100	15	(11000,2000)	100
8	(4000,8000)	100	16	(12000,10000)	100

Table 7. Capacity of DCs

i	a_1	a_2	a_3
1	1800		
2	1000	1000	
3	800	600	600

We found the best result among 100 times running and then compared with ALA (Cooper, 1963, Gong *et al.*, 1996), EP (Gong *et al.*, 1996) and proposed GA in Figure 14.

Table 8. Best solution by ALA (Cooper, 1963, Gong *et al.*, 1996)

i	a_i	Location	Allocation	Distance
1	a_1=18000	(1937.50,5312.24)	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16	76525.85
2	a_1=1000	(4000.00,7750.00)	2,3,4,5,6,8,9,13,16	62920.87
	a_2=1000	(5875.00,2875.00)	1,7,10,11,12,14,15	
3	a_1=800	(7750.00,2500.00)	10,11,12,15	47146.58
	a_2=600	(1500.00,3500.00)	1,2,3,5,6,7	
	a_3=600	(6500.00,9000.00)	4,8,9,13,14,16	

Table 9. Best solution by EP (Gong *et al.*, 1996)

i	a_i	Location	Allocation	Distance
1	a_1=18000	(1937.50,5312.24)	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16	76525.85
2	a_1=1000	(2300.00,4900.00)	1,2,3,4,5,6,7,8,9,10	55298.45
	a_2=1000	(9333.33,6000.00)	11,12,13,16,45,16,	
3	a_1=800	(1428.57,4285.71)	1,2,3,4,5,6,7	43670.80
	a_2=600	(8000.00,1333.33)	10,12,15	
	a_3=600	(7500.00,8500.00)	8,9,11,13,14,16	

Table 10. Best solution by proposed GA

i	a_i	Location	Allocation	Distance
1	a_1=18000	(1937.50,5312.24)	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16	76525.85
2	a_1=1000	(2300.00,4900.00)	1,2,3,4,5,6,7,8,9,10	55298.45
	a_2=1000	(9333.33,6000.00)	11,12,13,16,45,16,	
3	a_1=800	(2731.00,8466.00)	4,6,8,9	43324.09
	a_2=600	(9281.00,6538.00)	11,12,13,14,15,16	
	a_3=600	(1739.00,1961.00)	1,2,3,5,7,10	

We consider more realistic example in order to test the effectiveness of the proposed method. We consider an example, consists of 5 DCs and 100 customers. We assume that the capacity of each DC is equal to 150. The result of this experiment is summarized in Figure 11. The GA parameters for this problem were set as follows:

$popSize$ =100, $maxGgen$ = 1000, p_C= 0.5and p_M= 0.5
$[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \rightarrow [35,752]\times[29,517]$

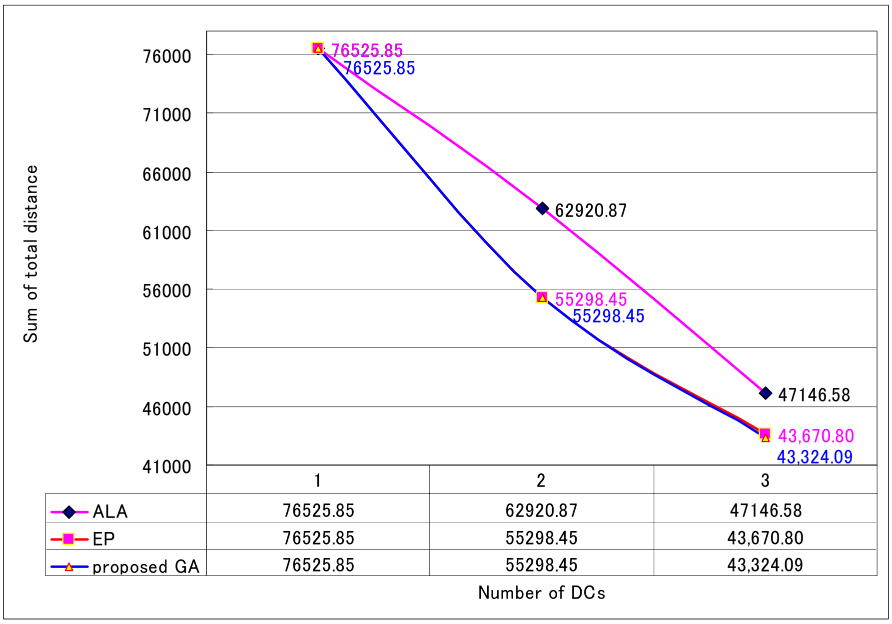


Figure 14. Comparison result

Distribution centre location problem is an extension of the location allocation problem, which is a more realistic model. It is a multimodal optimization problem and traditional methods are not efficient to find global or near global solutions. We adopted a hybrid genetic algorithm (hGA) method to find the global or near global optimal solution for this problem. Although the alternative location-allocation (ALA) method has widely used to solve this kind of problem, it heavily depends on the selection of initial location of DCs and the optimality is not guaranteed. The proposed hGA and constructive heuristic algorithm are solved location-allocation problem. The numerical experiments is showed the effectiveness of GA in finding better solution which maybe regarded as near global solution.

7. PRACTICAL LOGISTICS APPLICATION

In its logistics network, there are 6 distribution centres (DCs) located in city of Saga, Fukuoka, Okayama, Shiga, Saitama, and Tochigi in Japan, in which the products and materials are deposited, then transported from one to another, and waiting for delivering to customers there. Figure 15 shows the current delivery network among DCs. This study mainly aims at delivery cost reduction by optimizing the vehicle routing in logistics network among these DCs.

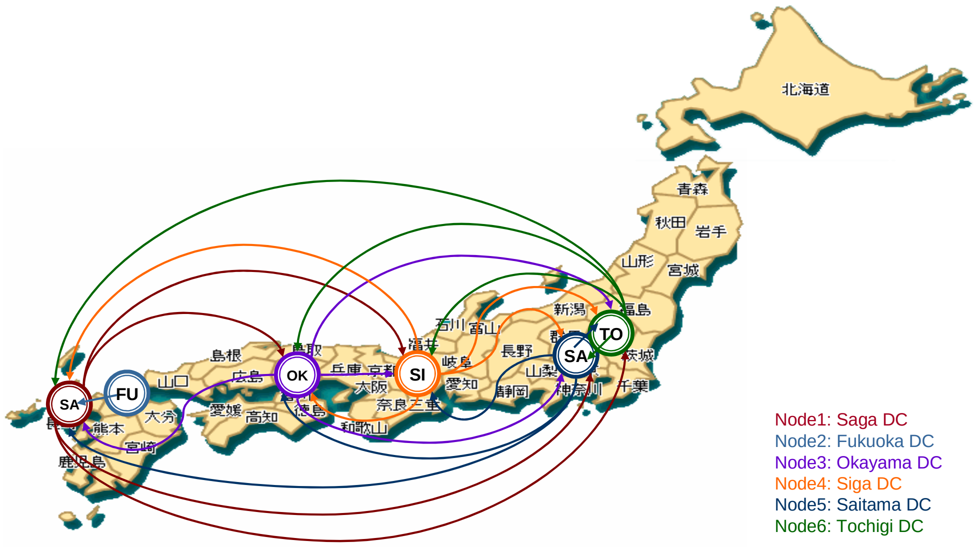


Figure 15. The current logistics network among DCs

According to the transportation contract between Nippon Steel Transportation and some third party logistics providers (3PLs), if the destination place of a truck is different from the start place, the company should pay 0.3 times more than the normal price. For example, the normal transportation price from DC 1 to DC 2 is 2,600 yen per ton. If we rent a truck to deliver 10 ton of goods from DC 1 to DC 2, and also the same quantity of goods needs to be delivered when it comes back. Then the total delivery cost is

$$2*10*2,600 = 52,000 \text{ (yen)}$$

If there are no goods need to be delivered from place 2 to place 1, it means the truck should ends its travel at place, so the total delivery cost becomes

$$1.3*10*2,600=33,800 \text{ (yen)}$$

This operating of vacant vehicle results in an inefficient element in the logistics network. In this case we studied, a truck started from the DC1 to DC4 with 500.9t goods, would deliver just 25.8t when return to DC1. The transform power of 475.1t is made little use on their way back.

We consider this problem as the minimum cost flow (MCF) problem. The objective of this problem is to determine the policy of delivery flow with minimum cost through a network to satisfy supply and demand requirements. MCF problem is defined as a connected graph $G=(V, E)$, with a given set of arcs V and a set of nodes E.

7.1. Mathematical Formulation

First, we give some assumptions based on which we formulate this problem:

Assumption 1: There is only one kind of goods delivered in the network. The goods from different DCs are considered to be consubstantial.

Assumption 2: The sum of delivery flow into each DC must equal the sum of delivery flow out of the DC (flow conservation requirement). This assumption implicates constrain of vehicle.

Assumption 3: The operating of vacant vehicle can be taken for dummy delivery flow, the unit cost of which is equal to 0.3 times of the normal.

Assumption 4: The amount of deliveries to DC i is not smaller than the amount of demand d_i, the amount of deliveries from DC i is not more than the amount of supply b_i.

And then, we define the notation used in this study as following:

Indices

i, j, k: index of DC (i, j, k =1, 2,…, N)*Parameters*
c_{ij}: delivery cost between DC i and DC j
d_i: amount of demand of DC i
b_i: amount of supply of DC i y_{ij}: dummy delivery flow from DC i to DC j
suc(i): set of all successors of DC i
pre(i): set of all predecessors of DC I
N: total number of DC (in this study N =6)
E: set of nodes
V: set of arcs
S_i: set of nodes adjacent to node i

Decision Variables

x_{ij}: delivery amount from DC i to DC j
y_{ij}: dummy delivery flow from DC i to DC j

In a MCF problem, the total delivery cost is usually used as objective function. In this study, in order to reduce the vacant vehicle operating, we add a term of cost of dummy delivery flow to the mathematical model. Based on the assumptions mentioned above, we formulate this problem as follows:

$$\begin{aligned}
\min \quad & z = f_1(\boldsymbol{x}) + f_2(\boldsymbol{x}) \\
& = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(c_{ij}x_{ij} + 0.3\cdot c_{ji}y_{ji}\right) \\
\text{s.t.} \quad & \sum_{j\in \mathrm{Succ}(i)}\left(x_{ij}+y_{ij}\right) - \sum_{k\in \mathrm{Pred}(i)}\left(x_{ki}+y_{ki}\right) = 0, \quad \forall i \\
& \sum_{k\in \mathrm{Pred}(i)} t_{ki}x_{ki} \ge d_i, \quad \forall i \\
& \sum_{j\in \mathrm{Succ}(i)} t_{ij}x_{ij} \le b_i, \quad \forall i \\
& x_{ij} \ge l_{ij}, \quad \forall i, j
\end{aligned}$$

The objective function (7.1) is to minimize the total delivery cost c_T in the logistics network. Constraint (7.2) makes sure the sum of delivery amount into a DC must equal the sum of delivery amount flows out of the DC. Constraint (7.3) shows the total delivery amount to DC i is more than or equal to the amount of its demand d_i. Constraint (7.4) shows the total delivery amount out of DC i is smaller than or equal to the amount of its supply b_i.

7.2. Proposed GA Approach

Genetic Representation: Prior to the application of GA, we need to design suitable chromosomes representing the candidate solutions. We here adopted the priority-based encoding method, which can escape the repair mechanisms in the search process of GA, had been developed by Cheng and Gen (2000). The priority-based encoding method is an indirect approach: encode some guiding information for constructing a path, but a path itself, in a chromosome. In this method, a gene contains two kinds of information: the locus, the position of a gene located within the structure of a chromosome, and the allele, the value taken by the gene. The position of a gene is used to represent node ID in a graph and its value is used to represent the priority of the node for constructing a path among candidates.

To find out transportation routes among DCs, only one arc with the highest priority within a chromosome is selected at each time. In this way, a path can be uniquely determined from this encoding. Figure 16 shows the representation of the routing among 6 DCs, and the original data are shown in Table 11, and we can get an alternative solution of the problem from this chromosome.

node ID:	1	2	3	4	5	6
priority $v_k(i)$:	6	5	4	2	1	3

Figure 16. Sample representation of chromosome by priority-based encoding

Table 11. Supply and demand of each DC (unit: ton)

DC	d_i [t]	b_i [t]
1	477.2	1392.4
2	0.0	251.7
3	746.1	37.2
4	1016.5	66.3
5	1231.0	235.1
6	160.7	1648.8

Table 12. Data of sample chromosome

#	Path	Flow on the path	Dummy flow on the path
1	{1-5-6-4-3-1}	37.2	0.0
2	{1-5-6-4-1}	29.1	0.0
3	{1-5-6-1}	94.4	0.0
4	{1-5-1}	74.4	0.0
5	{1-5-1}	995.9	995.9
6	{1-4-1}	161.4	161.4
7	{2-1-2}	241.9	241.9
8	{2-4-2}	9.8	9.8
9	{6-4-6}	779.0	779.0
10	{6-3-6}	708.9	708.9

Crossover operator: Generally, crossover is used as the primary operator and the performance of a genetic algorithm is affected greatly by it. It generates offspring combined both parents' features by exchange the information of parents. Many crossover methods have been prompted, such as one-cut-point crossover, multi-cut-point crossover and uniform crossover. In this study, we adopt Order crossover (OX), which can escape from the complex repairing procedure.

***Mutation operator*:** Used as a background operator, mutation creates new individual to increase the variability of population by modifying one or more of the gene values of existing individuals. We use swap mutation operation, which select two positions of an individual randomly, and change the alleles they contain.

7.3. Case Study

The original data of this case are the unit delivery cost and delivery amount in current logistics network, which are shown in Table 13 and Table 14.

To examine the effectivity of the prompted approach, we apply the formulation and pGA approach on the case to optimize transformation routes. In order to apply the GA approach, we using the following parameters: *popSize*=10, *MaxGen*=100, P_C=0.4and P_M=0.4. The best solution we got by using this pGA approach is given in Table 15.

Table 13. Unit delivery cost between each DC (unit: yen)

DC	1	2	3	4	5	6
1	100000	2600	6400	8500	15600	16300
2	2600	100000	9200	7500	14600	15600
3	6400	9200	100000	6000	13200	10700
4	8500	7500	6000	100000	9200	9100
5	15600	14600	13200	9200	100000	5800
6	16300	15600	10700	9100	5800	100000

Table 14. Delivery amounts between each DC in current logistics network (unit: ton)

DC	1	2	3	4	5	6
1	0.0	0.0	563.2	500.9	259.5	68.8
2	251.7	0.0	0.0	0.0	0.0	0.0
3	21.1	0.0	0.0	8.3	5.0	2.8
4	25.8	0.0	20.1	0.0	15.7	4.7
5	36.7	0.0	26.6	87.4	0.0	84.4
6	141.9	0.0	136.2	419.9	950.8	0.0

Table 15. Delivery amounts between each DC in optimized logistics network (unit: ton)

DC	1	2	3	4	5	6
1	0.0	0.0	542.1	475.1	222.8	0.0
2	251.7	0.0	0.0	0.0	0.0	0.0
3	708.9	0.0	0.0	0.0	0.0	0.0
4	206.3	251.7	11.8	0.0	0.0	492.2
5	0.0	0.0	21.6	71.7	0.0	995.9
6	73.1	0.0	133.4	415.2	866.4	0.0

To compare the two results, we calculate the total based on the objective function basing the data above, and then we calculate the improvement.

Table 16. Comparison result of case studied

Total delivery cost (current) c_T (yen)	37,040,120
Total delivery cost (optimize) c^*_T (yen)	35,337,120
Improvement (=(c_T-c^*_T)/ c_T)	4.8193%

8. AUTOMATED GUIDED VEHICLES DISPATCHING

Automated material handling has been called the key to integrated manufacturing. An integrated system is useless without a fully integrated, automated material handling system. In the manufacturing environment, there are many automated material handling possibilities. Currently, automated guided vehicles systems (AGV Systems), which include automated guided vehicles (AGVs), are the state–of–the–art, and are often used to facilitate automatic storage and retrieval systems (AS/RS).

In this study, we focus on the simultaneous scheduling and routing of AGVs in a flexible manufacturing system (FMS). A FMS environment requires a flexible and adaptable material handling system. AGVs provide such a system. An AGV is a material handling equipment that travels on a network of guide paths. The FMS is composed of various cells, also called working stations (or machine), each with a specific operation such as milling, washing, or assembly. Each cell is connected to the guide path network by a pickup/delivery (P/D) point where pallets are transferred from/to the AGVs. Pallets of products are moved between the cells by the AGVs.

8.1. Network Modelling for AGV Dispatching

In this paper, the problem is to dispatch AGVs for transports the product between different machines in a FMS. At first stage, we model the problem by using network structure.

Assumptions considered in this paper are as follows,

For FMS scheduling:

1) In a FMS, n jobs are to be scheduled on m machines.
2) The ith job has n_i operations that have to be processed.
3) Each machine processes only one operation at a time.
4) The set-up time for the operations is sequence-independent and is included in the processing time.

For AGV dispatching:

1) Each machine is connected to the guide path network by a pick-up/delivery (P/D) station where pallets are transferred from/to the AGVs.
2) The guide path is composed of aisle segments on which the vehicles are assumed to travel at a constant speed.
3) As many vehicles travel on the guide path simultaneously, collisions be avoided by hardware, not be considered in this paper.

Subject to the constraints that,

For FMS scheduling:

1) The operation sequence for each job is prescribed;
2) Each machine can process only one operation at a time;
3) Each AGV can transport only one kind of products at a time.

For AGV dispatching:

1) AGVs only carry one kind of products at same time.
2) The vehicles just can travel forward, not backward.

The objective function is minimizing the following two criteria:

1) Time required to complete all jobs (*i.e.* makespan): t_{MS}
2) Number of AGVs: n_{AGV}

The notation used in this paper is summarized in the following:

Indices

i, i' : index of jobs, $i, i' = 1,2,\ldots,n$;

j, j' : index of processes, $j, j' = 1,2,\ldots,n_i$;

Parameters

n : total number of jobs;

m : total number of machines;

n_i : total number of operations of job j;

o_{ij} : the j-th operation of job i;

p_{ij} : processing time of operation o_{ij};

M_{ij} : machine assigned for operation o_{ij}

T_{ij} : transition task for operation o_{ij};

t_{ij} : transition time from $M_{i,j-1}$ to M_{ij};

The objective of this network problem assigns all of tasks to several AGVs, and gives the priority of each task to make the AGV routing sequence with minimizing time required to complete all jobs (i.e. makespan)

Decision variables

x_{ij} : assigned AGV number for task T_{ij}

t_{ij}^S : starting time of task T_{ij};

c_{ij}^S : starting time of operation o_{ij};

The problem can be formulated as follows:

$$\min \quad t_{\mathrm{MS}} = \max_i \left\{ t_{i,n_i}^S + t_{M_{i,n_i},0} \right\} \tag{8.1}$$

$$\text{s.t.} \quad c_{ij}^S - c_{i,j-1}^S \ge p_{i,j-1} + t_{ij}, \quad \forall i, j = 2,\ldots,n_i \tag{8.2}$$

$$\left(c_{ij}^S - c_{i'j'}^S - p_{i'j'} + \Gamma\left|M_{ij} - M_{i'j'}\right| \ge 0\right) \vee \left(c_{i'j'}^S - c_{ij}^S - p_{ij} + \Gamma\left|M_{ij} - M_{i'j'}\right| \ge 0\right), \quad \forall (i,j),(i',j') \tag{8.3}$$

$$\left(t_{ij}^S - t_{i'j'}^S - t_{i'j'} + \Gamma\left|x_{ij} - x_{i'j'}\right| \ge 0\right) \vee \left(t_{i'j'}^S - t_{ij}^S - t_{ij} + \Gamma\left|x_{ij} - x_{i'j'}\right| \ge 0\right), \quad \forall (i,j),(i',j') \tag{8.4}$$

$$\left(t_{i,n_i}^S - t_{i'j'}^S - t_{i'j'} + \Gamma\left|x_{ij} - x_{i'j'}\right| \ge 0\right) \vee \left(t_{i'j'}^S - t_{i,n_i}^S - t_i + \Gamma\left|x_{ij} - x_{i'j'}\right| \ge 0\right), \quad \forall (i,n_i),(i',j') \tag{8.5}$$

$$c_{ij}^S \ge t_{i,j+1}^S - p_{ij} \tag{8.6}$$

$$x_{ij} \ge 0, \quad \forall i, j \tag{8.7}$$

$$t_{ij}^S \ge 0, \quad \forall i, j \tag{8.8}$$

where Γ is a very large number, and t_i is the transition time for pickup point of machine $M_{i,ni}$ to delivery point of Loading/ Unloading. Inequality (8.2) describes the operation precedence constraints. In inequities (8.3), (8.4) and (8.5), since one or the other constraint must hold, it is called disjunctive constraint. It represents the operation un-overlapping constraint (Inequality 8.3) and the AGV non-overlapping constraint (Inequality 8.4, 8.5).

8.2. Priority-based GA

We firstly give a priority-based encoding method that is an indirect approach: encode some guiding information for constructing a sequence of all tasks. As it is known, a gene in a chromosome is characterized by two factors: locus, *i.e.*, the position of gene located within the structure of chromosome, and allele, *i.e.*, the value the gene takes. In this encoding method, the position of a gene is used to represent task ID and its value is used to represent the priority of the task for constructing a sequence among candidates. A feasible sequence can be uniquely determined from this encoding with considering operation precedence constrain. An example of generated chromosome and its decoded path is shown as following:

Task ID :	1	2	3	4	5	6	7	8	9
Priority :	1	5	7	2	6	8	3	9	4

$$T_{11} \to T_{12} \to T_{13} \to T_{14} \to T_{21} \to T_{22} \to T_{31} \to T_{32} \to T_{33}$$

At the beginning, we try to find a task for the position next to source node s. Task T_{11}, T_{21} and T_{31} (Task ID: 1, 2 and 3) are eligible for the position, which can be easily fixed according to adjacent relation among tasks. The priorities of them are 1, 5 and 7, respectively. The node 1 has the highest priority and is put into the task sequence. The possible tasks next to task T_{11}, is task T_{12} (Task ID: 4), and unselected task T_{21} and T_{31} (Task ID: 2 and 3). Because node 4 has the largest priority value, it is put into the task sequence. Then we form the set of tasks available for next position and select the one with the highest priority among them. Repeat these steps until all of tasks be selected,

$$T_{11} \to T_{12} \to T_{13} \to T_{14} \to T_{21} \to T_{22} \to T_{31} \to T_{32} \to T_{33}$$

After generated task sequence, we secondly separate tasks to several groups for assigning different AGVs. First, separate tasks with a separate point in which the task is the final transport of job i form pickup point of operation $O_{i,\,ni}$ to delivery point of Loading/Unloading. Afterward, unite the task groups which finished time of a group is faster than the starting time of another group. The particular is introduced in next subsection. An example of grouping is shown as following:

$$\begin{aligned}&\text{AGV}1:\ T_{11} \to T_{12} \to T_{13} \to T_{14}\\ &\text{AGV}2:\ T_{21} \to T_{22}\\ &\text{AGV}3:\ T_{31} \to T_{32} \to T_{33}\end{aligned}$$

8.3. Case Study

For evaluating the efficiency of the AGV dispatching algorithm suggested in a case study, a simulation program was developed by using Java on Pentium 4 processor (3.2-GHz clock). The problem was used by Yang (2001) and Kim *et. al.* (2004). GA parameter settings were taken as follows: population size, *popSize* =20; maximum generation, *maxGen*=1000; crossover probability, p_C=0.70; mutation probability, p_M =0.50; immigration rate, μ = 0.15.

Table 17. Job Requirements of Example (Kim *et. al.*, 2004)

O_{ij}	M_{ij}				p_{ij}			
J_i \ P_j	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4
J_1	1	2	1	-	80	120	60	-
J_2	2	1	-	-	100	60	-	-
J_3	5	3	3	-	70	100	70	-
J_4	5	3	2	2	70	100	100	40
J_5	4	2	-	-	90	40	-	-
J_6	4	4	1	2	90	70	60	40
J_7	1	3	-	-	80	70	-	-
J_8	5	4	5	4	70	70	70	80
J_9	5	4	1	-	70	70	60	-
J_{10}	5	1	3	-	70	60	70	-

In a case study of FMS, 10 jobs are to be scheduled on 5 machines. The maximum number process for the operations is 4. Table 17 gives the assigned machine numbers and process time. And Table 18 gives the transition time among pickup points and delivery points.

Table 18. Transition Time between Pickup Point *u* and Delivery Point *v*

t_{uv} / c_{uv}	Loading / Unloading	M_1	M_2	M_3	M_4	M_5
Loading / Unloading	1 / 1	1 / 7	8 / 13	14 / 18	16 / 23	18 / 20
M_1	13 / 18	3 / 3	2 / 9	8 / 14	10 / 19	13 / 18
M_2	18 / 22	22 / 28	2 / 2	2 / 7	4 / 12	12 / 18
M_3	13 / 11	17 / 22	24 / 29	1 / 1	1 / 6	7 / 11
M_4	8 / 14	12 / 20	18 / 26	24 / 29	3 / 3	2 / 10
M_5	5 / 7	9 / 12	15 / 18	19 / 23	23 / 28	2 / 2

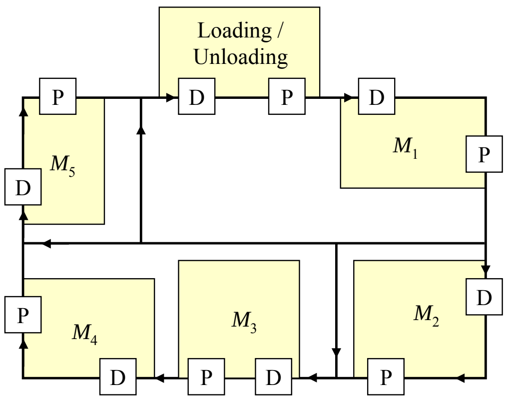


Figure 17. Layout of facility (P: Pickup Point, D: Delivery Point)

Depending on (Naso and Turchiano, 2005), we give a layout of facility for the experiment in Figure 17. We can draw a network (as Figure 18) depend on the precedence constraints among tasks $\{T_{ij}\}$ of Example 2. The best result of Example 2 is shown as follows and final time required to complete all jobs (*i.e.* makespan) is 574 and 4 AGVs are used. Figure 19 shows the result on Gantt chart.

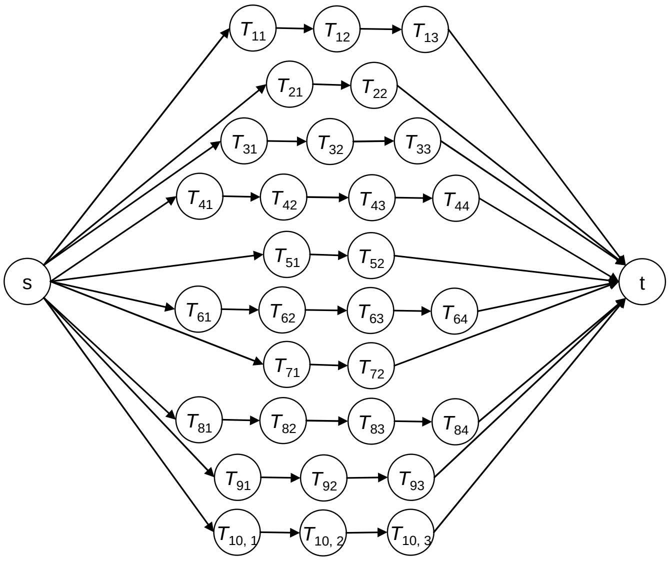
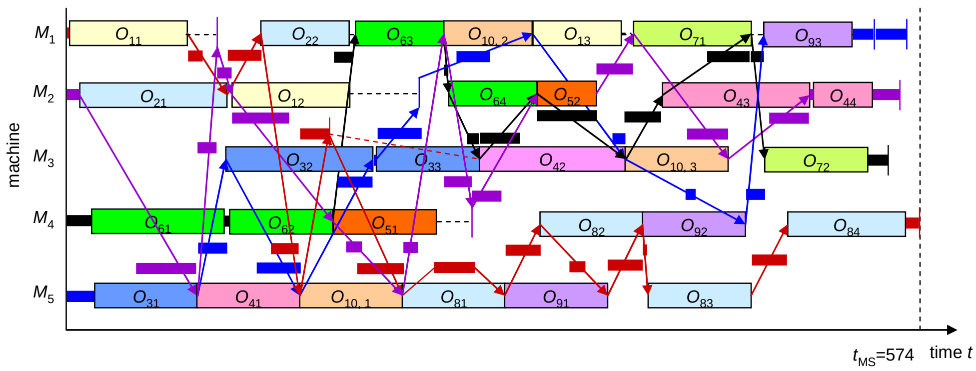
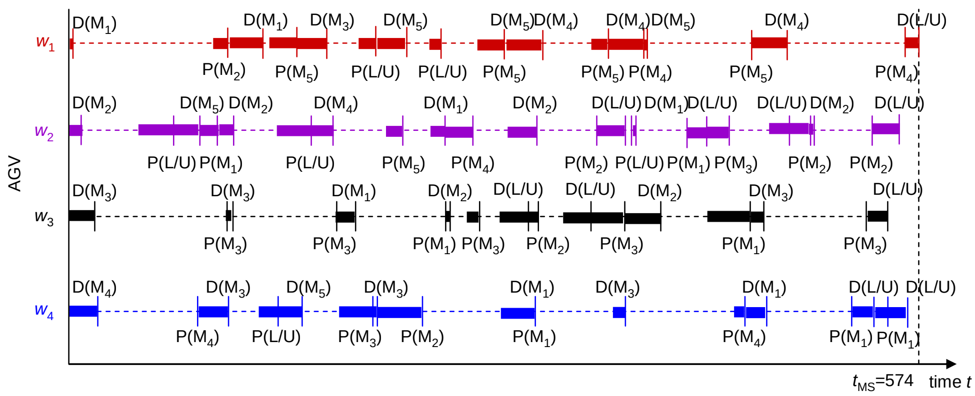


Figure 18. Illustration of the network structure of Example



(a) Based on operations processing



(b) Based on AGVs dispatching

Figure 19. Gantt chart of the schedule of Example 2 with considering AGVs routing

9. CONCLUSION

The use of evolutionary techniques in the logistics networks design has been growing the last few decades due to the fact that logistics networks design problem is NP hard. This paper examined recent developments in the field of evolutionary optimization for logistics problems in various areas. A wide range of problem is covered as follows: first, we applied the hybrid Genetic Algorithm (hGA) approach for solving Fixed Charge Transportation Problem (fcTP). We have done several numerical experiments and compared the results with those of simple

GA. The proposed approach is more effective in larger size than benchmark test problems. Second, we gave the several resent GA approach for solving Multistage Logistic Network Problems. Third, we introduced Vehicle Routing Problem (VRP) and variants of VRP. We apply the priority-based Genetic Algorithm (pGA) approach for solving Multi-depot vehicle routing problem with time windows (mdVRP-tw). Fourth, we discussed distribution centre location problem of distribution system which consists of customers and a number of distribution centres to be located. We adopted a hybrid genetic algorithm (hGA) method to find the global or near global optimal solution for location-allocation problem. Fifth, as a case study model, practical logistics applications to find the optimal routing was introduced. Last, we modelled an automated guided vehicles (AGV) system by using network structure. This network model of an AGV dispatching has simplexes decision variables with considering most AGV problem's constraints. Furthermore, we applied an evolutionary approach for solving this problem with minimizing time required to complete all jobs (i.e., makespan).

ACKNOWLEDGEMENTS

This work was partly supported by Waseda University Grant for Special Research Projects 2004, Japanese International Communication Foundation, and the Ministry of Education, Science and Culture, the Japanese Government: Grant-in-Aid for Scientific Research (No.17510138).

REFERENCES

Adlakha, V. and Kowalski, K. (2003), Simple heuristic algorithm for the solution of small fixed-charge problems, *Omega, Int. J. Mgmt. Sci.*, **31(3)**, 205-212.

Altiparmak, F., Gen, M. and Lin, L. (2004), A Priority-based Genetic Algorithm for Supply Chain Design, *Proc. of the 33rd Inter. Conf. on Computers and Industrial Engineering*, Jeju, Korea, Mar. 25-27, 2004.

Altiparmaka, F., Gen, M., Lin, L. and Paksoy, T., A Genetic Algorithm Approach for Multi-objective Optimization of Supply Chain Networks, *Computers & Industrial Engineering*, Accepted.

Angelelli, E. and Mansini, R. (2001), The Vehicle Routing Problem with Time Windows and Simultaneous Pick-up and Delivery, *Proc. of the Triennnal Symposium on Transportation Analysis*, 581-586.

Azevedo, A. L. and Sousa, J. P. (2000), Order planning for networked made-to-order enterprises- a case study, *Journal of Operational Research Society*, **51**, 1116-1127.

Bazaraa, M., Jarvis, J. and Sherali, H. (1993), *Linear Programming and Network Flows*, 2nd ed., John Wiley & Sons, New York.

Cao, B. and Uebe, G. (1995), Solving Transportation Problems with Nonlinear Side Constraints with Tabu Search, *Computer & Ops. Res.* **22(6)**, 593-603.

Chopra, S. and Meindl, P. (2004), *Supply Chain Management – Strategy, Planning, and Operation*, 2^{nd} ed., Pearson Education, Inc., New Jersey.

Cooper, L. (1963), Location-allocation problems, *Operations Research*, **11**, 331–343.

Gamsworld [Online]. Available : http://www.gamsworld.org

Garey, M. and Johnson, D. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman.

Gen, M. and Syarif, A. (2005), Hybrid genetic algorithm for multi-time period production/distribution planning, *Computers and Industrial Engineering,* **48(4)**, 799-810.

Gen, M., Altiparamk, F. and Lin, L., A Genetic Algorithm for Two-stage Transportation Problem using Priority-based Encoding, *OR Spectrum*, Accepted.

Gen. M. and Lin, L. (2004), Multiobjective hybrid genetic algorithm for bicriteria network design problem, *Proc. of Asia Pacific Symposium on Intelligent and Evolutionary Systems*, **8**, 73-82.

Goldberg, D. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.

Gong, D., Yamazaki, G. and Gen, M. (1996), Evolutionary program for optimal design of material distribution system, *Proc. of IEEE International Conference on Evolutionary Computation*, 131-134.

Gong, D., Xu, W., Gen, M. and Yamazaki, G. (1995), Hybrid evolutionary method for obstacle location-allocation, *Computers and Industrial Engineering*, **29**, 525-530.

Hitchcock, F. (1941), The distribution of a product from several sources to numerous locations, *Journal of Mathematical Physics*, **20**, 224-230.

ITWM. [Online]. Available: http://www.itwm.fhg.de/en/opt_Schwerpunkte/ulmain/

Jayaraman, V. and Ross, A. (2003), A simulated annealing methodology to distribution network design and management, *European Journal of Operational Research,* **144**, 629-645.

Kim, K., Yamazaki, G., Lin, L. and Gen, M. (2004), Network-based Hybrid Genetic Algorithm to the Scheduling in FMS environments, *J. of Artificial Life and Robotics*, **8(1)**, 67-76.

Lee, M. and Takagi, H. (1995), Dynamic Control of Genetic Algorithm using Fuzzy Logic Techniques, *Proc. of 5th Inter. Conf. on Genetic Algorithm*.

Michalewicz, Z. (1996), *Genetic Algorithm + Data Structures = Evolution Programs*, Revised ed., New York: Springer-Verlag.

Naso, D. and Turchiano, B. (2005), Multicriteria meta-heuristics for AGV dispatching control based on computational intelligence, *IEEE Trans. on Sys. Man & Cyb.-B,* **35(2)**, 208-226.

Pirkul, H. and Jayaraman V. A. (1998), Multi-commodity, mulit-plant, capacitated location allocation problem: Formulation and efficient heuristic solution, *Computers and Operations Research*, **25(10)**, 869-878.

Rogers, D. S. and Tibben-Lembke, R. S. (1998), Going Backwards: Reverse Logistics Trends and Practices, Working paper, Centre for Logistics Management, University of Nevada, Reno.

Syam, S.S. (2002), A model and methodologies for the location problem with logistical components, *Computers and Operations Research*, **29**, 1173-1193.

Syarif, A. and Gen, M. (2003), Solving exclusionary side constrained transportation problem by using a hybrid spanning tree-based genetic algorithm, *Journal of Intelligent Manufacturing*, **14**, 389-399.

Syarif, A., Yun, Y. and Gen, M. (2002), Study on multi-stage logistics chain network: a spanning tree-based genetic algorithm approach, *Computers and Industrial Engineering,* **43**, 299-314.

Tansini, L., Urquhart, M. and Viera, O. (1999), Comparing assignment algorithms for the Multi-Depot VRP, *Jornadas de Informática e Investigación Operativa*, Technical Report, University of Montevideo, Uruquay.

Tragantalerngsak, S., Holt, J. and Ronnqvist, M. (1997), Lagrangian heuristics for the two-echelon, single-source, capacitate location problem, *European Journal of Operational Research*, **102**, 611-625.

Vignaux., G. A. and Michalewicz, Z. (1991), A genetic algorithm for the linear transportation problem, *IEEE Transactions* on *Systems, Man, and Cybernetics,* **21(2)**, 445 – 452.

VRP Web [Online]. Available : http://neo.lcc.uma.es/radi-eb/WebVRP/

Wang P. T., Wang, G. S. and Hu, Z. G. (1997), Speeding up the Search Process of Genetic Algorithm by Fuzzy Logic, *Proc. of 5th EUFIT*, 665-671.

Wu, T. H., Low, C. and Bai, J. W. (2002), Heuristic solutions to multi-depot location-routing problems, *Computers & Operations Research*, **29**, 1393-1415.

Xu Q. and Vukovich, G. (1994), Fuzzy Evolutionary Algorithms and Automatic Robot Trajectory Generation, *Proc. of IEEE International Conference on Evolutionary Computation*.

Yang, J. B. (2001), GA-Based Discrete Dynamic Programming Approach for Scheduling in FMS Environment, *IEEE Trans. on Sys, Man, and Cyb.-B*, **31(5)**, 824-835.

You, P. S. and Chen, T. C. (2005), An Efficient Heuristic for Series–parallel Redundant Reliability Problems, *Computers and Operations Research*, **32(8)**, 2117-2127.

Zeng, X. and Rabenasolo, B. (1997), A Fuzzy Logic Based Design for Adaptive Genetic Algorithms, *Proc. of the 5th European Congress on Intelligent Techniques and Soft Computing*, 660 – 664.