Available online at www.sciencedirect.com
ScienceDirect
Computers & Industrial Engineering 51 (2006) 196–215
computers & industrial engineering
www.elsevier.com/locate/dsw

A genetic algorithm approach for multi-objective optimization of supply chain networks

Fulya Altiparmak [a,*], Mitsuo Gen [b], Lin Lin [b], Turan Paksoy [c]

[a] *Department of Industrial Engineering, Gazi University, Turkey*
[b] *Graduate School of Information, Production and Systems, Waseda University, Japan*
[c] *Department of Industrial Engineering, Selcuk University, Turkey*

Available online 22 August 2006

Abstract

Supply chain network (SCN) design is to provide an optimal platform for efficient and effective supply chain management. It is an important and strategic operations management problem in supply chain management, and usually involves multiple and conflicting objectives such as cost, service level, resource utilization, etc. This paper proposes a new solution procedure based on genetic algorithms to find the 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. An experimental study using actual data from a company, which is a producer of plastic products in Turkey, is carried out into two stages. While the effects of weight approaches on the performance of proposed solution procedure are investigated in the first stage, the proposed solution procedure and simulated annealing are compared according to quality of Pareto-optimal solutions in the second stage.

Keywords: Supply chain network; Genetic algorithm; Multi-objective optimization

1. Introduction

A supply chain is a set of facilities, supplies, customers, products and methods of controlling inventory, purchasing, and distribution. The chain links suppliers and customers, beginning with the production of raw material by a supplier, and ending with the consumption of a product by the customer. In a supply chain, the flow of goods between a supplier and customer passes through several stages, and each stage may consist of many facilities (Sabri & Beamon, 2000). In recent years, the supply chain network (SCN) design problem has been gaining importance due to increasing competitiveness introduced by the market globalization (Thomas & Griffin, 1996). Firms are obliged to maintain high customer service levels while at the same time

* Corresponding author.
E-mail addresses: fulyaal@gazi.edu.tr (F. Altiparmak), gen@waseda.jp (M. Gen), lin@ruri.waseda.jp (L. Lin), tpaksoy@yahoo.com (T. Paksoy).

0360-8352/$ - see front matter
doi:10.1016/j.cie.2006.07.011

they are forced to reduce cost and maintain profit margins. Traditionally, marketing, distribution, planning, manufacturing, and purchasing organizations along the supply chain operated independently. These organizations have their own objectives and these are often conflicting. But, there is a need for a mechanism through which these different functions can be integrated together. Supply chain management (SCM) is a strategy through which such integration can be achieved. Illustration of a supply chain network is shown in Fig. 1.

The network design problem is one of the most comprehensive strategic decision problems that need to be optimized for long-term efficient operation of whole supply chain. It determines the number, location, capacity and type of plants, warehouses, and distribution centers to be used. It also establishes distribution channels, and the amount of materials and items to consume, produce, and ship from suppliers to customers. SCN design problems cover wide range of formulations ranged from simple single product type to complex multi-product one, and from linear deterministic models to complex non-linear stochastic ones. In literature, there are different studies dealing with the design problem of supply networks and these studies have been surveyed by Vidal and Goetschalckx (1997), Beamon (1998), Erenguc, Simpson, and Vakharia (1999), and Pontrandolfo and Okogbaa (1999).

An important component in SCN design and analysis is the establishment of appropriate performance measures. A performance measure, or a set of performance measures, is used to determine efficiency and/or effectiveness of an existing system, to compare alternative systems, and to design proposed systems. These measures are categorized as qualitative and quantitative. Customer satisfaction, flexibility, and effective risk management belong to qualitative performance measures. Quantitative performance measures are also categorized by: (1) objectives that are based directly on cost or profit such as cost minimization, sales maximization, profit maximization, etc. and (2) objectives that are based on some measure of customer responsiveness such as fill rate maximization, customer response time minimization, lead time minimization, etc. (Beamon, 1998). In traditional supply chain management, the focus of the integration of SCN is usually on single objective such as minimum cost or maximum profit. For example, Jayaraman and Pirkul (2001), Jayaraman and Ross (2003), Yan, Yu, and Cheng (2003), Syam (2002), Syarif, Yun, and Gen (2002), Amiri (2006), Gen

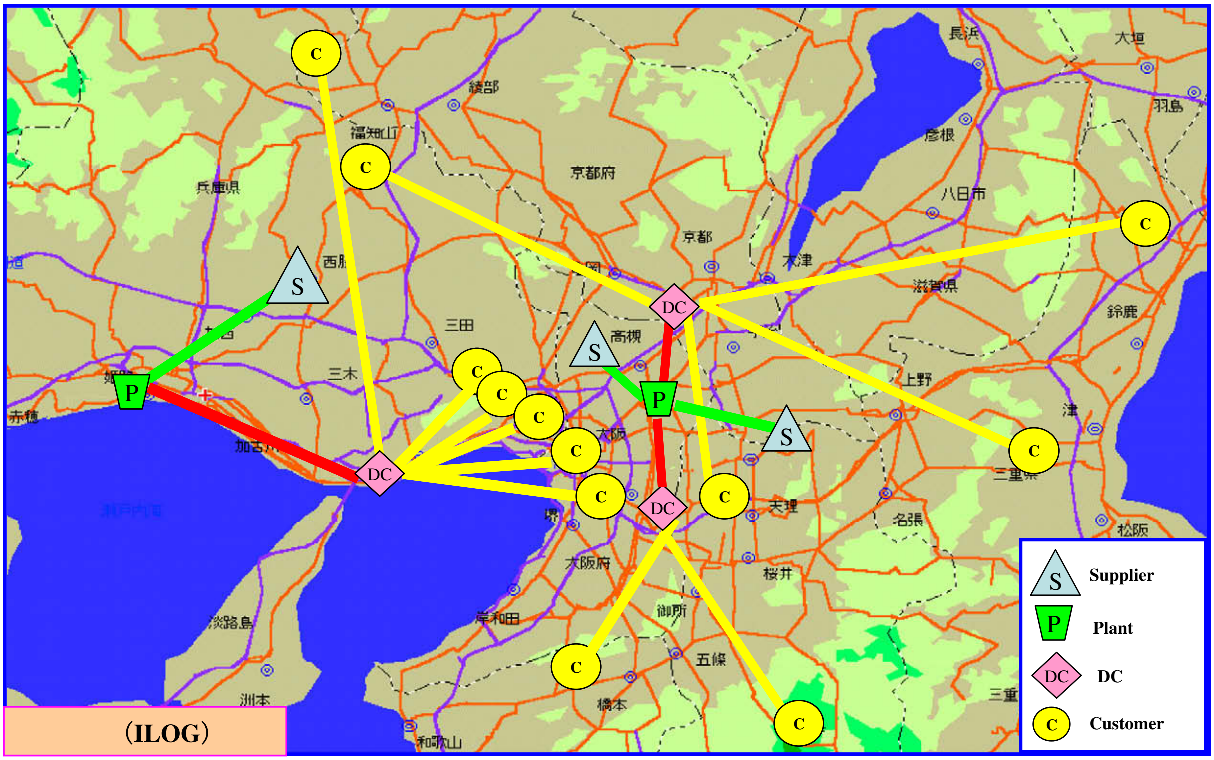


Fig. 1. Illustration of a Supply Chain Network (ILOG).

and Syarif (2005), and Truong and Azadivar (2005) had considered total cost of supply chain as an objective function in their studies. However, there are no design tasks that are single objective problems. The design/planning/scheduling projects are usually involving trade-offs among different incompatible goals. Recently, multi objective optimization of SCNs has been considered by different researchers in literature. Sabri and Beamon (2000) developed an integrated multi-objective supply chain model for strategic and operational supply chain planning under uncertainties of product, delivery and demand. While cost, fill rates, and flexibility were considered as objectives, ε-constraint method had been used as a solution methodology. Chan and Chung (2004) proposed a multi-objective genetic optimization procedure for the order distribution problem in a demand driven SCN. They considered minimization of total cost of the system, total delivery days and the equity of the capacity utilization ratio for manufacturers as objectives. Chen and Lee (2004) developed a multi-product, multi-stage, and multi-period scheduling model for a multi-stage SCN with uncertain demands and product prices. As objectives, fair profit distribution among all participants, safe inventory levels and maximum customer service levels, and robustness of decision to uncertain demands had been considered, and a two-phased fuzzy decision-making method was proposed to solve the problem. Erol and Ferrell (2004) proposed a model that assigning suppliers to warehouses and warehouses to customers. They used a multi-objective optimization modeling framework for minimizing cost and maximizing customer satisfaction. Guillen, Mele, Bagajewicz, Espuna, and Puigjaner (2005) formulated the SCN design problem as a multi-objective stochastic mixed integer linear programming model, which was solved by ε-constraint method, and branch and bound techniques. Objectives were SC profit over the time horizon and customer satisfaction level. Chan, Chung, and Wadhwa (2004) developed a hybrid approach based on genetic algorithm and Analytic Hierarch Process (AHP) for production and distribution problems in multi-factory supply chain models. Operating cost, service level, and resources utilization had been considered as objectives in their study. The studies reviewed above have found a Pareto-optimal solution or a restrictive set of Pareto-optimal solutions based on their solution approaches for the problem. Our purpose in this paper is to present a solution methodology to obtain all Pareto-optimal solutions for the SCN design problem and enable the decision maker for evaluating a greater number of alternative solutions.

During the last decade, there has been a growing interest using genetic algorithms (GA) to solve a variety of single and multi-objective problems in production and operations management that are combinatorial and NP hard (Gen & Cheng, 2000; Dimopoulos & Zalzala, 2000; Aytug, Khouja, & Vergara, 2003). In this study, we proposed a new approach based on GA for multi-objective optimization of SCNs which is one of the NP hard problems. Three objectives were considered: (1) minimization of total cost comprised of fixed costs of plants and distribution centers (DCs), inbound and outbound distribution costs, (2) maximization of customer services that can be rendered to customers in terms of acceptable delivery time (coverage), and (3) maximization of capacity utilization balance for DCs (i.e. equity on utilization ratios). The proposed GA was designed to generate Pareto-optimal solutions considering two different weight approaches. To investigate the effectiveness of the proposed GA, an experimental study using actual data from a company, which is a producer of plastic products in Turkey, was carried out into two stages. While the effects of weight approaches on the performance of proposed GA were investigated in the first stage, the proposed GA and multi-objective simulated annealing (MO_SA) proposed by Ulungu, Teghem, Fortemps, and Tuyttens (1999) were compared according to quality of Pareto-optimal solutions in the second stage.

The paper is organized as follows: In Section 2, multi-objective SCN design problem is formulated and discussed. Comprehensive explanation of the proposed GA is given in Section 3. Section 4 gives the computational results to show the performance of the GA using actual data obtained from a company in Turkey. Finally, concluding remarks are outlined and future research directions highlighted in Section 5.

2. Problem statement

The problem considered in this paper has been from a company which is one of the producers of plastic products in Turkey. The company is planning to produce plastic profile which is used in buildings (vinyl sidings, doors, windows, fences, etc.), pipelines and consumer materials. The main raw material of the plastic profile is PVC. The company wishes to design of SCN for the product, i.e. select the suppliers, determine the subsets of plants and DCs to be opened and design the distribution network strategy that will satisfy all capacities and demand requirement for the product imposed by customers. The problem is a single-product, multi-stage SCN design problem. Considering company managers' objectives, we formulated the SCN design problem as a multi-objective mixed-integer non-linear programming model. The objectives are minimization of the total cost of supply chain, maximization of customer services that can be rendered to customers in terms of acceptable delivery time (coverage), and maximization of capacity utilization balance for DCs (i.e. equity on utilization ratios). The assumptions used in this problem are: (1) the number of customers and suppliers and their demand and capacities are known, (2) the number of potential plants and DCs and their maximum capacities are known, (3) customers are supplied product from a single DC. Fig. 2 presents a simple network of three-stages in supply chain network.

The mathematical notation and formulation are as follows:

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.

$$z_j = \begin{cases} 1 & \text{if DC } j \text{ is open} \\ 0 & \text{otherwise} \end{cases}, \quad p_k = \begin{cases} 1 & \text{if plant } k \text{ is open} \\ 0 & \text{otherwise} \end{cases}$$

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

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 $C(\mathrm{j}) = \{i \mid h_{ji} \leqslant \tau\}$. 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.

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

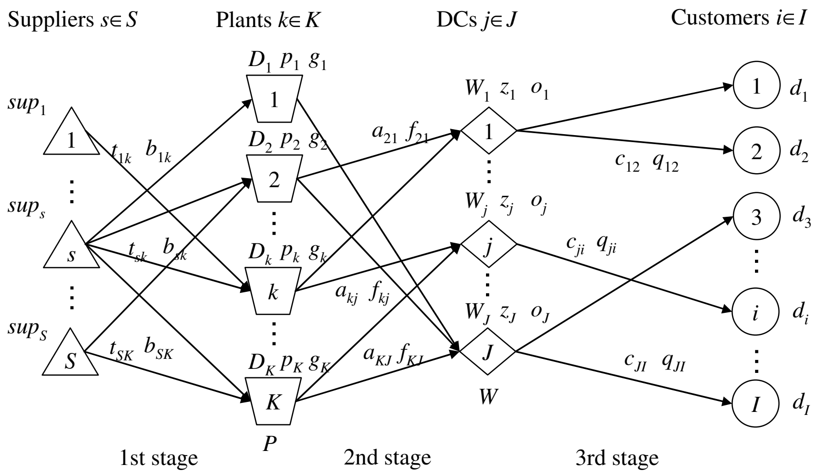


Fig. 2. A simple network of three-stages in supply chain network.

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.

$$\min f_1 = \sum_k g_k p_k + \sum_j v_j z_j + \sum_s \sum_k t_{sk} b_{sk} + \sum_k \sum_j a_{kj} f_{kj} + \sum_j \sum_i c_{ji} q_{ji} \tag{1}$$

$$\max f_2 = \left(\sum_{j \in o_{\mathrm{D}}} \sum_{i \in C(j)} q_{ji} \right) / \left(\sum_i d_i \right) \tag{2}$$

$$\begin{aligned} \min f_3 = {} & r_1 \left[\sum_{k \in o_{\mathrm{P}}} \left[\left(\sum_{j \in o_{\mathrm{D}}} f_{kj} / D_k \right) - \left(\sum_{k \in o_{\mathrm{P}}} \sum_{j \in o_{\mathrm{D}}} f_{kj} / \sum_{k \in o_{\mathrm{P}}} D_k \right) \right]^2 / |o_{\mathrm{P}}| \right]^{1/2} \\ & + r_2 \left[\sum_{j \in o_{\mathrm{D}}} \left[\left(\sum_i q_{ji} / W_j \right) - \left(\sum_{j \in o_{\mathrm{D}}} \sum_i q_{ji} / \sum_{j \in o_{\mathrm{D}}} W_j \right) \right]^2 / |o_{\mathrm{D}}| \right]^{1/2} \end{aligned} \tag{3}$$

$$\text{s.t.} \sum_j y_{ji} = 1, \quad \forall i \tag{4}$$

$$\sum_i d_i y_{ji} \leqslant W_j z_j, \quad \forall j \tag{5}$$

$$\sum_j z_j \leqslant W \tag{6}$$

$$q_{ji} = d_i y_{ji}, \quad \forall i, j \tag{7}$$

$$\sum_k f_{kj} = \sum_i q_{ji}, \quad \forall j \tag{8}$$

$$\sum_k b_{sk} \leqslant \mathrm{sup}_s, \quad \forall s \tag{9}$$

$$u \sum_j f_{kj} \leqslant \sum_s b_{sk}, \quad \forall k \tag{10}$$

$$u \sum_j f_{kj} \leqslant D_k p_k, \quad \forall k \tag{11}$$

$$\sum_k p_k \leqslant P \tag{12}$$

$$z_j = \{0, 1\}, \quad \forall j \tag{13}$$

$$p_k = \{0, 1\}, \quad \forall k \tag{14}$$

$$y_{ji} = \{0, 1\}, \quad \forall i, j \tag{15}$$

$$b_{sk} \geqslant 0, \quad \forall s, k \tag{16}$$

$$f_{kj} \geqslant 0, \quad \forall j, k \tag{17}$$

$$q_{ji} \geqslant 0, \quad \forall i, j \tag{18}$$

Eqs. (1)–(3) gives the objectives. While (1) defines the total cost of the SCN, (2) and (3) give the objectives about customer service and equity of the capacity utilization ratio (i.e., capacity utilization balance), respectively. Constraint (4) represents the unique assignment of a DC to a customer, (5) is the capacity constraint for DCs, (6) limits the number of DCs that can be opened, (7) and (8) gives the satisfaction of customer and DCs demands for the product, (9) describes the raw material supply restriction, (10) gives the supplier capacity constraint, (11) is the plant production capacity constraint, (12) limits the number of plants that are opened, (13)–(15) impose the integrality restriction on the decision variables z_j, p_k, y_{ij}, (16)–(18) impose the non-negativity restriction on decision variables b_{sk}, f_{kj}, q_{ij}. Since the third objective is nonlinear, the model given above is a mixed-integer non-linear programming model.

The multi-objective optimization problems as given above often contain many optimal solutions. These are Pareto-optimal solutions. The set of Pareto-optimal solutions of a multi-objective optimization problem consists of all decision vectors for which the corresponding objective vectors cannot be improved in a given dimension without worsening another (Chankong & Haimes, 1983). When a minimization problem and two decision vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ are considered, the concept of Pareto optimality can be defined as follows: $\boldsymbol{X}$ is said to dominate $\boldsymbol{Y}$ (also written as $\boldsymbol{X} \succ \boldsymbol{Y}$) iff:

$$f_i(\boldsymbol{X}) \leqslant f_i(\boldsymbol{Y}) \text{ for all } i \in \{1, 2, \ldots, m\} \text{ and}$$
$$f_i(\boldsymbol{X}) < f_i(\boldsymbol{Y}) \text{ for at least one } i \in \{1, 2, \ldots, m\}$$

All decision vectors, which are not dominated by another decision vector of a given set, are called non-dominated as regards that set. There are various solution approaches for solving the multi-objective problem. Among the most widely used techniques are sequential optimization, ε-constraint method, weighting method, goal programming, goal attainment, distance-based method and direction-based method. Recently, GA has been successfully applied to obtain Pareto-optimal solutions for multi-objective optimization problems (Ceollo, Van Veldhuizen, & Lamont, 2002; Deb, 2001; Gen & Cheng, 2000). GA deals simultaneously with a set of possible solutions (population) instead of having to perform a series of separate runs in the case of the traditional mathematical programming techniques. This property increases its popularity on multi-objective optimization. In this study, we also proposed a new GA approach to obtain Pareto-optimal solutions for SCN design problem.

3. Proposed genetic algorithm

In this section, representation and genetic operators which were used in GA for multi-objective design of SCN will be explained.

3.1. Representation

Representation is one of the important issues that affect the performance of GAs. Tree-based representation is known to be one way for representing network problems. Basically, there are three ways of encoding tree: (1) edge-based encoding, (2) vertex-based encoding, and (3) edge-and-vertex encoding (Gen & Cheng, 2000).

The first application of GAs to transportation/distribution problems was carried out by Michalewicz, Vignaux, and Hobbs (1991). They used matrix-based representation which belongs to edge-based encoding to represent transportation tree. When $|K|$ and $|J|$ are the number of sources and depots, respectively, the dimension of matrix is $|K|\cdot|J|$. Although it is a direct representation of the transportation tree, it needs not only excessive memory on the computer environment, but also special genetic operators to obtain feasible solutions. Another representation for transportation tree is Prüfer number. It belongs to vertex-based encoding and needs only $|K| + |J|$-2 digits to represent a transportation tree with $|K|$ sources and $|J|$ depots. Although Prüfer number, which was actually developed to encode of spanning trees, had been successfully applied to transportation problem by Gen and Cheng (2000), it needs some repair mechanisms to obtain feasible solutions after classical genetic operators.

In this study, to escape from these repair mechanisms in the search process of GA, we used priority-based encoding developed by Gen and Cheng (2000). 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, Altiparmak, and Lin (2006), and its extension to design of multi-product, multi-stage SCN had been made by Altiparmak, Gen, and Lin (2005). 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. Fig. 3 represents a transportation tree with 3 sources and 4 depots, its cost matrix and priority based encoding. The decoding algorithm of the priority-based encoding is given in Fig. 4. Table 1 gives trace table of the decoding procedure to obtain transportation tree in Fig. 3.

In SCN design problem, a chromosome consists of three segments. Each of the segments is used to obtain a transportation tree of a stage on the supply chain, i.e. the rth segment of a chromosome matches the rth stage on the SCN. We utilized two different encoding methods to design SCN. The priority-based encoding had been used on the first and second stages. Since each customer for the product has to be assigned only one DC on the last stage of our problem, integer encoding was used to define this situation. The length of the last segment on a chromosome equals to number of customers on SCN. The position of a gene on the segment represents a customer, and its value also represents the DC that corresponding customer will be assigned. At the same time, gene values show that which DCs will be opened. The chromosome of SCN is decoded on the backward direction. Firstly, transportation tree between opened DCs and customers is obtained with decoding of the last segment of chromosome. In the second step, firstly, a decision about which plants will be opened is given. Plants considering their priorities are opened consecutively until their number reaches to maximum number of plants to be opened or their total capacity is greater than or equal to total demand. After that, second segment is decoded and transportation tree between opened DCs and opened plants is obtained. Additionally, the amount of product which will be produced in opened plants and total amount of requirement for raw material on each plant are also determined in this stage. Lastly, transportation tree between suppliers and opened plants is obtained with decoding of the first segment of chromosome. It is worthy note that decisions about which DCs and plants will be opened are given during the decoding of the third and second segments of the chromosome. If the number of opened DCs or plants is greater than their upper limit or their capacities are not enough to meet customer demands, corresponding segments are repaired to obtain feasible transportation tree for each stage of the SCN. Fig. 5 gives an illustration of a feasible chromosome for the problem. In this example, we considered a SCN that has 3 suppliers, 3 plants, 4 DCs and 4 customers and the upper limits of opened plants and DCs were taken as 3. It is important to note that since there is an unbalanced transportation problem on each stage of the SCN (i.e. total capacity of sources is greater than total demands of depots for each stage) it is balanced by introducing a dummy depot on each segment of the chromosome. Overall decoding procedure for priority-based encoding on SCN, decoding procedures for 3rd, 2nd, and 1st stages, respectively, and repair algorithm, which is used when total capacity of opened DCs or plants is not enough to meet customer demands, are given in Appendix A.

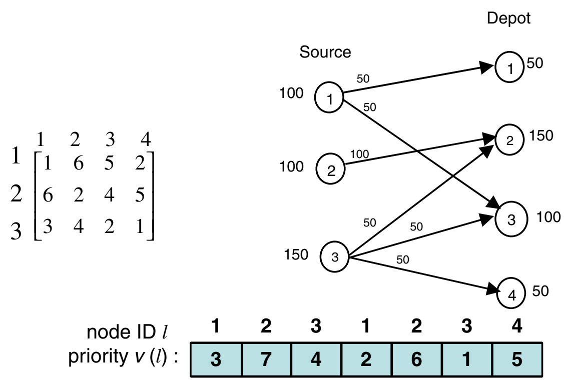


Fig. 3. A sample of transportation tree and its encoding.

procedure 1: Decoding of the chromosome for transportation tree
input : K : set of sources,
J : set of depots,
b_j : demand on depot j, $\forall j \in J$,
a_k : capacity of source k, $\forall k \in K$,
c_{kj} : transportation cost of one unit of product from source k to depot j, $\forall k \in K, \forall j \in J$,
$v(k+j)$: chromosome, $\forall k \in K$, $\forall j \in J$,
output : g_{kj} : the amount of product shipped from source k to depot j
step 1. $g_{kj} \leftarrow 0$, $\forall k \in K$, $\forall j \in J$,
step 2. $l \leftarrow \arg\max\{v(t),\ t \in |K|+|J|\}$; select a node
step 3. **if** $l \in K$, **then** $k^* \leftarrow l$; select a source
$j^* \leftarrow \arg\min\{c_{kj} \mid v(j) \neq 0, j \in J\}$; select a depot with the lowest cost
else $j^* \leftarrow l$; select a depot
$k^* \leftarrow \arg\min\{c_{kj} \mid v(j) \neq 0, k \in K\}$; select a source with the lowest cost
step 4. $g_{k^*j^*} \leftarrow \min\{a_{k^*}, b_{j^*}\}$; assign available amount of units
Update availabilities on source (k^*) and depot (j^*)
$a_{k^*} = a_{k^*} - g_{k^*j^*}$ $b_{j^*} = b_{j^*} - g_{k^*j^*}$
step 3. **if** $a_{k^*} = 0$ **then** $v(k^*) = 0$
if $b_{j^*} = 0$ **then** $v(j^*) = 0$
step 5. **if** $v(|K|+j) = 0$, $\forall j \in J$, **then** calculate transportation cost and return,
else goto Step 1.

Fig. 4. Decoding algorithm for the priority-based encoding.

Table 1
Trace table of decoding procedure

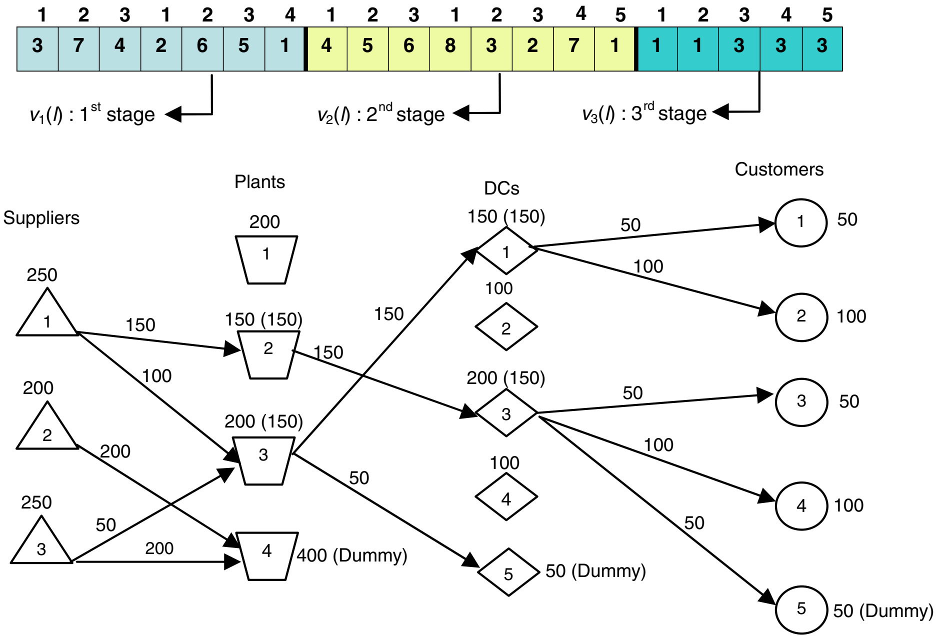
Iteration	$v(k+j)$	a	b	k	j	g_{kj}
0	[3 7 4 | 2 6 1 5]	(100, 100, 150)	(50, 150, 100, 50)	2	2	100
1	[3 0 4 | 2 6 1 5]	(100, 0, 150)	(50, 50, 100, 50)	3	2	50
2	[3 0 4 | 2 0 1 5]	(100, 0, 100)	(50, 0, 100, 50)	3	4	50
3	[3 0 4 | 2 0 1 0]	(100, 0, 50)	(50, 0, 100, 0)	3	3	50
4	[3 0 0 | 2 0 1 0]	(100, 0, 0)	(50, 0, 50, 0)	1	1	50
5	[3 0 0 | 0 0 1 0]	(50, 0, 0)	(0, 0, 50, 0)	1	3	50
6	[0 0 0 | 0 0 0 0]	(0, 0, 0)	(0, 0, 0, 0)			

3.2. 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 m objective functions, fitness function is obtained by combining the objective functions

$$\text{eval}(f) = \sum_{i=1}^{m} w_i f_i, \tag{19}$$

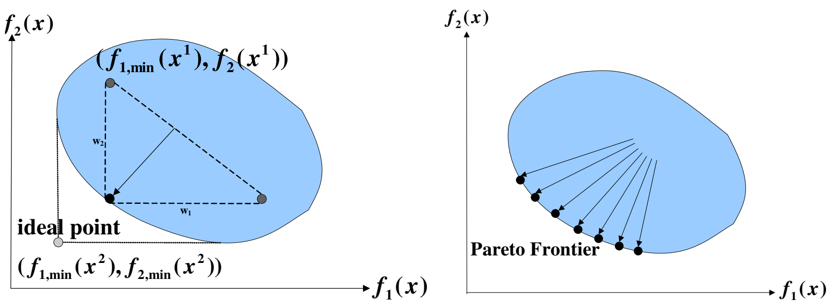
where w_i is constant representing weight for f_i, and $\sum_{i=1}^{m} w_i = 1$.



$$t=\begin{array}{c} \\ 1\\ 2\\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array}\\ \begin{bmatrix} 4 & 3 & 1\\ 3 & 5 & 2\\ 1 & 6 & 4 \end{bmatrix} \end{array} \qquad a=\begin{array}{c} \\ 1\\ 2\\ 3 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array}\\ \begin{bmatrix} 5 & 2 & 4 & 3\\ 4 & 6 & 3 & 5\\ 3 & 5 & 1 & 6 \end{bmatrix} \end{array} \qquad c=\begin{array}{c} \\ 1\\ 2\\ 3\\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array}\\ \begin{bmatrix} 3 & 5 & 2 & 4\\ 6 & 2 & 5 & 1\\ 4 & 3 & 6 & 5\\ 2 & 4 & 3 & 2 \end{bmatrix} \end{array}$$

Fig. 5. An illustration of chromosome, transportation trees and transportation costs for each stage on SCN.

To determine the weight values, we adopted two approaches proposed by Murata, Ishibuchi, and Tanaka (1996) and Zhou and Gen (1999). Approach 1 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 2, weights are determined based on the ideal point generated in each evolutionary process (Zhou & Gen, 1999). Fig. 6 illustrates these two strategies in the



(1) Zhou and Gen's strategy (2) Murata's et al. Strategy

Fig. 6. Illustration of the weight strategies.

objective space. When weight-sum approach is used, firstly, objectives have to be normalized since they generally have different units. Normalization is carried out for each objective as follows:

$$f_i' = \frac{f_i - f_{i,\min}}{f_{i,\max} - f_{i,\min}} \quad i = 1, 2, \ldots, m, \tag{20}$$

where $f_{i,\min}$ and $f_{i,\max}$ are the minimum and maximum value of ith objective on the current generation, respectively.

Approach 1. The weights in this approach are specified with Eq. (21) in each generation. After weights are determined, the fitness value of each individual on a population is calculated using Eq. (19).

$$\begin{aligned} &\text{random}_i \sim U(0,1) \\ &w_i = \text{random}_i/(\text{random}_1 + \cdots + \text{random}_m) \quad i = 1, 2, \ldots, m. \end{aligned} \tag{21}$$

Approach 2. Since the idea in this approach is to obtain Pareto-optimal solutions using ideal point generated in each evolutionary process, the weight of each objective for an individual in current generation is determined using Eq. (22).

$$w_1 = \frac{w_1'}{w_1' + w_2'}, \quad w_2 = \frac{w_2'}{w_1' + w_2'}, \tag{22}$$

where

$$w_1' = f_1'(x) - f_{1,\min}'(x), \quad w_2' = f_2'(x) - f_{2,\min}'(x)$$

$f_{1,\min}'$ and $f_{2,\min}'$ are the minimum of $f_1'(x)$ and $f_2'(x)$ in the current population, i.e. they are ideal point in the objective space whose value has been normalized.

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

3.3. Genetic operators

3.3.1. Crossover

The crossover is done to explore new solution space and crossover operator corresponds to exchanging parts of strings between selected parents. We employed a segment-based crossover operator which was based on uniform crossover. In this operator, each segment of offspring is randomly selected with equal chance among the corresponding segments of parents. As it is seen in Fig. 7, crossover operator utilizes from a binary mask. Its length is equal to number of stage in SCN. While "0" means that the first parent will transfer its genetic materials to the offspring, "1" means that the offspring will take genetic materials from the second parent for the corresponding segment. This crossover operator tends to preserve good gene segments of both parents.

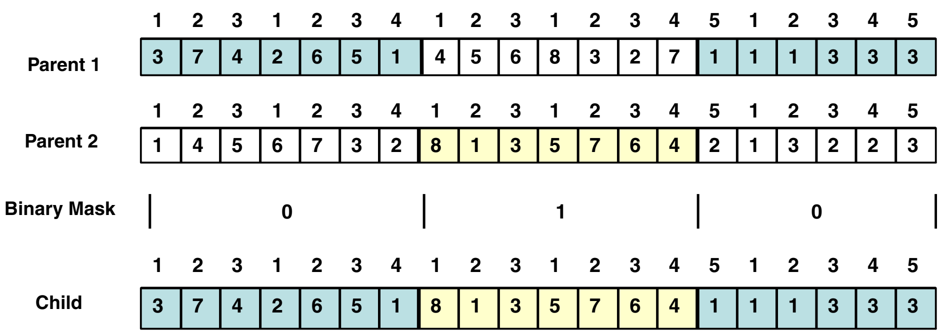


Fig. 7. An illustration of crossover operator.

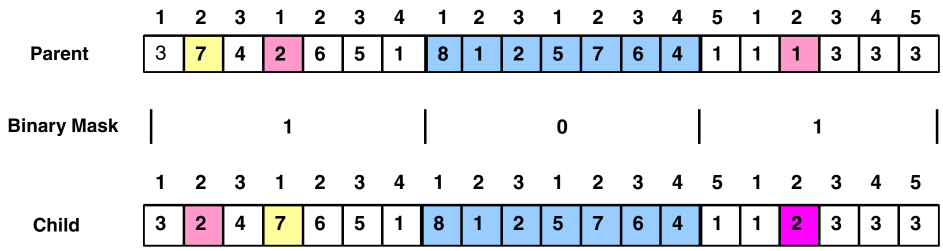


Fig. 8. An illustration of mutation operator.

3.3.2. Mutation

Similar to crossover, mutation is used to prevent the premature convergence and explore new solution space. However, unlike crossover, mutation is usually done by modifying gene within a chromosome. As in the crossover operator, we utilized from segment-based mutation as a mutation operator. In this operator, firstly, a decision about which segments will be mutated is given with probability of 0.5 (i.e. using a binary mask), and then selected segments are mutated. Since the chromosome consists of two different encoding structure, mutation on the structures is also different. Swap operator is used for the first two segments in where priority-based encoding is used. This operator selects two genes from the corresponding segment and exchanges their places. A conventional mutation operator is used for the last segment of the chromosome. In this operator, the value of randomly selected gene is replaced with new one which is selected between 1 and number of DCs except to its current value. Fig. 8 gives an illustration of segment-based mutation operator. As it is seen from figure, the first and third segments of the chromosomes are mutated.

3.4. Selection mechanism

In the proposed GA, initial population is randomly generated and Pareto-optimal set is created by non-dominated solutions in the initial population. This set is updated by new individuals obtained with genetic operators at every generation. As a selection mechanism, we adopted the $(\mu + \lambda)$ selection strategy. In this strategy, μ and λ, respectively, represent the number of parents and offspring, which constitutes the evolving pool in the current generation and competes for survival. After randomly selected two individuals from Pareto-optimal set as elite solutions are placed the population, the rest of population is filled by (μ-2) different best individuals selected from the evolving pool. If there are no (μ-2) different individuals available, the vacant pool of population is filled with randomly generated individuals. Additionally, we utilized from a diversification strategy to increase the capability of proposed GA for reaching more Pareto-optimal solutions. This strategy is based on the restart of genetic search. If the set of Pareto-optimal solutions has not been updated in the last moves (number of generations/5, in our study), the population is reset. While the 10% of the population is filled by non-dominated solutions which are randomly selected from the set of Pareto-optimal solutions, randomly generated solutions are placed to the rest of population. If there are no enough non-dominated solutions in the set of Pareto-optimal set to fill the 10% of the population, all non-dominated solutions are used in the new population.

4. Performance evaluation of the algorithm

The proposed GA is tested with the actual data obtained from a company which is one of the producers of plastic products in Turkey. In this section, after giving brief information about the company, computational results, which are carried out into two stages, will be presented. While the effects of the weight-sum approaches on the performance of GA are investigated in the first stage, the performances of GA and SA to obtain Pareto-optimal solutions are comparatively examined in the second stage.

Based on the market research, the company is planning to produce plastic profile which is used in buildings (vinyl sidings, doors, windows, fences, etc.), pipe lines and consumer materials. The market research shows that the company can capture a portion of the national market. PVC is the main raw material of the product to be produced. In Turkey, this material is supplied from Petkim Co. in Izmir. But Petkim cannot afford to supply all domestic demand. Thus, the company has to import this material from foreign suppliers in USA, Belgium, France and Japan. The company intends to establish new plants. There are three potential locations for the plants. These locations were determined depend on the some specific considerations. The first location has been considered as Izmir, since the national supplier Petkim had been settled in there. The second is Istanbul, because customs and duties are paid, and vessels are entered in customhouses for all imported goods. The last is Konya in where all other facilities of the company had been located. The company is planning to open at most six DCs. Locations of DCs had been determined according to demand densities of 63 customer zones to be served and access time from DCs to customer zones. The locations of DCs are Konya, Istanbul, Izmir, Ankara, Trabzon, and Adana. The company intends to establish supply chain network that satisfying the company objectives for the product. The company objectives, as given in mathematical model, are the minimization of overall supply chain cost, maximization of customer services, i.e. the percentage of customer demand that can be delivered within the stipulated access time τ and the maximization of capacity utilization balance for DCs (i.e. equity on utilization ratios). Table 2 gives information about suppliers' capacities, and capacity and fixed costs for plants and DCs. As it is seen from Table 2, 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.

The company is planning to meet customer demands from DCs within half of day (i.e. 12 h). The scatter diagram of the annual customer demand versus access time from the closest DCs is plotted to obtain information about how large the customer demands are, and how far away they are located from DCs. When Fig. 9 is examined, it is seen that the 93.4% of the customers have demands smaller than 20,000 packages per year. Also, when the capacities of DCs are not taken in the consideration, it is possible to reach the 98.3% of the customers within 12 h.

Unit costs between suppliers and plants including purchasing and transportation costs change between \$707 and \$775 per ton. Since the production costs do not exhibit any change for potential plant locations, unit

Table 2
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	10,000	Konya	640,000	440,000	Konya	200,000	70,000
Belgium	10,000	Istanbul	640,000	1,100,000	Istanbul	160,000	60,000
France	10,000	Izmir	640,000	720,000	Izmir	80,000	40,000
Japan	10,000				Ankara	120,000	50,000
Petkim	7200				Trabzon	80,000	40,000
					Adana	120,000	50,000

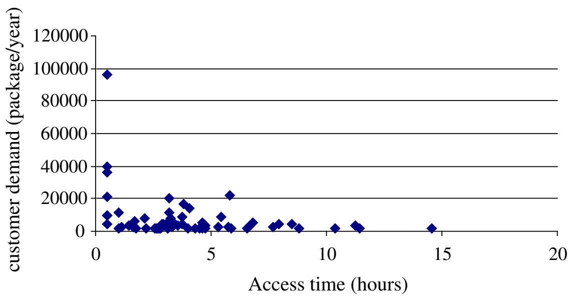


Fig. 9. Access time-demand distribution.

costs include only transportation costs between plants and DCs, and DCs and customers. The unit costs for stage 2 and stage 3 of the SCN take a value between \$0.18 and \$3.56, \$0.18, and \$7.49 per package, respectively.

4.1. Effects of weight-sum approaches on the performance of GA

In order to evaluate the performances of the GA_A1 and GA_A2 on SCN design problems with two-objective and three-objective, we considered three problems generated from original problem. They differ from each other only according to selected objectives. While the first two problems include two objectives, the last problem has three objectives, i.e. it is an original problem. 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 $\min f_3$

The proposed algorithm with two different evaluation approaches, GA_A1 and GA_A2, were coded with C++ programming language and run on Pentium 4, 2.8 GHz clock pulse with 512 MB memory. GA_A1 and GA_A2 run 10 times for each problem considering following parameters: population size = 400; crossover rate = 0.5, mutation rate = 0.7, number of generation = 500. These parameters had been determined after preliminary experiments. To evaluate the GA_A1 and GA_A2, we used two performance measures, which were obtained over 10 runs. These are: (1) average number of Pareto-optimal solutions, and (2) average ratio of Pareto-optimal solutions. The second performance measure was calculated following manner.

Let P_1 and P_2 be the sets of Pareto-optimal solutions obtained from one run of GA_A1, and GA_A2, respectively, and P be the union of the sets of Pareto-optimal solutions (i.e., $P = P_1 \cup P_2$) so that it includes only non-dominated solutions. The ratio of Pareto-optimal solutions in P_i that are not dominated by any other solutions in P is calculated using Eq. (23):

$$R_{\mathrm{POS}}(P_i) = \frac{|P_i - \{X \in P_i | \exists Y \in P : Y \prec X\}|}{|P_i|}, \tag{23}$$

where $Y \prec X$ means that the solution X is dominated by the solution Y. In (23), dominated solutions X by the solutions Y in P are removed from the solution set P_i. The higher the ratio $R_{\mathrm{POS}}(P_i)$ is, the better the solution set P_i is.

Experimental results are summarized in Table 3. As it is seen from table, while the average numbers of Pareto-optimal solutions are approximately equal on GA_A1 and GA_A2, GA_A1 outperforms the GA_A2 in terms of average ratio of Pareto-optimal solutions for all problems. The average ratio of Pareto-optimal solutions on GA_A1 changes between 52% and 78%. This ratio is between 51% and %70 on GA_A2. This result suggests that GA_A1 tends to find higher quality solutions than GA_A2. It is expected result. Because the proposed GA with Approach 1 randomly searches as many Pareto-optimal solutions as possible in the Pareto frontier, while other (i.e. GA with Approach 2) only focuses on some areas on the Pareto frontier. Figs. 10–12 also support this result. These figures give the examples of Pareto-optimal solutions obtained by GA_A1 and GA_A2 on a single run for each problem. As it is seen from these figures, most of the solutions generated by

Table 3
Comparison of GA_A1 and GA_A2

	Average number of Pareto-optimal solutions		Average ratio of Pareto-optimal solutions	
	GA_A1	GA_A2	GA_A1	GA_A2
Problem 1	2.3	2.5	0.77	0.56
Problem 2	12.6	14.7	0.52	0.51
Problem 3	32.3	33.3	0.78	0.70

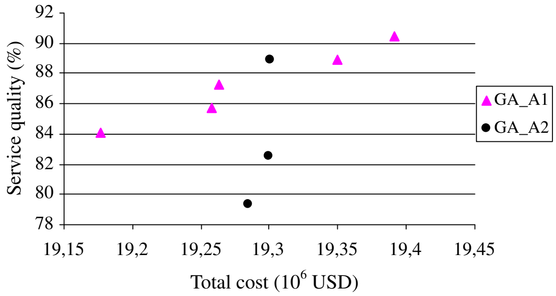


Fig. 10. Pareto-optimal solutions of GA_A1 and GA_A2 for Problem1.

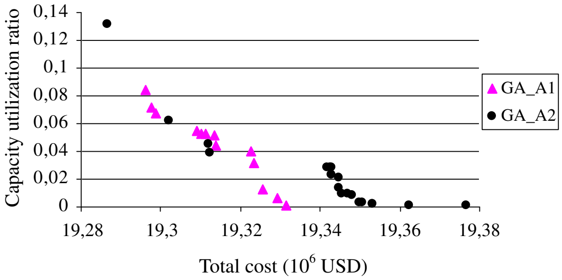


Fig. 11. Pareto-optimal solutions of GA_A1 and GA_A2 for Problem 2.

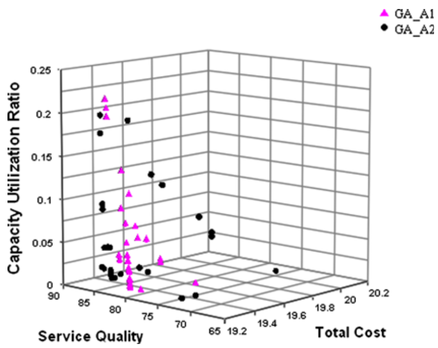


Fig. 12. Pareto-optimal solutions of GA_A1 and GA_A2 for Problem 3.

GA_A2 are dominated by the solutions obtained with GA_A1. Additionally, we investigated the effect of diversification mechanism on the quality of Pareto-optimal solutions obtained by GA_A1 and GA_A2. For this purpose, GA_A1 and GA_A2 without diversification mechanism were run 10 times for the third problem. Table 4 gives the average number of Pareto-optimal solutions and average ratio of Pareto-optimal solutions for GA_A1 and GA_A2 with diversification and without diversification. From Table 4, we can see that while average numbers of Pareto-optimal solutions obtained by GA_A1 with and without diversification are approximately equal, GA_A2 without diversification generates more Pareto-optimal solutions than its version with diversification. Meanwhile, the 81% and 87% of Pareto-optimal solutions obtained by GA_A1 and GA_A2 with diversification, respectively, are not dominated by GA_A1 and GA_A2 without diversification. This result is an indicator that diversification mechanism increases the quality of Pareto-optimal solutions.

To give information about which plants and DCs are opened in Pareto-optimal solutions of original problem, we selected five solutions among the Pareto-optimal solutions because of the space limitation. Table 5 gives the objective function values, and locations of plants and DCs on the selected solutions. We could

Table 4
Comparison of GA_A1 and GA_A2 with diversification and without diversification for the problem 3

	Average number of Pareto-optimal solutions	Average ratio of Pareto-optimal solutions
GA_A1 with diversification	32.3	0.81
GA_A1 without diversification	33	0.44
GA_A2 with diversification	33.3	0.87
GA_A2 without diversification	37.7	0.54

Table 5
Examples for Pareto-optimal solutions ($f_1 \times 10^6$)

Solutions	(f_1, f_2, f_3)	Locations of opened plants	Locations of opened DCs
1	(19.37, 0.92, 0.092)	Konya	Konya, İstanbul, Trabzon, Ankara
2	(19.34, 0.76, 0.013)	İzmir	Konya, İstanbul, İzmir, Ankara
3	(19.41, 0.91, 0.001)	Konya	Konya, İstanbul, Trabzon, Ankara
4	(19.36, 0.81, 0.096)	Konya	Konya, İstanbul, Ankara, Adana
5	(19.29, 0.67, 0.113)	İzmir	Konya, İstanbul, İzmir, Ankara

not give the allocation of 63 customers to DCs, since it will consume more space. As it is seen from Table 5, while the cost of SCN in the examples changes between $\$19.29 \times 10^6$ and $\$19.41 \times 10^6$, service quality and equity on utilization ratios take a value between 0.67 and 0.92, 0.001 and 0.113, respectively. There is a trade-off between solutions. When the cost of SCN decreases, it is observed that there is a reduction on the service quality, and equity on utilization ratios of the SCN. It is also important to note that when all Pareto-optimal solutions are examined, it is seen that one plant is opened on each solution, and its location in the 99% of solutions is Konya (43%) or Izmir (56%). Also, we observe from Pareto-optimal solutions that the number of opened DCs changes between three and five. While four DCs are opened in the 90% of solutions, three and five DCs are opened in the 8% and 2% of solutions, respectively. Another important issue on the Pareto-optimal solutions is the locations of DCs. While Konya, Istanbul, Izmir, and Ankara are selected in the 60% of solutions, Konya, Istanbul, Ankara, and Adana are selected in the 33% of solutions.

4.2. Comparison of GA_A1 and MO_SA

To investigate the effectiveness of the GA_A1 for SCN design problem, the SA approach proposed by Ulungu et al. (1999), called as MO_SA, was employed for the problem. Our purpose on selecting MO_SA was that it was also based on weight-sum approach. This property provides us making a comparison of the approaches on the same basis. We considered five problems on the comparison. While the first problem was original problem, others were generated from the original problem by increasing the customer demands, and the number of potential plants and DCs. The locations of additional plants were selected among the developing cities in Turkey, and the locations of additional DCs were determined considering regional demand

Table 6
The size of new problems

	Number of Plants, $\lvert K\rvert$	Number of DCs, $\lvert J\rvert$	Number of maximum opened plants, P	Number of maximum opened DCs, W
Problem 3	3	6	3	6
Problem 4	3	8	2	5
Problem 5	5	10	3	7
Problem 6	6	15	4	10
Problem 7	8	20	5	15

densities. Table 6 gives information about the number of potential plants and DCs, and number of maximum opened plants and DCs in the new problems.

In MO_SA, a set of weight vectors are randomly generated, and the Pareto-optimal solutions are obtained by running SA with predefined number of iterations for each random weight vector. SA starts with the randomly generated solution, and a new solution in the each iteration is obtained by moving strategy. If the new solution improves the current solution according to weight vectors or enters the set of Pareto-optimal solutions, it is accepted as current solution; otherwise it is accepted with the probability of $\exp(-\Delta s/T)$. We defined Δs as $[(\text{eval}(f') - \text{eval}(f))/\text{eval}(f)] * 100$ in where $\text{eval}(f')$ and $\text{eval}(f)$ were objective function values according to weight vector for the new solution and current solution, respectively. Δs is a relative percent deviation of quality of the new solution from the current solution. When the MO_SA was implemented for SCN design problem, the encoding structure in GA was used to represent a solution, and the mutation operator in GA was chosen as a moving strategy. The initial temperature was taken as 975 in which an inferior solution (inferior by 50% relative to current solution) was accepted with a probability of 0.95. To make comparison of GA_A1 and MO_SA on the same basis, the number of solutions searched (NSS) was used as a stopping criterion, and it depended on the problem size. In GA_A1, the population size and the number of generations were taken as 400 and NSS/400, respectively. In MO_SA, SA with the length of 100 iterations run for each of 400 different randomly generated weight vectors. The reduction rate of temperature and the number of evaluated solutions in each iteration of SA were 0.90 and NSS/(100 * 400), respectively. It is important to note that the MO_SA was also coded with C++ programming language. GA_A1 and MO_SA run 10 times, and two performance measures, mentioned in Section 4.1, were used to compare them. Table 7 summarizes experimental results. As it is seen from table, MO_SA is inferior to the GA_A1 in terms of the average number of Pareto-optimal solutions for all problems except to Problem 3. The comparison of GA_A1 and MO_SA with respect to average ratio of Pareto-optimal solutions shows that while the average ratio of Pareto-optimal solutions on GA_A1 is between 56% and 68%, it changes between 47% and 63% on MO_SA. These results suggest that GA_A1 tends to find more solutions with higher quality than MO_SA.

Table 8 gives NSS and computation times on GA_A1 and MO_SA for each problem size. As it is seen from the table that the computation times on GA_A1 and MO_SA increase based on problem size. Additionally,

Table 7
Comparison of GA_A1 and MO_SA

	Average number of Pareto-optimal solutions		Average ratio of Pareto-optimal solutions	
	GA_A1	MO_SA	GA_A1	MO_SA
Problem 3	32.3	36.7	0.58	0.54
Problem 4	59.6	45.1	0.56	0.47
Problem 5	55	55.2	0.65	0.55
Problem 6	42.9	38	0.64	0.63
Problem 7	53.9	48.3	0.68	0.62

Table 8
Number of solutions searched and CPU times for GA_A1 and MO_SA

	Number of solutions searched (NSS)	CPU times (min)	
		GA_A1	MO_SA
Problem 3	2×10^5	1.136	0.447
Problem 4	3×10^5	2.338	1.040
Problem 5	4×10^5	3.796	1.686
Problem 6	5×10^5	10.689	5.875
Problem 7	6×10^5	14.229	7.535

the computation times on GA_A1 are approximately two times higher than MO_SA on each problem. Although the same stopping criterion had been used on the algorithms, this difference came from the fact that the GA had some additional mechanisms such as selection mechanism and crossover, which were time consuming.

5. Conclusion

In this paper, we presented mixed-integer non-linear programming model for multi-objective optimization of SCN and a genetic algorithm (GA) approach to solve the problem which was met on a producer of the plastic products in Turkey. Three objectives were considered: (1) minimization of total cost comprised of fixed costs of plants and distribution centers (DCs), inbound and outbound distribution costs, (2) maximization of customer services that can be rendered to customers in terms of acceptable delivery time (coverage), and (3) maximization of capacity utilization balance for DCs (i.e. equity on utilization ratios). To deal with multi-objective and enable the decision maker to evaluate a greater number of alternative solutions, two different weight approaches were implemented in the proposed GA. In order to evaluate the performances of the GA with two different weight approach, called as GA_A1 and GA_A2, we considered three problems generated from original problem, which were different from each other according to selected objectives. Experimental results showed that while GA_A1 was capable to generate more Pareto-optimal solutions than GA_A2, diversification mechanism was very effective on the quality of Pareto-optimal solutions. In addition, GA_A1 was compared with the MO_SA using five problems which were generated from original problem. This comparison showed that GA_A1 outperformed MO_SA according to not only average number of Pareto-optimal solutions but also quality of Pareto-optimal solutions. In future, new solution methodology based on tabu search can be developed to obtain Pareto-optimal solutions for the multi-objective SCN design problem, and the effectiveness of GA_A1 according to this solution methodology can be investigated. Additionally, uncertainty of costs and demands can be considered in the model and new solution methodologies including uncertainty can be developed.

Acknowledgments

This research had been supported by The Matsumae International Foundation in Japan, while Dr. Fulya Altiparmak was a visiting researcher at Graduate School of Information, Production and Systems, Waseda University. Also this work was partly supported by Waseda University Grant for Special Research Projects 2004 and the Ministry of Education, Science and Culture, the Japanese Government: Grant-in-Aid for Scientific Research (No. 17510138).

Appendix A. Decoding procedure of the chromosome for SCN

See Figs. 13–17.

procedure 2: decoding of chromosome for SCN

step 1 : find $[q_{ij}]$ by procedure 3 (3rd stage decoding);

step 2 : **for** $j = 1$ **to** $|J|$
  if $b_j' = 0$ **then** $v_2(j) \leftarrow 0$;

step 3 : find $[f_{jk}]$ by procedure 4 (2nd stage decoding);

step 4 : **for** $k = 1$ to $|K|$
  if $b_k' = 0$ **then** $v_1(j) \leftarrow 0$;

step 5 : find $[b_{sk}]$ by procedure 5 (1st stage decoding);

step 6 : calculate the value of objective functions (z_1, z_2, and z_3) and stop.

Fig. 13. Decoding procedure for priority based encoding.

procedure 3 : 3rd stage decoding

input : J : set of DCs, I : set of customers,
W_j : capacity of DC j, $\forall j \in J$,
d_i : demand for product of customer i, $\forall i \in I$
$v_3(i)$: chromosome, $\forall\ i \in I$

output : o_D : set of opened DCs, c_D : set of closed DCs
q_{ij} : the amount of product shipped from DC j to customer i
W_j' : total customer demand for product on DC j

step 0 : $z_j \leftarrow 0, \forall j \in J, y_{ij} \leftarrow 0, \forall i \in I, \forall j \in J, o_D \leftarrow \varnothing, c_D \leftarrow J$
$q_{ij} \leftarrow 0, \forall i \in I, \forall j \in J; W_j' \leftarrow 0, \forall j \in J$

step 1 : determine the opened and closed DCs and find DC j serves customers i
for $i = 1$ to I
$$z_{v_3(i)} \leftarrow 1,\quad y_{iv_3(i)} \leftarrow 1,\quad o_D = o_D + \{v_3(i)\},\quad c_D = c_D - \{v_3(i)\}$$

step 2 : Calculate total capacity of opened DCs and total customer demand, and number of opened DCs
$$tot\_cap = \sum_j W_j z_j\ ,\ tot\_dem = \sum_i d_i\ ,\ \left|o_D\right| = \sum_j z_j$$

step 3 : **if** $o_D \le W$ **and** $tot\_cap \ge tot\_dem$ **then goto** Step 5; **else goto** step 4.

step 4 : $DOK \leftarrow o_D, DCK \leftarrow c_D, DOP \leftarrow o_P, DP \leftarrow P,$
$d\_tot\_cap \leftarrow tot\_cap, d\_tot\_dem \leftarrow tot\_dem$
call procedure 6,
obtain new $v_3(i)$ with reallocating the customers considering opened and closed DCs and **goto** step 0.

step 5 : for $i = 1$ to I
$$q_{iv_3(i)} \leftarrow d_i,\ \ W_{v_3(i)*} \leftarrow W_{v_3(i)*} - q_{iv_3(i)},\ \ W'_{v_3(i)} \leftarrow W'_{v_3(i)} + q_{iv_3(i)},\ \ d_i \leftarrow 0$$

step 6 : **if** $W_{j*} < 0$ for any $j \in J$, **then** select a customer from DC j and reallocate it to another opened DC that satisfy $W_{j*} \ge 0$ for all $j \in J$, obtain new $v_3(i)$, and recalculate the W_{j*}, W_j', q_{ij}

step 7 : **output** W_j', q_{ij} and **return**

Fig. 14. Decoding procedure for 3rd segment of the chromosome.

procedure 4 : 2nd stage decoding

input : K : set of plants, J : set of DCs, P : maximum number of plants
W'_j : total customer demand for product on DC j, $\forall j \in J$,
p_{jk} : shipping cost of one unit of product from plant k to DC $j, \forall k \in K, \forall j \in J$,
$v_2((k+j))$: chromosome, $\forall k \in K, \forall j \in J$,

output : f_{kj} : the amount of product shipped from plant k to DC j
D'_k : total customer demand for product on plant k, $\forall k \in K$,
$p(k)$: priority of plant k for product, $\forall k \in K$,
NP : number of opened plants, o_P : set of opened plants,
tot_cap : total capacity of opened plants
tot_dem : total demand of DCs

step 0 : $q_{kl} \leftarrow 0, \forall k \in K, \forall j \in J, D_k' \leftarrow 0, \forall k \in K,$
$$tot\_dem = \sum_i \sum_j q_{ij}$$

Fig. 15. Decoding procedure for 2nd segment of the chromosome.

step 1 : obtain $p(k)$ from $v_2((k+j))$, $k \in K$,

$p_d(k) \leftarrow p(k), \quad \forall k \in K$

step 2 : open the plants having high priorities until $tot\_cap \geq tot\_dem$ or $NP \geq P$

$hp\_k \leftarrow \arg\max\{p_d(k),\ k \in K\}$

$p_{hp\_k} \leftarrow 1, \quad tot\_cap = tot\_cap + D_{hp\_k},$

$NP = NP + 1,\ p_d(hp\_k) \leftarrow 0, \quad o_P \leftarrow o_P + \{k\}$

step 3 : set the priorities of the closed plants to 0 and keep the current priorities on $v_2((k+j))$ for the opened plants.

if $o_P \leq P$ and $tot\_cap \geq tot\_dem$ then goto step 5; else goto step 4.

step 4 : $DOK \leftarrow o_P$, $DCK \leftarrow K - o_P$, $DOP \leftarrow NP$, $DP \leftarrow P$,

$d\_tot\_cap \leftarrow tot\_cap$, $d\_tot\_dem \leftarrow tot\_dem$

call procedure 6, and goto Step 3.

step 5 : set the DCs as depots and plants as sources, **call Procedure 1** to obtain q_{kl} (i.e. transportation tree for the 2nd stage of the SCN), calculate D_k' considering q_{kl} and **return**.

Fig. 15 (*continued*)

procedure 5 : 1st stage decoding

input :
- S : set of suppliers,
- K : set of plants,
- D'_k : total customer demand for product on plant k, $\forall k \in K$,
- t_{sk} : unit transportation and purchasing cost of raw material from supplier s to plant k $\forall s \in S$, $\forall k \in K$
- u : utilization rate of raw material per unit of product ,
- a_k : the amount of raw material to produce the product on plant k
- $v_1(s+k)$: chromosome, $\forall s \in S$, $\forall k \in K$

output : b_{sk} : the amount of raw material shipped from supplier s to plant k, $\forall s \in S$, $\forall k \in K$

step 0 : $b_{sk} \leftarrow 0$, $\forall s \in S$, $\forall k \in K$

Calculate the amount of raw material to produce the product on plant k

$a_k = D'_k u \ , \forall k \in K$

step 1 : Set the plants as depots and suppliers as sources and **call Procedure 1** to obtain b_{sk} (i.e. transportation tree for the 1st stage of the SCN and **return**.

Fig. 16. Decoding procedure for 1st segment of the chromosome.

procedure 6: Repair algorithm

input: DOK : set of opened sources; DCK : set of closed sources;
DP : maximum number of sources; DOP : number of opened sources,
d_tot_cap : total capacity of opened sources
d_tot_dem : total requirement of depots

output: DOK : Set of opened sources

step 1. **if** $DOP > DP$ **and** $d\_tot\_cap \geq d\_tot\_dem$ **then goto** Step 2
If ($DOP < DP$ or $DOP \geq DP$) and $d\_tot\_cap < d\_tot\_dem$ or *then* goto Step 3

step 2. obtain a set of sources (CS) from the DOK that closing a source in CS will also satisfy the condition of $d\_tot\_cap \geq d\_tot\_dem$.

repeat

if $CS \neq \emptyset$, **then** close a source which is randomly selected from CS;
else close a randomly selected source from the set of opened sources.

until $DOP \leq DP$.

Recalculate the *tot_cap* considering closed sources and **return**.

step 3. **repeat**

open a source which is randomly selected from DCK

until $d\_tot\_cap \geq d\_tot\_dem$.

Recalculate the DOP considering opened sources and **return**.

Fig. 17. Repair algorithm.

References

Altiparmak, F., Gen, M., & Lin, L. (2005). A genetic algorithm for supply chain network design. In *Proceedings of the 35th International Conference on Computers and Industrial Engineering,* (pp. 111–116).

Amiri, A. (2006). Designing a distribution network in a supply chain system: formulation and efficient solution procedure. *European Journal of Operational Research, 171*(2), 567–576.

Aytug, H., Khouja, M., & Vergara, F. E. (2003). Use of genetic algorithms to solve production and operations management: a review. *International Journal of Production Researches, 41*(17), 3955–4009.

Beamon, B. M. (1998). Supply chain design and analysis: models and methods. *International Journal of Production Economics, 55*, 281–294.

Ceollo, C. A. C., Van Veldhuizen, D. A., & Lamont, G. B. (2002). *Evolutionary algorithms for solving multi-objective problems*. New York: Kluwer.

Chan, F. T. S., & Chung, S. H. (2004). A multi-criterion genetic algorithm for order distribution in a demand driven supply chain. *International Journal of Computer Integrated Manufacturing, 17*(4), 339–351.

Chan, F. T. S., Chung, S. H., & Wadhwa, S. (2004). A hybrid genetic algorithm for production and distribution. *Omega, 33*, 345–355.

Chankong, V., & Haimes, Y. Y. (1983). *Multiobjective decision making theory and methodology*. NewYork: Elsevier.

Chen, C., & Lee, W. (2004). Multi-objective optimization of multi-echelon supply chain networks with uncertain product demands and prices. *Computers and Chemical Engineering, 28*, 1131–1144.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester: Wiley.

Dimopoulos, C., & Zalzala, A. M. S. (2000). Recent developments in evolutionary computation for manufacturing optimization: problems, solutions and comparisons. *IEEE Transactions on Evolutionary Computation, 4*(2), 93–113.

Erenguc, S. S., Simpson, N. C., & Vakharia, A. J. (1999). Integrated production/distribution planning in supply chains: an invited review. *European Journal of Operational Research, 115*, 219–236.

Erol, I., & Ferrell, W. G. Jr., (2004). A methodology to support decision making across the supply chain of an industrial distributor. *International Journal of Production Economics, 89*, 119–129.

Gen, M., & Cheng, R. (2000). *Genetic algorithms and engineering optimization*. New York: Wiley.

Gen, M., Altiparmak, F., & Lin, L. (2006). A genetic algorithm for two-stage transportation problem using priority-based encoding. *OR Spectrum, 28*, 337–354.

Gen, M., & Syarif, A. (2005). Hybrid genetic algorithm for multi-time period production/distribution planning. *Computers and Industrial Engineering, 48*(4), 799–809.

Guillen, G., Mele, F. D., Bagajewicz, M. J., Espuna, A., & Puigjaner, L. (2005). Multiobjective supply chain design under uncertainty. *Chemical Engineering Science, 60*, 1535–1553.

ILOG, Inc. [Online]: <http://www.ilog.com/>.

Jayaraman, V., & Pirkul, H. (2001). Planning and coordination of production and distribution facilities for multiple commodities. *European Journal of Operational Research, 133*, 394–408.

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

Michalewicz, Z., Vignaux, G. A., & Hobbs, M. (1991). A non-standard genetic algorithm for the nonlinear transportation problem. *ORSA Journal on Computing, 3*, 307–316.

Murata, T., Ishibuchi, H., & Tanaka, H. (1996). Multi-objective genetic algorithm and its applications to flowshop scheduling. *Computers and Industrial Engineering, 30*(4), 957–968.

Pontrandolfo, P., & Okogbaa, O. G. (1999). Global manufacturing: a review and a framework for planning in a global corporation. *International Journal of Production Economics, 37*(1), 1–19.

Sabri, E. H., & Beamon, B. M. (2000). A multi-objective approach to simultaneous strategic and operational planning in supply chain design. *Omega, 28*, 581–598.

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

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

Thomas, D. J., & Griffin, P. M. (1996). Coordinated supply chain management. *European Journal of Operational Research, 94*, 1–115.

Truong, T. H., & Azadivar, F. (2005). Optimal design methodologies for configuration of supply chains. *International Journal of Production Researches, 43*(11), 2217–2236.

Ulungu, L. E., Teghem, J., Fortemps, P. H., & Tuyttens, D. (1999). MOSA: a tool for solving multiobjective combinatorial optimization problem. *Journal of Multi-Criteria Decision Analysis, 8*, 221–236.

Vidal, C. J., & Goetschalckx, M. (1997). Strategic production-distribution models: a critical review with emphasis on global supply chain models. *European Journal of Operational Research, 98*, 1–18.

Yan, H., Yu, Z., & Cheng, T. C. E. (2003). A strategic model for supply chain design with logical constraints: formulation and solution. *Computers and Operations Research, 30*(14), 2135–2155.

Zhou, G., & Gen, M. (1999). Genetic algorithm approach on multi-criteria minimum spanning tree problem. *European Journal of Operational Research, 114*, 141–152.