

A memetic algorithm with extended random path encoding for a closed-loop supply chain model with flexible delivery

Elham Behmanesh¹ · Jürgen Pannek^{2,3}

Received: 22 January 2016 / Accepted: 21 October 2016 / Published online: 7 November 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract Logistics network design is a major strategic issue in supply chain management of both forward and reverse flow, which industrial players are forced but not equipped to handle. To avoid sub-optimal solution derived by separated design, we consider an integrated forward reverse logistics network design, which is enriched by using a complete delivery graph. We formulate the cyclic seven-stage logistics network problem as a NP hard mixed integer linear programming model. To find the near optimal solution, we apply a memetic algorithm with a neighborhood search mechanism and a novel chromosome representation including two segments. The power of extended random path-based direct encoding method is shown by a comparison to commercial package in terms of both quality of solution and computational time. We show that the proposed algorithm is able to efficiently find a good solution for the flexible integrated logistics network.

Keywords Memetic algorithm · Closed-loop supply design · Random path · Flexible delivery

1 Introduction

Supply chain management (SCM) describes the discipline of optimizing the delivery of goods, services and information from supplier to customer. Logistics network design is known as one of the comprehensive strategic decision problems due to its impact on the efficiency and responsiveness of the supply chain including reducing cost and improving service quality. To this end, an optimal choice regarding number, location, capacity and type of plants, warehouses, and distribution centers as well as the amount of shipped materials needs to be obtained.

Within the full material cycle, we distinguish between the forward supply chain from the upstream supplier to the downstream customer, and the reverse one for possible recycling, reuse and disposal.

Due to economic benefits and environmental protection, industrial players are under a lot of pressure to take back the product after its use. Therefore, the reverse supply chain is becoming more relevant in both theory and practice. Although some firms such as General Motors, Kodak, and Xerox concentrate on reverse logistics and have obtained significant success in this area [1, 2], most logistics networks are not equipped to handle the reverse flow. To avoid sub-optimality of a solution derived by the separated design, we consider forward and reverse channel as an integrated model and analyse closed-loop supply chain as a system in total. To manage logistics system efficiently in terms of cost and delivery time as well as increase customer satisfaction, flexible and productive network design models are of particular interest. Many research works have been published in the fields of optimizing of supply chain network (SCN) [3, 4], while flexible delivery and multistage planning conditions are rarely found in the literature.

This article is part of a focus collection on “Dynamics in Logistics: Digital Technologies and Related Management Methods”.

✉ Elham Behmanesh
beh@biba.uni-bremen.de

¹ International Graduate School for Dynamics in Logistics, University of Bremen, Bremen, Germany

² Faculty of Production Engineering, University of Bremen, Bremen, Germany

³ BIBA Bremer Institut für Produktion und Logistik GmbH, Bremen, Germany

In this study, we investigate the integration of forward and reverse logistics network design, cf. Fig. 1. In the forward flow, new products are shipped from plants to customers via distribution centers and retailers to satisfy their demands. In the reverse flow, returned products are collected in collection-inspection centers for sorting disassembling for recovery, reuse or disposal, cf. [5, 6] for related frameworks. To enhance the logistic network efficiency and flexibility, we consider a full delivery graph in the forward flow with normal delivery (products are delivered from one echelon to another), direct delivery (products are transported from distribution centers to customers or via plants to retailers directly) and direct shipment (products are transported from plants to customers directly).

The objective of this paper is to develop and optimize a seven-stage closed-loop supply chain network with respect to number and capacity of plants, distribution centers (Dcs), retailers, collection/inspection centers (Cos) and disposal centers (Dis) as well as the product flow between the facilities. The aim of this study is to minimize the total cost including the transportation cost as well as operation cost.

The remainder of this paper is structured as follows. After systematically reviewing related literature in Sect. 2, we present our cyclic logistics network problem as a mixed integer linear programming (MILP) model in Sect. 3. As the problem is NP hard and therefore difficult to solve using classical methods such as branch-and-bound, Sect. 4 presents a random path, flexible delivery-based memetic algorithm (MA) with a neighborhood search mechanisms using a new chromosome representation. To assess the quality of the approach, we compare respective results for test cases to solutions obtained by LINGO in Sect. 5. The final Sect. 6 concludes the paper and points out directions of future research.

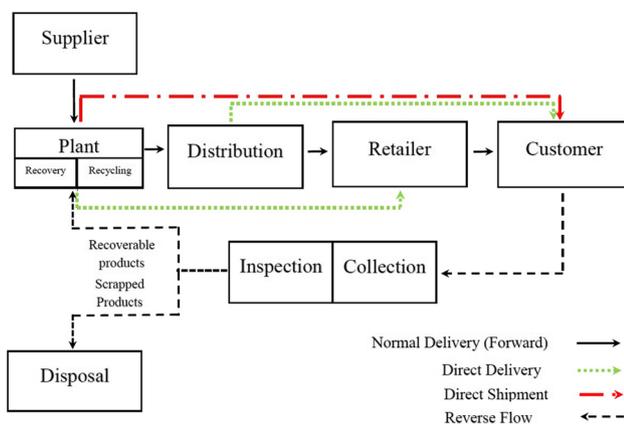


Fig. 1 Framework of the flexible IFRLN

2 Literature review and problem definition

Focusing on an efficient solution methodology, we split our literature review to the research areas network design with forward, reverse and integrated flows, and to flexible delivery paths.

2.1 Forward, reverse and integrated logistics network

In previous studies, the design of forward and reverse logistics network was typically separated. In the field of forward logistics many models were developed as part of a traditional logistics network design. The common MILP model involves the choice of facilities to be open and the distribution network design to satisfy the demand with minimum cost.

Yeh [7] developed a MILP model for a supplier-production-distribution network. He revised existing mathematical models presented by other researchers and developed an efficient hybrid heuristic, which was enriched by applying three different local search techniques. The efficiency of the proposed methodology was demonstrated via computational results. Syarif et al. [8] proposed a MILP formulation for a fixed charge and multistage transportation problem for a single commodity supply chain model. They considered a spanning tree-based GA using Prüfer number representation to solve this problem. Some comparison between results obtained by this method and LINDO showed the efficiency of proposed method. A two-echelon facility location problem was studied by Traganterngsak et al. [9]. In this work, each facility in the second echelon was limited in capacity and could only be supplied by one facility in the first echelon. Also, each customer is serviced by only one facility of the second echelon. The authors presented a mathematical model for the problem and developed a Lagrangian relaxation-based branch-and-bound algorithm to solve it. A different two-stage distribution-planning problem was addressed by Gen et al. [10] to minimize the total cost including transportation and opening costs. They presented a priority-based genetic algorithm (pb-GA) with a new decoding and encoding method. Also they introduced a new crossover operator called Weight Mapping Crossover and analyzed the effect of the latter on the computational performance. They showed the efficiency of proposed method with regards to solution quality and computing time in comparison to two different GA approaches.

Due to legislative changes regarding end-of-life (EOL) [11] issues as well as economic factors [12], considering the forward logistic network and omitting any reverse flow is impractical. A general review of the current developments in reverse logistics was reported by Pokharel and

Mutha [13]. They identified three factors, which differ for reverse logistics and a traditional supply chain: (1) Most logistics networks are not equipped to handle returned product movement; (2) Reverse distribution costs may be higher than moving the original product from the plant to customer; (3) Returned products may not be transported, stored, or handled in the same manner as in the regular channel [14]. In a study by Jayaraman et al. [15], an analytical model to minimize reverse distribution costs was developed. This MILP model limited supply of customer demand from a single distribution center. In addition, there was a tight bound on the number of collection and refurbishing sites. Apart from the formulation of a reverse distribution problem, the authors also presented a new heuristic solution method. The algorithm has three components: random selection of potential collection and refurbishing sites, heuristic concentration and heuristic expansion. Min et al. [16] designed a MINLP model to minimize the overall reverse logistics costs including spatial and temporal consolidation of returned products. The authors presented a mathematical model for the problem and solved it using GA. There are other studies on reverse logistics, which are limited to specific applications, such as carper recycling by Louwers et al. [17] and Realff et al. [18], battery recycling by Schultmann et al. [19] as well as Kannan et al. [20], tire recycling by Figueiredo and Mayerele [21], paper recycling by Pati et al. [22], plastic recycling by Huysman et al. [23], bottle recycling by Shen et al. [24], sand recycling by Listes and Dekker [25]. Notable work with a remanufacturing focus was presented by Krikke et al. [26] on copiers and Srivastava [27] on appliances and personal computers. Currently, no general model for reverse logistics exists.

In recent years, some researches started to integrate forward and reverse networks to close products cycles. The aim is to avoid sub-optimality of a solution due to separated design [28]. Lee and Dong [29] proposed a MILP model, which is capable to manage the forward and reverse flows at the same time for end-of-lease computer products. They presented the first attempt of solving the integrated design problem using a Tabu search-based MA. Lu and Bostel [30] designed a two-level location problem as a MILP model with three types of facility (producers, remanufacturing centers and intermediate centers). This model considers forward and reverse flows and their interactions simultaneously. The focus of this research was on remanufacturing to reduce costs of production and raw materials. The model was solved using Lagrangian heuristics, which requires lower and upper bound of the objective function. Pishvae et al. [31] focused on a MILP model to integrate reverse logistics activities into the forward supply chain. To deal with uncertainty, they presented a scenario-based stochastic optimization model to

minimize the total cost including fixed opening costs, transportation cost, processing costs and penalties for non-utilized capacities. Pishvae et al. [32] proposed a linear multi-objective programming model to improve the total cost as well as responsiveness of the integrated forward/reverse logistics network. To find the set of non-dominated solutions, the authors proposed a solution algorithm based on a new dynamic search strategy using three different local searches. Within the model, they allowed for a hybrid distribution-collection facility. The authors compared their pareto-optimal solutions to recent GA results.

2.2 Flexible delivery path

To increase market share, companies try to increase customer satisfaction via fast delivery, which requires supply chain responsiveness [32, 33]. To deal with the issues of cost efficiency and network responsiveness simultaneously, researchers have proposed models to optimize the supply chain network, respectively [4, 32]. However, results are typically limited to shipments between consecutive stages or just indirect shipment mechanisms [4, 32, 34]. Lin et al. [35], formulated a MILP model by including direct shipment and direct delivery as well as inventory control for a three stages forward logistics network. To solve the problem, they proposed a hybrid evolutionary algorithm composed of Wanger within algorithm, GA, and fuzzy logic controller. Pishvae and Rabbani [36] studied the responsiveness of a three-stage forward logistics network when (1) direct shipment between plant to customer is allowed and (2) direct shipment is forbidden. They proposed two mixed integer programming models for these conditions and proved that both of these problems can be modeled by a bipartite graph. To tackle these NP hard problems, a novel heuristic solution method was considered based on a new chromosome representation derived from graph theory. Based on the above review, extending the following restrictions can be considered a potential field of research:

- Flexibility in delivery paths as a measure to shorten the delivery time is typically ignored for simple and completely ignored for integrated forward/reverse logistics networks.
- The total number of echelons in most of the developed IFRLN models is not more than five echelons.
- It is still a critical need to develop an efficient solution to cope with NP hard problems as well as a general model to be applicable to a wide range of industries.

Within the literature, various facility location models based on mixed integer programming (MIP) were considered to determine maximal profit, optimal number and capacity of facilities as well as the optimal flow between them. Although typically larger models are required to represent

real supply chains, researchers developed many heuristics [7, 15, 34, 37] and metaheuristics such as genetic algorithm [4, 6, 20, 38–42], simulated annealing [43–45], tabu search [29, 46], memetic algorithm [32, 47] and scatter search [34] to solve these NP hard problems. However, there is still a critical need in this area to increase the efficiency of solution approaches, especially, when the complexity of the model increases (Melo et al. [48]).

The problem addressed in this work includes integrated design of forward and reverse logistics as well as flexibility in delivery paths for a seven-stage closed-loop supply chain network. The proposed model as a complete and general network covers the previously described cases in the literature with less complexity. Additionally, the full delivery graph allows us to solve the conflicting goals profit and responsiveness, which otherwise may lead to greater cost [32]. From the computational point of view, we incorporate the graph structure in the chromosome representation, thereby avoiding different model and solution methodologies as, e.g., considered by Pishvae and Rabbani [36].

3 Description for integrated forward/reverse logistics network

To support the presentation of the proposed mathematical model, we consider the general model area of our problem. To this end, we consider $G = (N, E)$ to be a digraph where N denotes the set of all nodes and E the set of all edges in the closed-loop network. The cost for node $i \in N$ are denoted by c_i , and the unit transportation cost on edge $(i, j) \in E$ are given by c_{ij} . The respective decision variables $y_i \in \{0, 1\}$ and $x_{ij} \in \mathbb{N}_0$ represent whether a stage $i \in N$ is used and which quantity is transported between node i and j . To determine the optimal distribution network and capacity of each node, we minimize the transportation and operation cost of the proposed network, which reveals the following mixed integer minimization problem:

$$\begin{aligned} \min_{x_{ij}, y_i} & \sum_{(i,j) \in E} c_{ij}x_{ij} + \sum_{i \in N} c_i y_i \\ \text{s.t.} & \sum_{(i,j) \in E} a_{ij}x_{ij} \leq b_i y_i \\ & x_{ij} \geq 0, y_i \in \{0, 1\} \end{aligned} \tag{1}$$

Next, we specialize this model to reflect the IFRLN properties.

3.1 Mathematical formulation and assumptions

The previously described IFRLN setting represents an integrated supply chain with seven echelons consisting of suppliers S , plants P , distribution centers Dc , retailers R and

customers C in forward flow, as well as collection/inspection centers Co and disposal centers Di in reverse flow, cf. Fig. 2 for a schematic sketch. We like to point out that in accordance to Fig. 1, we consider a hybrid manufacturing-recovery-recycling facility as well as a hybrid collection-inspection facility. Establishing several facilities at the same location can decrease the price in comparison with separating design.

To adapt problem (1), we impose the following assumptions:

- The set of nodes is given by $N = S \cup P \cup Dc \cup R \cup C \cup Co \cup Di$.
- There are no edges between facilities of the same stage, the delivery graph is complete and the return graph is simple, i.e., $E = (S \times P) \cup (P \times Dc) \cup (P \times R) \cup (P \times C) \cup (Dc \times R) \cup (Dc \times C) \cup (R \times C) \cup (C \times Co) \cup (Co \times Di) \cup (Co \times P)$.
- The demands of each customer are deterministic and must be satisfied.
- The number of facilities per stage and respective capacities are limited.
- All cost parameters (fixed and variables) are known in advance.
- The transportation rates are perfect and there are no storages. Moreover, the return rate p_j^{return} as well as the recovery and disposal rates p_j^{disposal} and $(1 - p_j^{\text{disposal}})$ are fixed. All returned products from each customer must be collected.
- The inspection cost per item for the returned products are included in the collection cost.
- The un-recyclable returned products will be sent to the disposal center. The remaining products are returned to the same plant.
- The required recycled materials are assumed to be of the same quality as the raw materials bought from suppliers and any plant chooses the raw material from the collection/inspection center over suppliers.
- Customers have no special preference. It means, price is the same in all facilities.

The objective of this model is to minimize the total cost of the proposed supply chain, which is composed of fixed costs for facilities and variable costs for transportation. In terms of the above notation, the cost function, the sign and the integer conditions remain identical. The constraints in

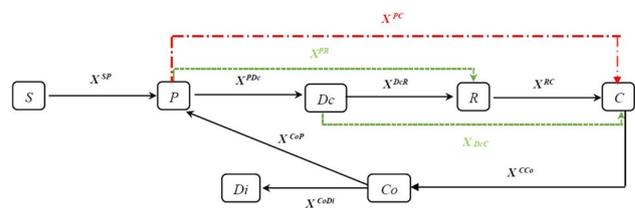


Fig. 2 Underlying structure of MILP

(1) are specialized and we have that the capacities in each node induce the inequalities

$$\sum_{(i,j) \in E} x_{ij} \leq \begin{cases} b_i & \forall i \in S \\ b_i y_i & \forall i \in N \setminus \{S \cup C\}. \end{cases} \tag{2}$$

Additionally, by assumption only a fraction p^{return} is returned by customers and a fraction p^{disposal} of the returned products has to be disposed off. Apart from these exceptions, the supply chain network is subject to the law of the flow conservation, i.e., in-flow and out-flow in each node must be identical for these nodes. These conditions reveal

$$\sum_{(j,k) \in E} x_{jk} = \begin{cases} \sum_{(i,j) \in E} x_{ij} & \forall j \in N \setminus \{C \cup Co\} \\ p_j^{\text{return}} \sum_{(i,j) \in E} x_{ij} & \forall j \in C \\ p_j^{\text{disposal}} \sum_{(i,j) \in Co \times Di} x_{ij} & \forall j \in Co \\ (1 - p_j^{\text{disposal}}) \sum_{(i,j) \in Co \times P} x_{ij} & \forall j \in Co \end{cases} \tag{3}$$

Last, the demands of customers must be satisfied.

$$\sum_{(i,j) \in E} x_{ij} = b_j \quad \forall j \in C \tag{4}$$

4 Solution approach

Because of our IFRLN model is a capacitated allocation and multi-choice problem, it is known as a NP hard problem [6, 38, 39, 49]. Hence, although the problem can be reformulated into an integer linear programming, we cannot compute a suitable solution for large-scaled problems within a short time. There are three main options to tackle NP hard problems: probabilistic algorithms, approximation algorithms and metaheuristic algorithms. To reduce the search space and increase the solution quality, we consider the class of metaheuristic algorithm to solve this model. According to [50], memetic algorithms are appropriate for the proposed model. The basic feature of MA is a multi-directional and global search by generating a population of solutions as well as local search to improve intensification of the search. According to the reviewed literature, two major issues affect the performance of memetic algorithm [41], i.e., the chromosome representation and the memetic operators.

4.1 Chromosome representation

A chromosome must have the necessary gene information for solving the problem. Selecting a proper chromosome representation highly affects the performance of metaheuristic algorithm. Therefore, the first step of applying MA

to a specific problem is to decide how to design a chromosome. The tree-based representation is known to be one way for representing network problems. Different methods have been developed to encode trees. One of them is matrix-encoding, was is developed by Michalewicz [51]. In this method, the solution is presented by a $|K| \cdot |J|$ matrix where $|K|$ and $|J|$ are the number of sources and depots, respectively. Although this solution approach has a simple representation, applying this method requires the development of a special crossover and mutation operator for obtaining a feasible solution as well as huge amount of memory. Another tree-based representation is the Prüfer number. The use of the Prüfer number representation for solving various network problems was introduced by Gen and Cheng [52]. It requires an array of the length $|K| + |J| - 2$ with $|K|$ sources and $|J|$ depots. Since this method may compute infeasible solutions [39], a repair mechanism has been developed. In this regard, Jo et al. [39] presented the procedure for repairing infeasible chromosomes. Later, Gen et al. [53] introduced determinant encoding using priority which does not need any repair mechanism to guarantee feasibility of solutions. In this approach, solutions are encoded as arrays of size $|K| + |J|$, in which the position of each cell represents the sources and depots and the value in cells represent the priorities.

From the literature [41], we have found that both Prüfer and determinant encoding are efficient for the encoding of the spanning tree problem. However, as the determinant encoding overcomes the bottlenecks of Prüfer encoding [54], we utilize determinant encoding in our study. In the following encoding and decoding are discussed.

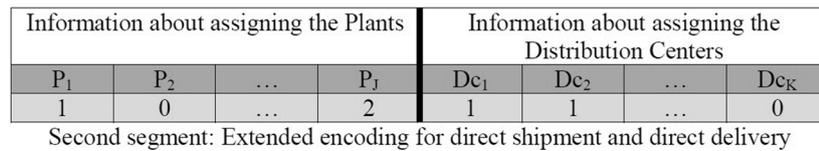
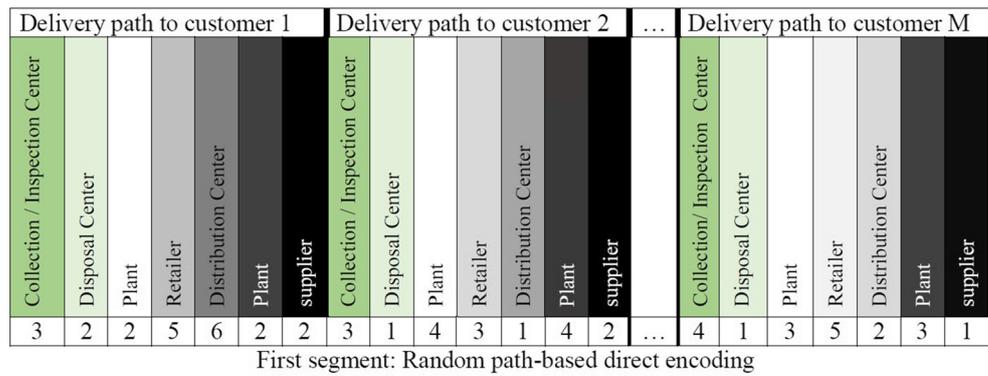
4.1.1 Random path-based direct encoding method

The delivery and recovery path can be conventionally determined by applying the random path direct encoding method introduced by Lin et al. [55]. Using this method computation time can be greatly cut down. One 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 this method, each gene is initialised with a random value from its domain and it contains M groups where M is the total number of customers. Each group represents a delivery path in forward flow as well as recovery path in reverse flow. Due to existence of three different delivery paths in the proposed problem, we extend the random path-based direct encoding method by adding a second segment into the chromosome.

4.1.2 Extended random path-based direct encoding

Although applying the new delivery paths improves the flexibility and efficiency of the supply chain network, it makes the problem more complex. In Fig. 3 the

Fig. 3 Representation of extended random path-based direct encoding method



representation of the extended random path-based direct encoding method in two segments is shown. The first segment is encoded by using random path-based direct encoding method which shows the delivery path for each customer. The second segment of a chromosome contains two parts: the first part with J locus including the guide information regarding plant assignments in the network, and the second part of length K containing the information of the distribution centers. As shown in Fig. 3, the length of chromosome is $(7 * M) + J + K$ where M , J and K are the total number of customers, plants and distribution centers, respectively. Each sequence of seven subsequent genes forms a group. Each group encodes four potential delivery paths through plant, distribution center and retailer to customer as well as a recovery path from customer through collection/inspection to disposal center or plant. The first three alleles of a group represent the reverse flow of the network, while the next four alleles of that group show the forward flow from supplier to customers. As an illustration, a randomly assigned ID to these facilities in the reverse and forward flow is shown in Fig. 3. Each locus in the second part is assigned an integer in the set $\{0, 2\}$ for plants due to existence of three delivery options for each plant in the network. Regarding distribution center, an integer from $\{0, 1\}$ is chosen to represent the two respective delivery options. The second segment is involved by determining the sort of delivery path for the selected plant as well as distribution center in first segment.

It should be noted that applying this encoding approach might generate infeasible solutions, which violate the facility capacity constraint; hence, a repairing procedure is needed. If the total demand of a depot from a source exceeds its capacity, the depot will be assigned to another source with sufficient product supply so that the transportation cost between that source and the depot is the

lowest. The procedure of encoding by extended random path-based direct encoding is shown in Algorithm 1 below.

Algorithm 1 The pseudocode procedure of initialization by extended random path-based direct encoding

```

Input: Number of customers  $M$ 
          Number of collection/inspection centers  $N$ 
          Number of disposal centers  $O$ 
          Number of plants  $J$ 
          Number of retailers  $L$ 
          Number of distribution centers  $K$ 
          Number of suppliers  $I$ 

Step 1:                                     ▷ (first segment)
1: for  $i = 0 : M - 1$  do
2:    $ch_k[7 * i + 1] \leftarrow random(1, N)$ 
3:    $ch_k[7 * i + 2] \leftarrow random(1, O)$ 
4:    $ch_k[7 * i + 3] \leftarrow random(1, J)$ 
5:    $ch_k[7 * i + 4] \leftarrow random(1, L)$ 
6:    $ch_k[7 * i + 5] \leftarrow random(1, K)$ 
7:    $ch_k[7 * i + 6] \leftarrow ch_k[7 * i + 3]$ 
8:    $ch_k[7 * i + 7] \leftarrow random(1, I)$ 
9: end for
Step 2:                                     ▷ (second segment, plant delivery path)
10: for  $i = 0 : J - 1$  do
11:    $ch_k[7 * M + i] \leftarrow random(0, 2)$ 
12: end for
Step 3:                                     ▷ (second segment, Dc delivery path)
13: for  $i = 0 : K - 1$  do
14:    $ch_k[7 * M + J + i] \leftarrow random(0, 1)$ 
15: end for
Output: Chromosome  $ch_k[\cdot]$ 
    
```

Remark 1 According to the assumption presented in Sect. 3, returned products have to be directed to the

original plant. To follow this limitation, the third and sixth position of first segment of the chromosome representation for any customer should be identical.

Remark 2 Since, the third and sixth position of first segment are identical, 6 has been considered as the number of each unit, instead of 7, to apply the chromosome representation.

4.1.3 Extended random path-based direct decoding

Decoding is the mapping from chromosomes to candidate solution to the problem. As an example, Fig. 4 represents an instance of a delivery and recovery path in our model.

In each gene unit, four delivery paths can be designed by applying normal delivery, direct shipment and direct delivery. All of them are from a neighborhood. For instance, we can obtain the neighborhood given in Algorithm 1 from the sample of gene unit shown in Fig. 4 that shows the delivery path to customer 2. Considering the second chromosome (customer 2) in Fig. 4 as an example, we start by supplier 2 and continue via plant 4, distribution center 1 and retailer 3 in forward flow as well as collection/inspection center 3, disposal center 1 and plant 4 in the reverse flow. Due to construction, four different delivery paths are possible, cf. Figure 4. The delivery and recovery path 1 occurs if normal delivery is chosen for all stages. By

skipping distribution centers, path number 2 is selected. Similarly, path number 3 is chosen if retailers are skipped. Last, if direct shipment is selected, the delivery path number 4 will be implemented.

An important difference between the traditional random path-based direct decoding method and the method adopted in this paper is that we include the delivery path information of the second segment. The detailed decoding procedure is shown in Fig. 5. Each locus in this segment is assigned to an integer in the range of $\{0, 1, 2\}$ for plants and $\{0, 1\}$ for distribution centers. Here, we encode normal delivery for plants and distribution centers by $P_j = 0$ and $D_{C_k} = 0$ respectively, where j and k denote the ID of the plant and of the distribution center. Moreover, $P_j = 0$ and $D_{C_k} = 1$ as well as $P_j = 1$ represent direct delivery and $P_j = 2$ direct shipment. The paths displayed in Figure 5 correspond to respective choices, i.e., we have

- Path1 $\iff P_j = 0, D_{C_k} = 0$
- Path2 $\iff P_j = 1, D_{C_k} \in \{0, 1\}$
- Path3 $\iff P_j = 0, D_{C_k} = 1$
- Path4 $\iff P_j = 2, D_{C_k} \in \{0, 1\}$.

It should be noted that because the amount of returned products shipped to each one should be known for decoding the forward flow, decoding of the forward flow is impossible until the reverse flow is decoded.

Fig. 4 Delivery path for a sample of gene unit

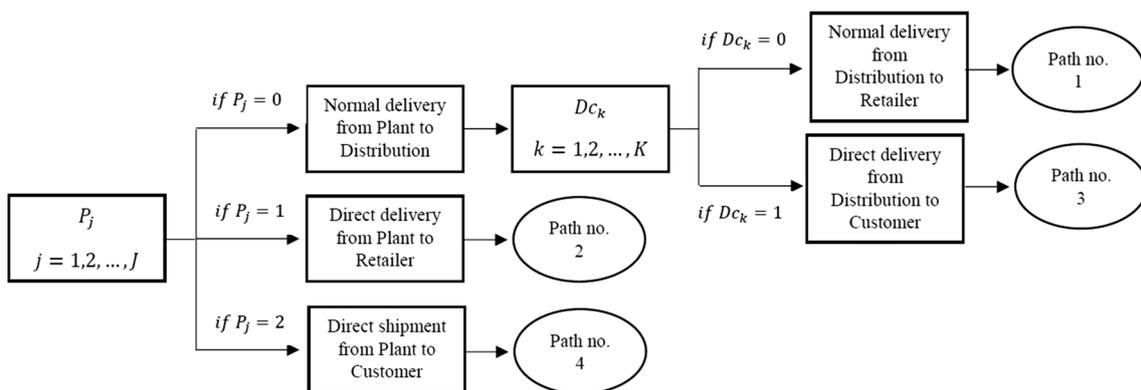
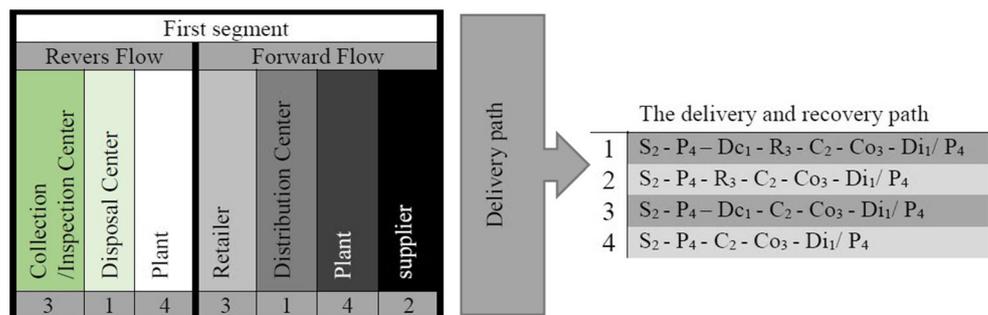


Fig. 5 Presentation of the second segment of the extended random path-based direct encoding

4.2 Evaluation

Fitter individuals have higher chances of survival. The evaluation assigns a fitness value to each individual, thereby inducing a measure. In our study, we apply the cost function as the fitness value. This fitness value is computed for the decoded chromosome to analyze the accuracy and efficiency of the proposed MA.

4.3 Crossover and Local Search

Crossover is known as the most important recombination of both parents' feature to explore new solution within the search space. There are many variants of crossover operations developed in the literature, cf. [10]. Based on the characteristics of the chromosome, we chose the two-cut point crossover, which applies the steps shown in Algorithm 2.

Algorithm 2 Pseudocode of the two point crossover for the proposed model

Input: Two parents

- 1: Generate two random positions
- 2: Swap data beyond the two points between parents

Output: Two offsprings

After crossover, the population is merged and sorted according to its fitness value. In the next step, a *local search technique* is applied, i.e., if the fitness value of a new solution is better than the old one, the latter is replaced. The detailed procedure is shown in Algorithm 3. The chromosome showing the best fitness is selected.

Algorithm 3 Pseudocode of the local search for the proposed model

Input: One parent A

Number of customers M

Number of collection/inspection centers N

Number of disposal centers O

Number of plants J

Number of retailers L

Number of distribution centers K

Number of suppliers I

- 1: Randomly select position a in chromosome of A

- 2: $b \leftarrow \text{mod}(a, 7)$

- 3: $X \leftarrow \begin{cases} I & \text{if } b = 0 \\ N & \text{if } b = 1 \\ O & \text{if } b = 2 \\ J & \text{if } b \in \{3, 6\} \\ L & \text{if } b = 4 \\ K & \text{if } b = 5 \end{cases}$

- 4: $n \leftarrow \text{round}(\text{random}(30/100 * X, 70/100 * X))$

- 5: **for** $i = 1 : n$ **do**

- 6: $A_i \leftarrow A$

- 7: $c \leftarrow \text{random}(1, X)$

- 8: $A_i(a) \leftarrow c$

- 9: Evaluate fitness function for A_i

- 10: **end for**

- 11: Select best chromosome among n new instances

Output: One offspring

4.4 Selection method

We apply the well-known roulette wheel selection for generating the next generation of chromosomes. The strategy of roulette wheel is a probabilistic selection based on fitness.

4.5 Procedure of proposed memetic algorithm

Combing the aforementioned components, we obtain the procedure displayed in Algorithm 4 for solving our problem IFRLN.

Algorithm 4 Pseudocode of the proposed memetic algorithm

```

Input: Number of population  $n$ 
          Number of crossover population  $m$ 
1: for  $k = 1 : n$  do
2:   Encode  $ch_k[\cdot]$  by Algorithm 1
3:   Evaluate  $ch_k[\cdot]$  according to fitness function
4: end for
5: Set  $i \leftarrow 0$ 
6: while termination condition not satisfied do
7:   for  $k = n + 1 : n + m$  do
8:     Select two parents via roulette wheel
9:     Generate  $ch_k[\cdot]$  by Crossover Algorithm 2
10:  end for
11:  Merge  $ch[\cdot] = \bigcup_{k=1:n+m} ch_k[\cdot]$ 
12:  Evaluate and sort  $ch[\cdot]$  by fitness value
13:  Select first  $n$  elements of  $ch[\cdot]$ 
14:  Obtain  $\overline{ch}[\cdot]$  via Algorithm 3 with  $ch_1[\cdot]$ 
15:  if fitness value of  $\overline{ch}[\cdot]$  is better than of  $ch_1[\cdot]$ 
16:    then
17:       $ch_1[\cdot] \leftarrow \overline{ch}[\cdot]$ 
18:    end if
19:   $i \leftarrow i + 1$ 
20: end while
Output: Chromosome of optimal solution  $ch_1[\cdot]$ 

```

Note that as we apply only one crossover and search step before selecting the next generation, our method belongs to the class of steady state MA.

5 Test problems and computational results

To assess the accuracy and efficiency of the developed MA, we generated various test problems of different sizes to compare the results obtained by our MA from Algorithm 4 and a branch-and-bound algorithm from LINGO15. Since the logistics network framework in this study differs from previous studies, the size of test problems considered in this work is selected randomly as shown in Table 1. The first six test problems relatively small and the number of decisions variables are 128, 209, 234, 468, 1006 and 1780, respectively, and the remaining problems are large sized. Other parameters are generated randomly using uniform distributions shown in Table 2. Each test problem has been solved 20 times to test the robustness of the method.

Our implementation was written in MATLAB R2015b and run on the PC with Intel® Core™ i5 2.40 GHz with 12 GB RAM. For our testing, we considered population size $n = 60$ and crossover rate $c_r = 0.5$. As a stopping criterion for Algorithm 4, we imposed a maximum iteration number of 100 as well as a maximum number of iteration without improvement 8, 10, 12, 20, 25 and 30 for our small size problems, respectively. For the large size problems, we increased the latter bound by 5 for each test problem. Also, we set the number of local search iterations to be equal to the number of retailers L for each test problem.

To evaluate the performance of proposed MA, firstly, we employed LINGO15 to solve the optimization problem and obtained the results displayed in Table 3. Although LINGO provides results for small size problems quickly, Table 3 indicates that LINGO is unsuitable for solving the large size problems and it is run out of memory.

According to Table 4, the proposed MA provides good solutions for our small size problems, which allows us to trust the method also for large size problems. To compare

Table 1 Settings of test problems

Problem	Supplier	Plants	Distribution centers	Retailers	Customers	Col/Ins centers	Disposal centers
1	2	2	5	8	2	2	1
2	2	3	8	9	3	3	2
3	2	4	6	10	2	2	1
4	2	4	10	16	4	4	2
5	3	6	15	24	6	6	2
6	4	8	20	32	8	8	4
7	6	12	30	48	12	12	6
8	8	16	40	64	16	16	8
9	12	24	40	96	24	24	12
10	16	32	40	128	32	32	16
11	24	48	60	192	48	48	24
12	32	64	80	256	64	64	32

Table 2 Parameters values used in the test problems

Parameters	Range
$b_j, j \in S$	Uniform (200, 1100)
$b_j, j \in P$	Uniform (100, 1000)
$b_j, j \in Dc$	Uniform (50, 900)
$b_j, j \in R$	Uniform (50, 850)
$b_j, j \in D$	Uniform (100, 500)
$b_j, j \in Co$	Uniform (20, 100)
$b_j, j \in Di$	Uniform (20, 100)
p_j^{return}	10%
p_j^{disposal}	50%
c_{ij}	Uniform (1,3)
$c_j, j \in P$	Uniform (100, 2500)
$c_j, j \in Dc$	Uniform (100, 2100)
$c_j, j \in R$	Uniform (100, 400)
$c_j, j \in Co$	Uniform (100, 500)
$c_j, j \in Di$	Uniform (50, 400)

Table 3 Results obtained by LINGO

Problem	Problem size	Solution
1	2 · 2 · 5 · 8 · 2 · 2 · 1	2905
2	2 · 3 · 8 · 9 · 3 · 3 · 2	2335
3	2 · 4 · 6 · 10 · 2 · 2 · 1	2345
4	2 · 4 · 10 · 16 · 4 · 4 · 2	1160
5	3 · 6 · 15 · 24 · 6 · 6 · 2	4100
6	4 · 8 · 20 · 32 · 8 · 8 · 4	11365
7	6 · 12 · 30 · 48 · 12 · 12 · 6	–
8	8 · 16 · 40 · 64 · 16 · 16 · 8	–
9	12 · 24 · 40 · 96 · 24 · 24 · 12	–
10	16 · 32 · 40 · 128 · 32 · 32 · 16	–
11	24 · 48 · 60 · 192 · 48 · 48 · 24	–
12	32 · 64 · 80 · 256 · 64 · 64 · 32	–

Table 4 Results for Algorithm 4 with $n = 60$ and $m = 30$ over 20 runs

Test problem	Min-cost	Max cost	Ave cost	Min time (s)	Max time (s)	Ave-time (s)
1	2905	2905	2905	2.3	4.3	3.05
2	2335	2735	2402	4.3	10	6.7
3	2345	2885	2381	6.7	13	9.3
4	1160	1560	1225.5	18	47	32.5
5	4100	4920	4576	19	57	38.45
6	11,365	12415	11821	175	410	275.75
7	17,268	21205	19324	260	430	310.6
8	24,933	30446	26995	570	630	600.25
9	33,555	40043	36571.4	1780	2010	1903
10	51,343	60251	52692.95	3740	4060	3935
11	11,986	15600	13132.2	4100	5600	4700
12	13,400	15804	14227.7	6200	7500	6680

the optimal solutions obtained by LINGO with the results of our MA Algorithm 4, the percentage error is calculated using formula (5).

$$\text{Error percent} = \frac{\text{MA}_{\text{answer}} - \text{LINGO}_{\text{answer}}}{\text{LINGO}_{\text{answer}}} \cdot 100 \tag{5}$$

Based on Table 5, we observe that the error percentages for the small size problems are zero, which indicate the high accuracy of proposed MA. Although the operation time is higher compared to LINGO, our implementation allows us to derive results for the large size problems. Hence, the proposed MA demonstrated that it is capable to prepare sufficiently accurate solution with the efficient computation time for our integrated forward/reverse logistics problem with flexible delivery.

6 Conclusion

In this paper, we focused on a comprehensive mixed integer linear programming formulation for a seven-stage closed-loop network design problem. We applied the extended direct delivery path representation-based memetic algorithm, which was developed for a full delivery graph and a combined forward/reverse logistics design to decrease delivery time and avoid sub-optimal solutions, respectively. The aim of this work is to minimize total cost, which we addressed as allocation problem to find the optimal number and capacity for any facility as well as the optimal transportation flow between facilities. Since the basic problem is NP hard, the combination with flexibility in delivery path makes the search space of the problem much larger and more complex and NP hard as well. Because existing methods are unable to solve this problem, we proposed a MA approach to compute a near optimal solution for large size problems. In this study, we introduced a new chromosome representation for MA to

Table 5 Comparison of results from LINGO and proposed MA

Problem	LINGO		MA		Error percent
	Min-cost	Ave-time (s)	Min-cost	Ave-time (s)	
1	2905	0.1	2905	3.05	0
2	2335	0.12	2335	6.7	0
3	2345	0.12	2345	9.3	0
4	1160	0.14	1160	32.5	0
5	4100	0.16	4100	38.45	0
6	11,365	0.17	11,365	275.75	0

enhance its search ability for the proposed flexible model. We verified correctness of the proposed method using numerical experiments and LINGO15. Additionally, we showed that the method is capable to solve larger size problems which cannot be solved by LINGO.

Apart from costs aspect considered here, other aims such as responsiveness and robustness can be considered in designing integrated forward/reverse logistics network that needs updating the algorithm to be capable to solve multi-objective models. Moreover, to be close to the real-world application, multi-product multi-capacity and multi-period networks with uncertainty as well as considering inventory can be employed. Last, the effect of different parameters on the behavior of the proposed metaheuristic algorithm can be analyzed in depth.

Acknowledgements The authors would like to appreciate the support International Graduate School (IGS) of Bremen University to support, help and advice as well as the Deutscher Akademischer Austausch Dienst (DAAD) for financial support of this research under the GSSP programme of the IGS.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Meade L, Sarkis J, Presley A (2007) The theory and practice of reverse logistics. *Int J Logist Syst Manag* 3(1):56–84
- Üster H, Easwaran G, Akali E, etinkaya S (2007) Benders decomposition with alternative multiple cuts for a multi-product closed-loop supply chain network design model. *Naval Res Logist (NRL)* 54(8):890–907
- Georgiadis M, Tsiakis P, Longinidis P, Sofioglou M (2011) Optimal design of supply chain networks under uncertain transient demand variations. *Omega* 39(3):254–272
- Altıparmak F, Gen M, Lin L, Paksoy T (2006) A genetic algorithm approach for multi-objective potimization of supply chain networks. *Comput Ind Eng* 51:197–216
- Beamon B (1999) Designing the green supply chain. *Logistics* 12(4):332–342
- Govindan K, Soleimani H, Kannan D (2015) Reverse logistics and closed supply chain: a comprehensive review to explore the future. *Eur J Oper Res* 240(3):603–626
- Yeh W (2005) A hybrid heuristic algorithm for the multistage supply chain network problem. *Int J Adv Manuf Technol* 26:675–685
- Syarif A, Yun YS, Gen M (2002) study on multi-stage logistic chain network: a apanning tree-based genetic algorithm approach. *Comput Ind Eng* 43(1):299–314
- Tragantalerngsak S, Holt S, Ronnqvist J (2000) An exact method for two-echelon, single-source, capacitated facility location problem. *Eur J Oper Res* 123:473–489
- Gen M, Altıparmak F, Lin L (2006) A genetic algorithm for two-stage transportation problem using priority-based encoding. *Oper Res-Spektrum* 28:337–354
- Neto JQF, Walther G, Bloemhof J, Nunen AEEV, Spengler T (2010) From closed-loop to sustainable supply cchain: the weee case. *Int J Prod Res* 48(15):4463–4481
- Alumur SA, Nickel S, da Gamad FS, Verter V (2012) Multi-period reverse logistics network design. *Eur J Oper Res* 220(1):67–78
- Mutha SPA (2009) Perspectives in reverse logistics: a review. *Conserv Recycl* 53(4):175–182
- Sarkis J, Darnal N, Nehman G, Priest J (1995) The role of supply chain management within the industrial ecosystem. In: *The 1995 IEEE international symposium on electronics and the environment, Orlando*, pp 229–234
- Jayaraman V, Patterson R, Rolland E (2003) The design of reverse distribution networks: model and solution procedure. *Eur J Oper Res* 150:128–149
- Min H, Ko CS, Ko HJ (2006) The spatial and temporal consolidation of returned pproduct in a closed-loop supply chain network. *Comput Ind Eng* 51(2):309–320
- Louwers D, Kip B, Peters E, Souren F, Flapper S (1999) A facility location allocation model for reusing carpet materials. *Comput Ind Eng* 36:855–869
- Realff M, Ammons J, Newton D (2004) Robust reverse proproduct system design for carpet recycling. *IIE Trans* 36:767–776
- Schulmann F, Engels B, Rentz O (2003) Closed-loop supply cchain for spent batteries. *Interfaces* 33:57–71
- Kannan G, Pokharel S, Kumar P (2009) A hybrid approach using ism and fuzzy tosis for the selection of reverse logistics provider. *Resour Conser Recycl* 54(1):28–36
- Figueiredo J, Mayerle S (2008) Designing minimum-cost recycling collection networks with required throughput. *Transp Res Part E* 44:731–752
- Pati R, Vrat P, Kumar P (2008) A goal programming model for paper recycling system. *Omega* 36:405–417
- Huysman S, Debaveya S, Schaubroeck T, Meester SD, Ardente F, Mathieux F, Dewulf J (2015) The recyclability benefit rate of closed-loop and open-loop systems: a case study on plastic recycling in flanders. *Resour Conserv Recycl* 101:53–60
- Shen L, Worrel E, Patel M (2010) Open-loop recycling: a lca case study of pet bottle-to-fibre recycling. *Resour Conser Recycl* 55(1):34–52

25. Listes O, Dekker R (2005) A stochastic approach to a case study for product recovery network design. *Eur J Oper Res* 160:268–287
26. Krikke H, Harten V, Schuur A (1999) Reverse logistic network redesign for copiers. *OR Spektrum* 21:381–409
27. Srivastava S (2008) Network design for reverse logistics. *Omega* 36:535–548
28. Fleischmann M, Beullens P, Bloemhof-ruwaard JM, Wassenhove L (2001) The impact of product recovery on logistics network design. *Prod Oper Manag* 10:156–173
29. Lee D, Dong M (2007) A heuristic approach to logistics network design for end-of lease computer pproduct recovery. *Transp Res Part E* 44:455–474
30. Lu Z, Bostel N (2007) A facility location model for logistics systems including reverse flows: the case of remanufacturing activities. *Comput Oper Res* 34:299–323
31. Pishvae MS, Jolai F, Razmi J (2009) A stochastic optimization model for integrated forward/reverse logistics network design. *J Manuf Syst* 28:107–114
32. Pishvae MS, Farahani RZ, Dullaert W (2010) A memetic algorithm for bi-objective integrated forward/reverse logistics network design. *Comput Oper Res* 37:1100–1112
33. Kilibi W, Martel A, Guitouni A (2010) The design of robust value-creating supply chain networks: a critical review. *Eur J Oper Res* 203:283–293
34. Du F, Evans G (2008) A bi-objective reverse logistics network analysis for post-sale service. *Comput Oper Res* 35:2617–2634
35. Lin L, Gen M, Wang X (2009) Integrated multistage logistics network design by using hybrid evolutionary algorithm. *Computers and Industrial Engineering* 56:854–873
36. Pishvae MS, Rabbani M (2011) A graph theoretic-based heuristic algorithm for responsive supply chain network design with direct and indirect shipment. *Adv Eng Softw* 42:57–63
37. Amiri A (2006) Designing a distribution network in a supply chain system: formulated and efficient solution procedure. *Eur J Oper Res* 171:567–576
38. Gottlieb J, Paulmann L (1998) Genetic algorithms for the fixed charge transportation problem. In: *Inn IC(Ed), IEEE World Congress on evolutionary computation (I. C. INN, ed.)*, pp 330–335
39. Jo JB, Yinzhen L, Gen M (2007) Nonlinear fixed charge transportation problem by spanning tree-based genetic algorithm. *Comput Ind Eng* 53:290–298
40. Min H, Ko H (2008) The dynamic design of a reverse logistics network from the perspective of third-party logistics service providers. *Int J Prod Econ* 113:176–192
41. Wang H, Hsu H (2010) A closed-loop logistic model with a spanning- tree based genetic algorithm. *Comput Oper Res* 37:376–389
42. Dengiz B, Altiparmak F, Smith AE (1997) Local search genetic algorithm for optimal design of reliable networks. *IEEE Trans Evol Comput* 1(3):179–188
43. Krikpatrick S (1984) Optimizing by simulated annealing: quantitative studies. *J Stat Phys* 34(5):975–986
44. Pishvae MS, Kianfar K, Karimi B (2010) Reverse logistics network design simulated annealing. *Int J Adv Manuf Technol* 47(1):269–281
45. Jayaraman V, Ross A (2003) A simulated annealing methodology to distribution network design and management. *Eur J Oper Res* 144:629–645
46. Sung C, Song S (2003) Integrated service network design for a cross-docking supply chain network. *J Oper Res Soc* 54:1283–1295
47. Moscato P, Cotta C (2003) An introduction to memetic algorithms. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial* 19:131–148
48. Melo M, Nickel S, da Gama FS (2009) Facility location and supply chain management: a review. *Eur J Oper Res* 196:401–412
49. Eckert C, Gottlieb J (2002) Direct representation and variation operators for the fixed charge transportation problem, in parallel problem solving from nature. Springer, Berlin
50. Behmanesh E, Pannek J (2016) Modeling and random path-based direct encoding for a closed loop supply chain model with flexible delivery paths. In: *The 7th IFAC conference on Management and Control of Production and Logistics, (Accepted)*
51. Michalewicz Z, Vignaux G, Hobbs M (1991) A non-standard genetic algorithm for the transportation problem. *ORSA J Comput* 3:307–316
52. Gen M, Cheng R (1997) Genetic algorithms and engineering design. Wiley, New York
53. Gen M, Cheng R, Lin L (2008) Network model and optimization: multiobjective genetic algorithm approach. Springer, Berlin
54. Abuali F, Wainwright R, Schoenefeld D (1995) Determinant factorization: a new encoding scheme for spanning trees applied to the probabilistics minimum spanning tree problem. In: *Eshelman L (ed), Proceedings of the 6th international conference on genetic algorithms*, pp 470–477
55. Gen M, Cheng R (2000) Genetic algorithms and engineering optimization. Wiley, New York