A memetic algorithm for bi-objective integrated forward/reverse logistics network design

Mir Saman Pishvaa, Reza Zanjirani Farahani b,c, Wout Dullaert c,d,e,*

a Department of Industrial Engineering, College of Engineering, University of Tehran, Iran
b Centre for Maritime Studies, National University of Singapore, Singapore, Singapore
c Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran
d Institute of Transport and Maritime Management Antwerp (ITMMA), University of Antwerp, Belgium
e Antwerp Maritime Academy, Antwerp, Belgium

ARTICLE INFO
Available online 6 October 2009
Keywords:
Integrated logistics network
Closed loop supply chain network
Memetic algorithm
Multi-objective optimization

ABSTRACT
Logistics network design is a major strategic issue due to its impact on the efficiency and responsiveness of the supply chain. This paper proposes a model for integrated logistics network design to avoid the sub-optimality caused by a separate, sequential design of forward and reverse logistics networks. First, a bi-objective mixed integer programming formulation is developed to minimize the total costs and maximize the responsiveness of a logistics network. To find the set of non-dominated solutions, an efficient multi-objective memetic algorithm is developed. The proposed solution algorithm uses a new dynamic search strategy by employing three different local searches. To assess the quality of the novel solution approach, the quality of its Pareto-optimal solutions is compared to those generated by an existing powerful multi-objective genetic algorithm from the recent literature and to exact solutions obtained by a commercial solver.

1. Introduction
Logistics network design is one of the most important strategic decisions in supply chain management. Decisions on the number of facilities, their locations and capacities and the quantity of flow between them affect both costs and customer service levels. As such, effective and efficient supply chain design can constitute a sustainable competitive advantage for firms. Since opening and closing a facility is both an expensive and time-consuming process, changing network design is impossible in the short run. Because tactical and operational decisions are conditional upon strategic decisions, the configuration of logistics networks will become a constraint for tactical and operational level decisions [1].

Over the last two decades, many companies such as Kodak and Xerox have focused on remanufacturing and recovery activities and have achieved significant successes in this area [2]. Meade et al. [3] classify the driving forces of this increased interest and investment in reverse logistics into two groups: environmental factors and business factors. The first set includes the environmental impact of used products, environmental legislation and the increasing environmental consciousness of customers. The importance of environmental aspects has prompted researchers to develop models and frameworks for assessing eco-efficiency in logistics networks (e.g. [4]). Business factors are related to the economic benefits of using returned products and liberal return policies for gaining customer satisfaction.

Triggered by these driving forces, reverse logistics network design addresses the number of collection, recovery and disposal centers needed, their location and capacities, buffer inventories and product flows between the facilities. In many cases, logistics networks are designed for forward logistics activities without considering the reverse flow of return products and most of them are not equipped to handle returned products in reverse channels [5].

The configuration of the reverse logistics network, however, has a strong influence on the performance of the forward logistics network and vice versa as they share a number of resources such as, e.g. transport and warehouse capacity. Due to the fact that designing the forward and reverse logistics separately leads to sub-optimal designs with respect to costs, service levels and responsiveness, the design of the forward and reverse logistics networks should be integrated [6–8]. This kind of integration can be considered as “horizontal integration”, as it encompasses the integration of related optimization problems at the same decision level (strategic, tactical or operational). For example, integrating supplier selection with network design or integrating the design...
of the forward and reverse supply chain are two examples of horizontal integration at strategic level. Integrating decision-making processes across decision levels, e.g. by taking tactical/operational inventory levels into account when addressing strategic network design issues, can be referred to as “vertical integration” (see [9] for a comprehensive literature review).

Real world network design problems are often characterized by multiple objectives. The minimization of total costs and maximization of network responsiveness are the most commonly used single objectives in forward logistics network design. These objectives are, however, typically conflicting, and considering them concurrently (see, for example, [10,11]) is the most favorable option for most decision makers. Network responsiveness is an important issue in reverse logistics too (see [12]), as it is undesirable for customers/retailers to keep used products for a long time because of the related holding costs. Since customers have a tendency to discard used products as soon as possible, companies aiming to collect more used products from customers should also consider network responsiveness when minimizing costs.

Previous research in the area of forward, reverse and integrated logistics network design often limited itself to only considering a single capacity level for each facility and often did not address how capacity levels can be determined (see Table 2 and [13]). Nevertheless, capacity levels are important decision variables in real-life applications due to their strong influence on logistics network efficiency and responsiveness.

Based on the aforementioned considerations, this paper addresses the issue of integrated multi-objective, multi-stage forward/reverse logistics network design including production, distribution, collection/inspection, recovery and disposal facilities with multiple capacity levels. The rest of this paper is structured as follows. Section 2 offers a code-based systematic literature review to assess the state-of-the-art in forward/reverse logistics network design. To design integrated forward/reverse logistics networks, a generalized mixed integer non-linear programming (MINLP) formulation and a mixed-integer linear programming (MILP) variant model are developed in Sections 3 and 4. Section 5 presents an efficient memetic algorithm using a dynamic search strategy to find the set of non-dominated solutions for large-scale problems. The computational performance of the heuristic is analyzed in Section 6. Section 7 concludes this paper and offers guidelines for further research.

Table 1 Classification of logistics network design problems.

<table>
<thead>
<tr>
<th>Objectives</th>
<th>Outputs</th>
<th>Modeling</th>
<th>Problem definition</th>
<th>Logistics network stages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max robustness Ro</td>
<td>Inventory I</td>
<td>Continuous (CA)</td>
<td>Multi-period MPri</td>
<td>Distribution centers DisC</td>
</tr>
<tr>
<td>Max responsiveness Res</td>
<td>Number of vehicles NV</td>
<td>Continuous aproximation CA</td>
<td>Single period SP</td>
<td>Production centers PC</td>
</tr>
<tr>
<td>Min cost/max profit C</td>
<td>Demand satisfaction quantity DS</td>
<td>Discrete (SMIP)</td>
<td>Endogenous (undetermined) En</td>
<td>Supply centers SC</td>
</tr>
<tr>
<td></td>
<td>Price of products P</td>
<td></td>
<td>Exogenous (determined) Ex</td>
<td>Reverse logistics stages</td>
</tr>
<tr>
<td></td>
<td>Transportation amount TA</td>
<td></td>
<td>Facility capacity FC</td>
<td>Redistribution centers RDIsC</td>
</tr>
<tr>
<td></td>
<td>Service region SR</td>
<td></td>
<td>Location/allocaion L</td>
<td>Disposal centers DC</td>
</tr>
<tr>
<td></td>
<td>Facility capacity FC</td>
<td></td>
<td></td>
<td>Recycling centers RY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Recovery centers RC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Collection/inspection centers CIC</td>
</tr>
</tbody>
</table>

2. Literature review

Dullaert et al. [14] give a general review of the supply chain design models to support the development of richer supply chain models, capable of taking into account all logistics costs. In this paper, we survey specific network design problems for forward, reverse and integrated supply chain design problems. As the overview of the literature on logistics network design in Tables 1 and 2 shows, mixed integer programming (MIP) models are the common models used in this area. These models range from simple single-product uncapacitated facility location models (e.g. [15]) to complex capacitated multi-commodity models (e.g. [16]) and are usually aimed at determining the cost minimizing or profit maximizing system design.

Because of the increasing importance of network responsiveness in supply chain management, this has recently been considered as a significant additional objective for multi-objective logistics network design (e.g. [10–12]). Along the same lines, the concern about major changes in the business environment (e.g. customer demand, custom duties, natural disasters) has spurred an interest in designing scalable and robust supply chains.

Meepetchdee and Shah [1] minimize system costs whilst imposing constraints on guaranteeing a minimum level of robustness in the MILP model.

Although MIP models are powerful tools for logistics network design, their data and computational requirements increase tremendously as models become more realistic. Thus, data reliability and model accuracy often decrease for larger scale models. Moreover, most of these models belong to the NP-hard class. Thus, when problem size increases, it is very difficult to find the optimal solution in reasonable computation time. To overcome these problems, some researchers advocate continuous models for logistics network design (e.g. [17]). Continuous approximation requires less data and provides an analytical framework to obtain the optimal solution. To learn more about continuous models, we refer the reader to [18,19].

During the last decade, many models were developed for reverse logistics network design. Jayaraman et al. [5] developed an MILP model for reverse logistics network design under a pull
The objective of the proposed model is to minimize total cost. Although forming a general concern, demand uncertainties and quality considerations are of special importance to designing reverse logistics networks. To this end, Listes and Dekker [20] illustrate as early as 2001 that an integrated logistics network design is related to forward logistics network design, a smaller part is associated with reverse logistics network design and in recent years a few papers have dealt with integrated logistics network design.

Table 2

<table>
<thead>
<tr>
<th>Reference articles</th>
<th>Logistics network stages</th>
<th>Problem definition</th>
<th>Modeling</th>
<th>Outputs</th>
<th>Objectives</th>
<th>Solution method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward logistics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sahib and Beamon [24]</td>
<td>SC, PC*, DisC*</td>
<td>Ca, D, UCF, MP, En, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C, Ro</td>
<td>Exact (small sizes)</td>
</tr>
<tr>
<td>Yasarli et al. [26]</td>
<td>PC*</td>
<td>UC, D, UCF, SP, En, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C</td>
<td>Exact (continuous)</td>
</tr>
<tr>
<td>Jayaraman and Ross [27]</td>
<td>PC, DisC*</td>
<td>Ca, D, UCF, MP, En, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C</td>
<td>SA</td>
</tr>
<tr>
<td>Sung and Song [15]</td>
<td>PC, DisC*</td>
<td>UC, D, UCF, SP, En, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C</td>
<td>TS-based heuristic</td>
</tr>
<tr>
<td>Miranda and Garrido [28]</td>
<td>PC, DisC*</td>
<td>Ca, S, UCF, SP, En, SpPr</td>
<td>MINLP</td>
<td>L</td>
<td>C</td>
<td>LR-based heuristic</td>
</tr>
<tr>
<td>Yeh [29]</td>
<td>PC, SDiscC*</td>
<td>Ca, D, UCF, SP, En, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C</td>
<td>Hybrid heuristic</td>
</tr>
<tr>
<td>Melachrinoudis et al. [31]</td>
<td>PC*</td>
<td>Ca, D, UCF, MP, En, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C, Res</td>
<td>Exact (small sizes)</td>
</tr>
<tr>
<td>Amiri [30]</td>
<td>PC, DisC*</td>
<td>Ca, D, UCF, SP, En, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C</td>
<td>LR-based heuristic</td>
</tr>
<tr>
<td>Gen et al. [31]</td>
<td>PC, DisC*</td>
<td>Ca, D, UCF, SP, En, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C</td>
<td>GA</td>
</tr>
<tr>
<td>Altiparmak et al. [10]</td>
<td>SC, PC<em>DisC</em></td>
<td>Ca, D, UCF, SP, En, SpPr</td>
<td>MINLP</td>
<td>L, TA</td>
<td>C, Res</td>
<td>GA</td>
</tr>
<tr>
<td>Meepethee and Shah [1]</td>
<td>PC, DisC*</td>
<td>UC, D, UCF, SP, Ex, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C, Ro</td>
<td>Exact (small sizes)</td>
</tr>
<tr>
<td>Tsaiachs and Papageorgiou [16]</td>
<td>PC*, DisC*</td>
<td>Ca, D, CF, MP, En, SpPr</td>
<td>MILP</td>
<td>L, PC, TA</td>
<td>C</td>
<td>Exact (small sizes)</td>
</tr>
<tr>
<td>Reverse logistics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marin and Pelegrin [31]</td>
<td>PC*</td>
<td>UC, D, UCF, SP, En, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C</td>
<td>Lagrangian decomposition</td>
</tr>
<tr>
<td>Jayaraman et al. [5]</td>
<td>CIC, RCC*, RDiscC*</td>
<td>Ca, D, UCF, MP, En, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C</td>
<td>Exact (small sizes)</td>
</tr>
<tr>
<td>Krikke et al. [32]</td>
<td>CIC, RCC*, RDiscC*</td>
<td>UC, D, UCF, SP, Ex, SpPr</td>
<td>MILP</td>
<td>L, TA, I</td>
<td>C</td>
<td>Exact (small sizes)</td>
</tr>
<tr>
<td>Jayaraman et al. [33]</td>
<td>CIC*, RCC*</td>
<td>Ca, D, UCF, SP, En, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C</td>
<td>Heuristic</td>
</tr>
<tr>
<td>Listes and Dekker [20]</td>
<td>CIC*, RCC*</td>
<td>Ca, D, UCF, MP, En, SpPr</td>
<td>SMIP</td>
<td>L, TA</td>
<td>C</td>
<td>Exact (small sizes)</td>
</tr>
<tr>
<td>Min et al. [34]</td>
<td>CIC*, RCC*</td>
<td>Ca, UCF, SP, En, MPPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C</td>
<td>GA</td>
</tr>
<tr>
<td>Uster et al. [2]</td>
<td>CIC*, RCC*, PC, DiscC*</td>
<td>UC, D, UCF, MP, En, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C</td>
<td>Benders decomposition</td>
</tr>
<tr>
<td>Demirer and Gökçen [35]</td>
<td>CIC*, RCC, PC, DiscC*</td>
<td>Ca, D, UCF, MP, En, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C</td>
<td>Exact</td>
</tr>
<tr>
<td>Aras et al. [22]</td>
<td>CIC*</td>
<td>UC, D, UCF, MP, Ex, SpPr</td>
<td>MINLP</td>
<td>L, TA, NV, P</td>
<td>C</td>
<td>TS-based heuristic</td>
</tr>
<tr>
<td>Integrated logistics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fleischmann et al. [6]</td>
<td>CIC*, RCC*, PC*, DiscC*</td>
<td>UC, D, UCF, SP, En, SpPr</td>
<td>MILP</td>
<td>L, TA, DS</td>
<td>C</td>
<td>Exact (small sizes)</td>
</tr>
<tr>
<td>Lu and Bozdoglu [36]</td>
<td>CIC*, RCC*, DC, PC*, DiscC*</td>
<td>UC, D, UCF, SP, En, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C</td>
<td>LR-based heuristic</td>
</tr>
<tr>
<td>Salema et al. [37]</td>
<td>CIC*, RCC*, PC*, DiscC*</td>
<td>Ca, D, UCF, SP, En, SpPr</td>
<td>MILP</td>
<td>L, TA, DS</td>
<td>C</td>
<td>Exact (small sizes)</td>
</tr>
<tr>
<td>Ko and Evans [23]</td>
<td>RCC, PC, DiscC*</td>
<td>Ca, D, UCF, MP, En, MPPr</td>
<td>MINLP</td>
<td>L, TA</td>
<td>C</td>
<td>GA-based heuristic</td>
</tr>
<tr>
<td>Salema et al. [21]</td>
<td>CIC*, RCC*, PC*, DiscC*</td>
<td>Ca, S, UCF, MP, En, SpPr</td>
<td>SMIP</td>
<td>L, TA, DS</td>
<td>C</td>
<td>Exact (small sizes)</td>
</tr>
<tr>
<td>Min and Ko [38]</td>
<td>CIC*, RCC, PC, DiscC*</td>
<td>Ca, D, UCF, MP, En, MPPr</td>
<td>MINLP</td>
<td>L, TA</td>
<td>C</td>
<td>GA</td>
</tr>
<tr>
<td>Lee and Dong [7]</td>
<td>CIC*, RCC, PC, DiscC*</td>
<td>Ca, D, CF, SP, Ex, SpPr</td>
<td>MILP</td>
<td>L, TA</td>
<td>C</td>
<td>TS-based heuristic</td>
</tr>
</tbody>
</table>

*Facilities to be located in the network.

To structure the literature review on logistics network design and to identify future research avenues, we have classified logistics network design problems according to five general specifications: logistics network stages, problem definition, modeling, outputs and objectives. A coding system is then developed based on Table 1, and models available in the literature in the last decade are coded based on this system in Table 2.

As shown in Table 2, a large part of literature in logistics network design is related to forward logistics network design, a smaller part is associated with reverse logistics network design and in recent years a few papers have dealt with integrated logistics network design.

Since the majority of logistics network design problems can be categorized as NP-hard, many powerful heuristics, metaheuristics and Lagrangian relaxation (LR)-based methods have been developed for solving these models. The solution methods of the reviewed papers are presented in the last column of Table 2.

3. Problem definition

The integrated forward/reverse logistics network (IFRLN) network operated by third party logistics (3PL) providers. To solve the bi-objective MILP model, a hybrid scatter search method is developed. Also Ko and Evans [23] consider a network operated by a 3PL service provider and they present a MINLP model for the simultaneous design of the forward and return network. They develop a genetic algorithm-based heuristic to solve the complex developed model. Finally, Lee and Dong [7] develop an MILP model for integrated logistics network design for end-of-lease computer products. They consider a simple network with a single production center and a given number of hybrid distribution-collection facilities to be opened which they solve using tabu search.
In the reverse flow, returned products are collected in collection/inspection centers and, after testing, the recoverable products are shipped to recovery facilities, and scrapped products are shipped to disposal centers. By means of this strategy, excessive transportation of returned products (especially scrapped products) is prevented and the returned products can be shipped directly to the appropriate facilities. In such an integrated logistics network, hybrid processing facilities offer potential cost savings compared to separate distribution or collection centers [7]. The IFRLN therefore considers a hybrid distribution-collection facility whereby both distribution and collection centers are established at the same location. The resulting cost saving is reflected in the objective function, which considers both the tradeoff of fixed opening costs of facilities and variable transportation costs and the responsiveness of the network. Thus, unlike previous models with hybrid distribution-collection facilities (e.g. [7]), the use of hybrid-collection facilities is a decision variable in the IFRLN model.

In the forward network, products are pulled through a divergent network and in the reverse network, returned products are shipped through a semi-convergent network according to push principles. A predefined percentage of demand from each customer zone is assumed to result in returned products and a pre-defined value is determined as an average disposal rate. All returned products from first market customers must be collected and all demand from second market customers must be satisfied. The recovery process is performed in production/recovery centers and recovered products are inserted in the forward network and are considered identical to new products. Thus, IFRLN is a closed-loop logistics network.

The complete process of the integrated forward/reverse logistics network (IFRLN) under consideration is illustrated in Fig. 2.

With the above situations in mind, the main issues to be addressed by this study are to determine the location, the number and the capacity of production/recovery, distribution, collection/inspection and disposal centers that represent the degree of centralization of the network, and also to determine the product flow between the facilities. IFRLN is not a case-based network and because of its generic nature, it can support a variety of industries.
such as electronic and digital equipment industries (e.g. [7,32]) and vehicle industries (e.g. [22]).

It is important to note that the design of the integrated logistics network may involve a trade-off between the total costs and the network’s responsiveness. In some cases, companies may decide to open more facilities to increase the responsiveness for higher customer satisfaction, which may lead to a greater investment cost. Thus, the IFRLN is designed to jointly take network costs and network responsiveness into account. According to Table 1, the problem in question can be coded as shown in Table 3.

### 4. Model formulation

To support the presentation of the proposed mathematical model, we first provide a verbal description of the model as follows.

**Minimize Costs**

\[ \text{Fixed opening costs} - \text{Savings from integrating facilities} + \text{Transportation Costs} \]

**Maximize Responsiveness**

\[ \text{Forward responsiveness} + \text{Reverse responsiveness} \]

Subject to:

- satisfying all forward and reverse demands,
- balancing of flows between nodes,
- capacity constraints,
- logical constraints related to the different capacity levels,
- non-negativity and binary constraints.

The following notations are used in the formulation of the IFRLN model:

- **Sets**
  - Set of potential production/recovery center locations: \( I = \{i \mid i \in I\} \)
  - Set of potential distribution center locations: \( J = \{j \mid j \in J\} \)
  - Set of fixed locations of customer zones: \( K = \{k \mid k \in K\} \)
  - Set of potential collection/inspection center locations: \( L = \{l \mid l \in L\} \)
  - Set of potential disposal center locations: \( M = \{m \mid m \in M\} \)
  - Set of capacity levels available for facilities: \( N = \{n \mid n \in N\} \)
  - Set of joint potential sites between collection/inspection centers and distribution centers: \( E = \{e \mid e \in E, E \subset J, E \subset L\} \)

- **Parameters**
  - \( d_k \): demand of customer zone \( k \)
  - \( r_k \): rate of return of used products from customer zone \( k \)
  - \( s \): average disposal fraction
  - \( f_{ini} \): fixed cost of opening production/recovery center \( i \) with capacity level \( n \)
  - \( c_{ini} \): fixed cost of opening distribution center \( j \) with capacity level \( n \)
  - \( h_{ini} \): fixed cost of opening collection/inspection center \( l \) with capacity level \( n \)
  - \( a_{mi} \): fixed cost of opening disposal center \( m \) with capacity level \( n \)
  - \( f_{e}^{mr} \): fixed saving cost associated with opening distribution center with capacity level \( n \) at location \( l \)
  - \( c_{xij} \): shipping cost per unit of products from production/recovery center \( i \) to distribution center \( j \)
  - \( c_{uki} \): shipping cost per unit of products from distribution center \( j \) to customer zone \( k \)
  - \( c_{qji} \): shipping cost per unit returned products from customer zone \( k \) to collection/inspection center \( l \)
  - \( c_{pil} \): shipping cost per unit of recoverable products from collection/inspection center \( l \) to production/recovery center \( i \)
  - \( c_{tm} \): shipping cost per unit of scrapped products from collection/inspection center \( l \) to disposal center \( m \)
  - \( c_{an} \): capacity of production with level \( n \)
  - \( c_{an} \): capacity of collection/inspection center \( l \)
  - \( c_{an} \): capacity of disposal center \( m \)
  - \( \beta \): expected collection time in reverse network
  - \( \rho \): weighting factor (importance) for the forward responsiveness in second objective function; \((1-\rho)\) denotes the weight for the reverse responsiveness

**Variables**

- \( X_{xij} \): quantity of products shipped from production/recovery center \( i \) to distribution center \( j \)
- \( U_{xij} \): quantity of products shipped from distribution center \( j \) to customer zone \( k \)
- \( Q_{xij} \): quantity of returned products shipped from customer zone \( k \) to collection/inspection center \( l \)
- \( P_{xij} \): quantity of recoverable products shipped from collection/inspection center \( l \) to production/recovery center \( i \)
- \( T_{xij} \): quantity of scrapped products shipped from collection/inspection center \( l \) to disposal center \( m \)

- \( W_{xij}^{n} \): with capacity level \( n \) is opened at location \( i \),
- \( Y_{xij}^{n} \): with capacity level \( n \) is opened at location \( j \),
- \( Z_{xij}^{n} \): with capacity level \( n \) is opened at location \( l \),
- \( V_{xij}^{n} \): with capacity level \( n \) is opened at location \( m \),

### Table 3

Coding of the problem in question.

<table>
<thead>
<tr>
<th>Objectives</th>
<th>Outputs</th>
<th>Modeling</th>
<th>Problem definition</th>
<th>Logistics network stages</th>
</tr>
</thead>
<tbody>
<tr>
<td>C, Res</td>
<td>L, FC, TA</td>
<td>MILP</td>
<td>Ca, D, UCF, SP, En, SPt</td>
<td>CLC*, RCC*, DiSC*, DC*, PC*</td>
</tr>
</tbody>
</table>
According to above notations, the structure of the IFRLN can be presented as follows (Fig. 3).

In terms of the above notation, the multi-objective IFRLN design problem can be formulated as follows:

Min \( W_1 = \sum_{i \in I} \sum_{j \in J} \sum_{n \in N} f_{jn}^m w_{jn}^m + \sum_{j \in J} \sum_{n \in N} q_{jn}^m v_{jn}^m + \sum_{i \in I} \sum_{j \in J} \sum_{n \in N} h_i^m z_{in}^m + \sum_{m \in M} \sum_{n \in N} a_m^m v_m^n + \sum_{k \in K} \sum_{n \in N} \sum_{l \in L} c_{ol}^k q_{ol}^k + \sum_{i \in I} \sum_{j \in J} c_{ij}^k u_{ij}^k \quad (1) \)

Max \( W_2 = \frac{\sum_{j \in J} \sum_{k \in K} U_{jk}}{\sum_{k \in K} d_k} + (1 - \rho) \left( \sum_{k \in K} Q_{ol}^k \right) \) \quad (2)

\( \sum_{j \in J} U_{jk} = d_k \quad \forall k \in K \) \quad (3)

\( \sum_{k \in K} Q_{ol}^k = r_{ol}^k d_k \quad \forall k \in K \) \quad (4)

\( \sum_{j \in J} U_{jk} \leq \sum_{k \in K} U_{jk} \quad \forall j \in J \) \quad (5)

\( \sum_{m \in M} T_{lm} = s \sum_{l \in L} Q_{ol}^k \quad \forall l \in L \) \quad (6)

\( \sum_{l \in L} P_{li} = (1 - s) \sum_{k \in K} Q_{ol}^k \quad \forall l \in L \) \quad (7)

\( \sum_{j \in J} W_{in}^m c_{av}^m \leq \sum_{n \in N} W_{in}^m \quad \forall i \in I \) \quad (8)

\( \sum_{n \in N} X_{jn} \leq \sum_{j \in J} \sum_{n \in N} W_{in}^m c_{av}^m \quad \forall j \in J \) \quad (9)

\( \sum_{k \in K} U_{jk} \leq \sum_{j \in J} \sum_{n \in N} Y_{jn}^m c_{av}^m \quad \forall j \in J \) \quad (10)

\( \sum_{k \in K} Q_{ol}^k \leq \sum_{n \in N} Z_k^m c_{av}^m \quad \forall l \in L \) \quad (11)

\( \sum_{l \in L} T_{lm} \leq \sum_{n \in N} V_l^m c_{av}^m \quad \forall m \in M \) \quad (12)

\( \sum_{l \in L} P_{li} \leq \sum_{n \in N} W_{in}^m c_{av}^m \quad \forall i \in I \) \quad (13)

\( \sum_{m \in M} T_{im} + \sum_{l \in L} P_{li} \leq \sum_{n \in N} Z_k^m c_{av}^m \quad \forall l \in L \) \quad (14)

\( \sum_{j \in J} X_{jn} \leq \beta \sum_{j \in J} U_{jk} \quad \forall i \in I \) \quad (15)

\( \sum_{n \in N} W_{in}^m \leq 1 \quad \forall i \in I \) \quad (16)

\( \sum_{n \in N} Y_{jn}^m \leq 1 \quad \forall j \in J \) \quad (17)

\( \sum_{n \in N} Z_k^m \leq 1 \quad \forall l \in L \) \quad (18)

\( \sum_{m \in M} V_l^m \leq 1 \quad \forall m \in M \) \quad (19)

\( W_n^m, Y_n^m, Z_k^m, V_l^m \in \{0, 1\} \quad \forall i \in I, j \in J, l \in L, m \in M, n \in N \) \quad (20)

Objective function (1) minimizes the total costs including fixed opening costs, transportation costs and the cost savings associated with integrating distribution and collection/inspection centers at the same locations. Objective function (2) maximizes the forward and reverse responsiveness of the integrated network. This objective function is adopted from the formulation of Altiparmak et al. [10] for a forward logistics network and its value is bounded between 0 and 2. Constraints (3) and (4) ensure that the demands of all customers are satisfied and the returned products from all customer zones are collected. Eqs. (5)–(7) assure the flow balance at production/recovery, distribution, collection/inspection, disposal and customer centers. Constraints (8)–(15) are capacity constraints on facilities, which also prohibit the units of products, returned products, recoverable and scrapped products from being transferred to facilities which are not opened. Constraints (16)–(19) ensure that a facility can be assigned at most one capacity level. Finally, Constraints (20) and (21) enforce the binary and non-negativity restrictions on the corresponding decision variables.

The term \( \sum_{i \in I} \sum_{j \in J} \sum_{n \in N} f_{jn}^m W_{jn}^m + \sum_{j \in J} \sum_{n \in N} q_{jn}^m V_{jn}^m + \sum_{i \in I} \sum_{j \in J} \sum_{n \in N} h_i^m Z_{in}^m + \sum_{m \in M} \sum_{n \in N} a_m^m V_m^n \) in the first objective function is non-linear because it involves the multiplication of two binary variables. To avoid the complexity of such mixed integer non-linear programming (MINLP) model, the above model is linearized by defining a new variable and reformulating the objective function as follows:

\( Q_{in}^m = Z_{kn}^m \cdot V_{kn}^m \)

\( Q_{in}^m \in \{0, 1\} \quad \forall e \in E, m \in M, n \in N \) \quad (22)

Min \( W_1 = \sum_{i \in I} \sum_{j \in J} \sum_{n \in N} f_{jn}^m W_{jn}^m + \sum_{j \in J} \sum_{n \in N} q_{jn}^m Y_{jn}^m + \sum_{i \in I} \sum_{j \in J} \sum_{n \in N} h_i^m Z_{in}^m + \sum_{m \in M} \sum_{n \in N} a_m^m V_m^n + \sum_{k \in K} \sum_{n \in N} c_{ol}^k Q_{ol}^k + \sum_{i \in I} \sum_{j \in J} c_{ij}^k u_{ij}^k \quad (23) \)

Because the objective function minimizes costs, it has a tendency to put the value of \( Q_{in}^m \) variable to 1. Therefore, we should only prohibit the value of \( Q_{in}^m \) to be 1 in three conditions: when both of \( Z_{kn}^m \) and \( V_{kn}^m \) or one of them is equal to zero. This can be achieved by adding the following constraint to the IFRLN model:

\( 2Q_{in}^m \leq Z_{kn}^m + V_{kn}^m \quad \forall e \in E, m \in M, n \in N \) \quad (24)

The resulting model is a multi-objective MILP with \((I+JK+KL+LM+LI)\) continuous variables and \(N(I+J+L+M+EN)\) binary variables. The number of constraints is \(2(2I+2J+K+M)+5L+EN^2\), excluding constraints (20) and (21). Thus, \( N \) plays a significant role in increasing the complexity of the IFRLN model.

The two objective functions are in conflict with each other as an increase of responsiveness leads to an increase in total costs and vice versa as illustrated in Fig. 4. Fig. 4 shows the Pareto front solutions obtained by LINGO 8.0 for an illustrative example, The Pareto front solutions are obtained by first solving the original model for each of the objective functions separately and second by solving a model in which one of the objective functions is replaced.
by a side constraint imposing the optimal value obtained in the first step. In doing so, two extreme points of the Pareto frontier are obtained. Then one of the objective functions is randomly selected to be transformed into a side constraint with a right-hand side variable randomly chosen in the range of its two extreme values. The resulting model is solved to obtain the corresponding optimal value for other objective function. As shown in Fig. 4, the first objective has a tendency towards network centralization to minimize the costs; on the other hand the second objective has a tendency towards network decentralization to maximize the responsiveness of the logistics network.

As the IFRLN design problem includes the capacitated plant location problem which is known to be NP-complete [39], the IFRLN design problem is NP-hard.

5. Solution approach

The inherent characteristics of genetic algorithms (GAs) demonstrate why GAs may be an appropriate approach for multi-objective optimization. The basic feature of GAs is multiple directional and global search by maintaining a population of solutions from generation to generation. The population-based nature of GA is useful when exploring Pareto solutions [40]. However, pure GAs often lack the capability of sufficient search intensification. Moscato and Norman [41] first defined a memetic algorithm (MA) to integrate local search in GAs to improve the intensification of the search. MAs are population-based heuristic search approaches for optimization problems similar to GAs with often additional local searches proved to be of practical success in a variety of problem domains and in particular for the approximate solution of NP-hard optimization problems [42]. MAs combine the advantages of efficient heuristics incorporating domain knowledge and population-based search approaches such as GAs. In recent years MAs were used in variety of optimization problems such as production–distribution problems (e.g. [43]), scheduling problems (e.g. [44]), minimum span frequency assignment problems (e.g. [45]) and partitioning problems (e.g. [46]). In this paper, a multi-objective memetic algorithm (MOMA) with dynamic local search is developed to solve the IFRLN problem as detailed below.

5.1. Chromosome representation

Different methods have been developed to encode trees; one of them is the matrix encoding method attributed to Michalewicz et al. [47]. In this method, the solution is presented by a $|K| \times |J|$ matrix in which $|K|$ denotes the number of sources and $|J|$ the number of depots. In spite of its simple representation, the implementation of this method requires the development of special operators as well as huge memory space.

Gen and Cheng [40] develop a spanning tree method that employs Prüfer numbers [48] for solution representation. In this method, solutions are represented by arrays of length $|K| + |J| - 2$. As this method may result in infeasible solutions, repair mechanisms are developed. As an alternative, Gen et al. [30] present a priority-based encoding method that does not need a repair mechanism. 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.

To apply the priority-based encoding method to the IFRLN, some modifications of the Gen et al. [30] method are needed. First, as each facility in the IFRLN potentially has more than one capacity level, one row should be added to the solution array to present the capacity level of each facility. Thus, in this approach solutions are encoded as $2(|K| + |J|)$ matrix. Moreover, the Gen et al. [30] priority-based decoding algorithm should be modified with respect to IFRLN model assumptions. The modified algorithm is presented below.

Algorithm 1: Modified priority-based decoding

**Inputs:** $K$: set of sources

$J$: set of depots

$c_{0j}$: capacities available for source (depot) $j$

$c_{k0}$: transportation cost of one unit of product from source $k$ to depot $j$

$v(2^i(\{K\} + |J|))$: encoded solution

**Outputs:** $g_{kj}$: amount of shipment between nodes

$y_{kn}$: binary variable shows the opened facilities

While $\sum_j b_j \geq 0$

**Step 1:** $g_{kj} = 0 \ \forall j \in J, \ k \in K$

**Step 2:** select a node based on $l = \arg\max_{t \in |K| + |J| \ \forall j \in J, \ k \in K} v(1,t)$

**Step 3:** if $l \in K$ then a source is selected $k^* = l$

$$j = \arg\min_{c_{0j} \mid v(1,j) \neq 0, \ k \in K} n^* = v(2,k)$$

else $j = l$ a depot is selected

$$k = \arg\min_{c_{k0} \mid v(1,j) \neq 0, \ k \in K} n = v(2,k)$$

**Step 4:** $g_{kj} = \min(C_{0kj}, b_j)$

Update demands and capacities

$C_{0kj} = C_{0kj} - g_{kj}$, $b_j = b_j - g_{kj}$

**Step 5:** if $C_{0kj} = 0$ then $v(1,k) = 0$

if $b_j = 0$ then $v(1,j) = 0$

**end of loop**

**Step 6:** for 1 to $K$

if $\sum_j y_{kj} \geq 0$

$n = v(2,k), \ Y_{kn} = 1$

End

A chromosome for the IFRLN model is therefore presented as a $2(|I| + |J| + |K| + |L| + |M| + |L| + |I| + |L|)$ matrix (see Fig. 5). The chromosome consists of five segments, in which each segment is related to one echelon of the IFRLN.

To decode an IFRLN chromosome in forward network, the second segment should be decoded before the first segment. In other words, decoding the first segment is impossible before the second segment is decoded. Also in the reverse network,
decoding the fourth and fifth segments is impossible before the third segment is decoded. The decoding algorithm of an IFRLN chromosome is presented in Algorithm 2.

Algorithm 2: IFRLN decoding algorithm

Inputs: \( s, r_i, d_{ij}, cX_{ij}, cY_{ij}, cZ_{ij}, cP_{ji}, cQ_{ij}, cW_{ij}, cA_{ij}, cV_{ij}, cW_{ai}, cA_{wi} \)
Outputs: \( Z_i^n, V_i^n, Y_i^n, W_i^n, X_{ij}, U_{ik}, Q_{ij}, P_{ij}, T_{lm} \)
Step 1: calculate \( U_{ik}, Y_i^n \) \( \forall j, k \in K, n \in N \) using Algorithm 1
Step 2: calculate \( X_{ij}, W_i^n \) \( \forall i \in I, j \in J, n \in N \) using Algorithm 1
Step 3: calculate \( Q_{ij}, Z_i^n \) \( \forall i \in I, k \in K, n \in N \) using Algorithm 1
Step 4: calculate \( T_{lm}, V_i^m \) \( \forall l \in L, m \in M, n \in N \) using Algorithm 1
Step 5: calculate \( P_{ij} \) \( \forall l \in L, i \in I \) using Algorithm 1

5.2. Fitness and evaluation

In the proposed multi-objective memetic algorithm (MOMA), the fitness evaluation of chromosomes for survival is calculated by the random-weight approach of Murata et al. [49]. The advantage of this approach compared to the traditional weighted-sum approach is that it gives the algorithm a tendency to demonstrate a variable search direction, enabling it to sample the solution space uniformly over the entire frontier [40]. Given \( n \) objective functions, the fitness function is obtained by combining the objective functions as follows:

\[
Z = \sum_{i=1}^{n} w_i f_i(x) \tag{25}
\]

If \( r_i \) are nonnegative random numbers, the random weights \( w_i \) are calculated by the following equation:

\[
w_i = \frac{r_i}{\sum_{j=1}^{n} r_j}, \quad i = 1, 2, ..., n \tag{26}
\]

At the beginning of each evolution loop (generation), the values of each of objective function are normalized by Eq. (27). Since the second objective function is to be maximized the inverse value should be used in Eq. (27). Then, a new set of random weights is specified and the fitness value of each solution (chromosome) is calculated by Eq. (25):

\[
f_{i}^{\text{normal}} = \frac{f_i}{f_{i}^{\text{max}}}, \quad i = 1, 2, ..., n \tag{27}
\]

Based on the fitness values, a tentative set of Pareto solutions (selected among current population) is stored and passed on to new population for elite protection. The other individuals of the new population are generated through an evolutionary loop employing the crossover operator and local search operators.

5.3. Crossover

MOMA uses a segment-based crossover, randomly selecting the corresponding segments of parents with equal probability. As is shown in Fig. 6, the crossover operator uses a binary array of five positions. Because production and recovery process are performed in the same facilities the value of the fifth position of the binary array must be equal to the first one.

Each time a crossover operator is employed, two offspring solutions are generated but only one of them is selected for the local search phase based on their fitness values.

5.4. Selection

In the proposed algorithm the well-known roulette wheel selection method is used for selecting parents from the old population, making the selection probability of a chromosome proportional to its fitness value.

If \( \text{pop\_size} \) denotes the population size and \( \text{pareto\_size} \) denotes the number of tentative Pareto solutions to preserve, then the inner loop in each iteration must repeat \((\text{pop\_size} - \text{pareto\_size})\) times to generate sufficient individuals to complete the new population. The selection process is based on spinning the wheel the \(2^{(\text{pop\_size} - \text{pareto\_size})}\) times, each time selecting a single chromosome from the old population.
5.5. Dynamic local search

In the crossover phase, two children are created. The child with the best fitness value is selected for further improvement during the subsequent local search phase in which three local search methods are used: 3-opt, 2-opt and CA (capacity adjustment). The first two are employed on the priorities (first row of the chromosome) and they include two stages. In the first stage, each segment of the solution chromosome is randomly selected with equal chance by a binary array. In the second stage, the values of 3 and 2 cells, respectively, from each selected priority segment will be exchanged by the first (3-opt) or the second (2-opt) local search operator (Fig. 7).

The third local search method (CA) is applied to the capacity levels (second row of the chromosome). In the first stage, cells are selected by a random binary array and subsequently new capacity levels are assigned to them randomly, based on a uniform distribution over possible capacity levels.

During the search, the probability of applying each of the three local searches is adjusted to control the diversification and intensification of the search. As the algorithm approaches the last iterations, the share of 3-opt local search is decreased in favor of the 2-opt and the number of CA local searches being increased. This strategy is used to obtain a larger variety of solutions and a faster improvement path during the first iterations and to explore more carefully and in greater detail the obtained solutions at further iterations. The logic of this strategy is illustrated in Fig. 8.

After applying a local search operator, a new neighborhood solution is obtained. If the fitness value of this solution is better than the old solution, the old solution is replaced by the new one. To make the search more diversified and to escape from local optima during the local search, the probabilistic acceptance strategy is used if the new solution has a worse fitness value than the old solution. To use this strategy in a more guided way, the probability of accepting a worse solution is gently decreased during the search as specified below. The individual that comes out of the local search stage will be directly added to new population.

The detailed structure of MOMA can be outlined as follows:

**Algorithm 3: MOMA algorithm**

**Step 1: Initiation**
Randomly generating $4^*(pop\_size)$ chromosomes.
Select best $(pop\_size)/2$ chromosomes regarding $Z_1$ (first objective function) and best $(pop\_size)/2$ chromosomes regarding $Z_2$ (second objective function) as initial population.

**Step 2: Evaluation**
Calculate the value of two objective functions.
Update the set of tentative Pareto solutions.
Put the tentative Pareto solution into new population.

**Step 3: Fitness**
Specify the random weights by Eq. (26).
Calculate the fitness value by Eq. (25).

**Step 4: Selection.**
Calculate the selection probability based on fitness values for each individual in current population.
Select a pair of parents by roulette wheel selection method from current population.

**Step 5: Crossover.**
Apply crossover operator on selected parents to generate offspring.
Select the best offspring for next step regarding to fitness value and delete the other offspring.

**Step 6: Local search.**
With the probability of (0.75) do the following steps
For $i = 1$ to $10 + (k/8)$
if $i < 4$ apply the 3-opt local search
else apply the 2-opt local search
For $j = 1$ to $(4^*N/2) + k/8$
apply CA local search
if the fitness of obtained neighborhood solution is better than the current solution, the current solution is replaced by the new one.
otherwise, with the probability of $(0.05*pc)$ the current solution is replaced by the new solution.
End
Put the current solution (chromosome) into new population.

Step 7: Termination.
If the evolution loop repeats
$tun = 15 + (I + J + K + L + M + I + L)/60$ times stop the run, otherwise, return to Step 3.

Test problems were used to set the parameter values of the proposed algorithm. As is apparent from Algorithm 3, the number of iterations ($tun$) is set to $15 + (I + J + K + L + M + I + L)/60$), increasing the number of iterations with problem size as expressed by the number of potential production and recovery center locations ($I$), potential distribution center locations ($J$), the customer zones ($K$), potential collection/inspection center locations ($L$), potential disposal center locations ($M$) and the possible capacity levels for the facilities ($N$). The number of local searches is increased by a rate of $(k/8)$, $k$ denoting the current iteration number. For the CA local search, this increment also depends on the number of available capacity levels ($N$).

The probability of accepting a worse neighborhood solution in local search step is set equal to $(0.05*pc)$, in which $pc$ is calculated by the following equation:

$$pc = (tun - k)/(tun + 3)$$

Other parameters of the local search step (e.g. number of loops) are tuned to support the dynamic strategy described in Section 5.5. Finally, the size of the population ($pop_{size}$) is set to 30.

6. Computational results

To evaluate the performance of MOMA for integrated forward/reverse logistics network design problem, MOMA is compared to the multi-objective genetic algorithm (MOGA) of Altiparmark et al. [10]. The reason for selecting MOGA as a basis of comparison is its similarity to MOMA in using a priority-based encoding method and random-weight approach and the fact that it uses a different evolutionary search strategy. MOGA uses a $(\mu+\lambda)$ selection strategy and a 0.5 and 0.7 probability for its crossover and mutation operators, respectively.

To compare the performance of MOMA and MOGA, both algorithms are coded in MATLAB 7.0. The parameters of MOGA
solutions obtained by MOMA, then two sets (RP).

- The values of the parameters used in the test problems.
- Table 5

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_k$</td>
<td>Uniform (80, 250)</td>
</tr>
<tr>
<td>$r_k$</td>
<td>Uniform (0.75, 0.9)</td>
</tr>
<tr>
<td>$s$</td>
<td>$=0.2$</td>
</tr>
<tr>
<td>$f_k^j$</td>
<td>Uniform (450000, 800000)</td>
</tr>
<tr>
<td>$q_k^j$</td>
<td>Uniform (250000, 500000)</td>
</tr>
<tr>
<td>$h_k^j$</td>
<td>Uniform (250000, 500000)</td>
</tr>
<tr>
<td>$c_k^j$</td>
<td>Uniform (200000, 450000)</td>
</tr>
<tr>
<td>$f_k^m$</td>
<td>Uniform (95000, 150000)</td>
</tr>
<tr>
<td>$c_{jkl}$</td>
<td>Uniform (4, 12)</td>
</tr>
<tr>
<td>$c_{klj}$</td>
<td>Uniform (600, 15000)</td>
</tr>
<tr>
<td>$c_{klj}$</td>
<td>Uniform (400, 10000)</td>
</tr>
<tr>
<td>$c_{klj}$</td>
<td>Uniform (250, 700)</td>
</tr>
<tr>
<td>$c_{klj}$</td>
<td>Uniform (150, 400)</td>
</tr>
<tr>
<td>$c_{klj}$</td>
<td>Uniform (950000, 150000)</td>
</tr>
<tr>
<td>$p_{klj}$</td>
<td>Uniform (500, 800)</td>
</tr>
<tr>
<td>$R_{jkl}$</td>
<td>Uniform [5, 8]</td>
</tr>
<tr>
<td>$\rho$</td>
<td>=0.5</td>
</tr>
</tbody>
</table>

(number of iterations, population size and number of random initial solutions) are tuned for best possible performance on four instances of different problem size (Nos. 2, 4, 5, 7). Both algorithms are also tested on seven test problems of different sizes (see Table 4), in such a way that the sizes are selected in the range of test problems in the recent literature (e.g. [7,12]). Other parameters are generated randomly using uniform distributions specified in Table 5.

Three instances are generated randomly in each size of test problems and MOMA and MOGA are executed five times in each instance. All the tests are carried out on a Pentium dual-core 1.60 GHz computer with 1 GB RAM.

Because MOGA does not support multiple capacity for the facilities, MOGA and MOMA can only be compared in those cases for which the number of capacity levels is equal to 1 ($N=1$). Thus, in each test problem we compare the two algorithms in single-level capacity instances (see Table 6). Two performance measures are used to evaluate the algorithms: the average number of Pareto-optimal solutions obtained and the average ratio of Pareto-optimal solutions. If $P_{CS}$ denotes the set of Pareto-optimal solutions obtained by MOGA and $P_{MA}$ denotes the set of Pareto-optimal solutions obtained by MOMA, then $P$ denotes the union of these two sets ($P=P_{CS}\cup P_{MA}$). If $P_{ND}$ denotes the set of non-dominated solutions in $P$ then the ratio of Pareto-optimal solutions for MOMA ($RP_{MA}$) is calculated by the following equation:

$$RP_{MA} = \frac{|\{x \in P_{MA}|x \in P_{ND}\}|}{|P_{MA}|}$$

The same equation is used for calculating the ratio of Pareto-optimal solutions for MOGA ($RP_{CA}$). The summary of the test results are presented in Table 6.

Table 6

<table>
<thead>
<tr>
<th>Problem no.</th>
<th>CPU time (s)</th>
<th>Average no. of Pareto solutions</th>
<th>Average ratio of Pareto-optimal solutions (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MOGA MOMA</td>
<td>MOGA MOMA</td>
<td>MOGA     MOMA</td>
</tr>
<tr>
<td>1</td>
<td>24.30</td>
<td>28.24</td>
<td>6.20     5.60</td>
</tr>
<tr>
<td>2</td>
<td>57.31</td>
<td>60.58</td>
<td>9.93     8.33</td>
</tr>
<tr>
<td>3</td>
<td>63.57</td>
<td>74.45</td>
<td>12.00    10.40</td>
</tr>
<tr>
<td>4</td>
<td>106.60</td>
<td>125.20</td>
<td>9.46     8.93</td>
</tr>
<tr>
<td>5</td>
<td>197.49</td>
<td>247.71</td>
<td>11.93    13.06</td>
</tr>
<tr>
<td>6</td>
<td>327.96</td>
<td>422.84</td>
<td>8.93     8.46</td>
</tr>
<tr>
<td>7</td>
<td>497.73</td>
<td>650.30</td>
<td>9.53     9.46</td>
</tr>
</tbody>
</table>

Total average | 9.71 | 9.17 | 61.53 | 68.32 |

Fig. 9. An example of Pareto fronts of the solution obtained by MOMA and MOGA.
As mentioned in Section 1, most of the existing models from the literature only consider a single capacity level for each facility. However, in real-life problems, capacity levels are important decision variables due to their strong influence on logistics network efficiency and responsiveness. Because MOGA does not support multiple capacity level problems, we compare the MOMA solutions for medium-sized multiple capacity problems to the Pareto optimal solutions of the mixed integer programming model of Section 4 obtained by the LINGO 8.0 optimization software. To compare the non-dominated solutions of MOMA and LINGO, a single objective model is constructed in LINGO by keeping the first objective function \( W_1 \) as an objective function and transforming the second objective function \( W_2 \) into a side constraint. The optimal value of the second objective function \( W_2 \) of each solution obtained by MOMA is inserted as a right-hand side of this constraint. The resulting model is solved by LINGO for each solution obtained by MOMA and the optimal solution is compared with the corresponding value of \( W_2 \) obtained by MOMA. Therefore, the computation time reported for the LINGO software is the sum of computation times of these optimization problems for the set of non-dominated solutions. To evaluate the MOMA solutions, the following performance measure is used:

\[
\% \text{error} = \frac{\sum_{i=1}^{P_{MA}} W_{1,MOMA} - W_{1,OPT}}{W_{1,OPT}} \times 100
\]

in which \( P_{MA} \) denotes the set of Pareto-optimal solutions obtained by MOMA and \( W_{1,MOMA} \) and \( W_{1,OPT} \) are the values of the first objective function for i_th non-dominated solution obtained by MOMA and LINGO, respectively. The results of the comparison between MOMA and LINGO are reported in Table 7.

As the results in Table 7 show, the average deviation of MOMA solutions from the optimum varies from 0.91% to 5.63% on small to medium-sized problems. Computation times for MOMA are significantly lower than for LINGO as its computation time rapidly increases with the growing number of binary variables. Both findings illustrate the potential of MOMA for solving multi capacity instances. The computational results also reveal that MOMA’s computation time is 2.5 to 3.5 times higher for multi-level capacity instances than for single-capacity level instances. The increased computational requirements are probably due to the additional local search on capacity levels (CA) during the local search stage (see Algorithm 3).

Moreover, a sensitivity analysis on the number of capacity levels (N) confirms the strong influence of considering multi-level capacity for facilities on the overall performance of the logistics network. Fig. 10 shows the results of a single MOMA run on an instance of problem No. 3 with one and three capacity levels.

### Table 7

<table>
<thead>
<tr>
<th>Problem no.</th>
<th>No. of capacity levels (N)</th>
<th>Average no. of Pareto solutions</th>
<th>Average CPU time (s)</th>
<th>Average error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MOMA</td>
<td>LINGO</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>13.45</td>
<td>99.32</td>
<td>15.34</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>8.66</td>
<td>338.34</td>
<td>459.33</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>11.35</td>
<td>724.45</td>
<td>4898.24</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>10.65</td>
<td>1078.54</td>
<td>10246.02</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>9.78</td>
<td>1285.44</td>
<td>–</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>10.52</td>
<td>1626.20</td>
<td>–</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>12.25</td>
<td>2371.42</td>
<td>–</td>
</tr>
</tbody>
</table>

Fig. 10. Influence of capacity levels on network costs and responsiveness.

### 7. Conclusions

Because of the increasing importance of network costs and network responsiveness in supply chain management and reverse logistics activities, this paper presents a bi-objective mixed integer non-linear programming (MINLP) model for integrated forward/reverse logistics network design. Moreover, the model supports multiple capacity levels for each facility and also considers cost savings associated with combined distribution centers and collection/inspection centers. To reduce the complexity of the proposed MINLP model, the model is linearized by defining a new variable and adding a constraint to the model.

To solve the proposed model, a multi-objective memetic algorithm with dynamic local search mechanism (MOMA) is designed to find the non-dominated set of solutions. The performance of the proposed memetic algorithm is compared to the multi-objective genetic algorithm (MOGA) of Altiparmark et al. [10] on seven test problems with single capacity levels per facility and to exact solutions obtained by the LINGO optimization solver on medium-sized problems with multiple capacity levels. The numerical results show that the proposed MOMA algorithm outperformed the existing MOGA in terms of average ratio of Pareto-optimal solutions obtained. Also the comparison between MOMA and LINGO shows that the quality of solutions obtained by MOMA on multiple capacity test problems is reasonable. The main contribution of the new MOMA heuristic, however, consists of its ability to easily handle multiple capacity levels for the facilities within the forward/reverse logistics network and applying dynamic local search strategy.

Future research could be aimed at robust models to accommodate the changing parameters of the business environment during the life-time of the logistics network. In addition, addressing the demand uncertainty and the supply of returned products in a multi-product integrated logistics network is a promising research avenue with significant practical relevance. Although our memetic dynamic search strategy proved to be competitive for the IFRNL network problem under consideration, other multi-objective metaheuristics algorithms such as multi-objective tabu search or multi-objective scatter search could offer promising avenues for developing richer integrated logistics networks as well.

### References


