RAIRO-Oper. Res. 49 (2015) 243–264
DOI: 10.1051/ro/2014032

**RAIRO Operations Research**
www.rairo-ro.org

# A STRAIGHT PRIORITY-BASED GENETIC ALGORITHM FOR A LOGISTICS NETWORK

Mehrdad Mehrbod[1], Zhaojie Xue[2], Lixin Miao[1] and Wei-Hua Lin[3]

**Abstract.** Closed-loop logistics (forward and reverse logistics) has received increased attention of late due to customer expectations, greater environmental concerns, and economic aspects. Unlike previous works, which consider single products or single periods in multi-objective function problems, this paper considers a multi-product multi-period closed-loop logistics network with regard to facility expansion as a facility location-allocation problem, which is closer to real-world scenarios. A multi-objective mixed integer nonlinear programming formulation is developed to minimize the total cost, the product delivery time, and the used product collection time. The model is linearized by defining new variables and adding new constraints to the model. Then, to solve the model, a priority-based genetic algorithm is proposed that uses straight encoding and decoding methods. To assess the performance of the above algorithm, its final solutions and CPU times are compared to those generated by an initial priority-based genetic algorithm from the recent literature and the lower bound obtained by CPLEX. The numerical results show that the straight priority-based genetic algorithm outperforms the initial priority-based genetic algorithm at least in terms of obtaining a reasonable quality of final solutions for closed-loop logistics problems.

**Keywords.** Closed-loop logistics, multi-objective decision making, genetic algorithm, forward and reverse logistics.

**Mathematics Subject Classification.** 90B06.

Received June 27, 2012. Accepted May 12, 2014.

[1] Research Center for Modern Logistics, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, P.R. China. mehrbod.mehrdad@gmail.com, lxmiao@tsinghua.edu.cn

[2] Department of Transportation Engineering, College of Civil Engineering, Shenzhen University, 518060 Shenzhen, P.R. China. xuezhaojie@126.com

[3] Department of Systems and Industrial Engineering, The University of Arizona, AZ 85721 Tucson, USA. whlin@email.arizona.edu

Article published by EDP Sciences 

## 1. Introduction

In recent times, due to increasing environmental and social concerns, along with the economic benefits to be gained, an increasing number of companies have begun to focus on reverse logistics in addition to forward logistics. Forward logistics encompasses material supply, production, distribution and consumption. In the case of reverse logistics, the flow of used products includes collection, inspection/separation, recovery, disposal, and redistribution. Such a network combination is considered to be a closed-loop logistics (CLL) network. At a planning level, different decision-making problems arise in CLL networks. One of these is the facility location-allocation problem. This type of problem includes designing the logistics configuration, selecting the facility location, assigning facilities and determining the flow quantities between facilities and consumers.

First presented by Marín and Pelegrín [12], research on CLL began to appear in scholarly journals from the late 1990s. Among CLL research, the minimization of total costs is the most commonly used single objective [8,16]. In contrast, profit maximization has received much less attention from researchers [4,11]. Real-world network design problems are often characterized by multiple and conflicting objectives. Network responsiveness is an important issue in reverse logistics. It is undesirable for customers to retain used products for an extended period of time because of the related holding costs. Therefore, companies should consider customer satisfaction in addition to cost minimization. Of around 50 papers in the area of CLL networks we have reviewed, a majority centered on a formulation and solution method for a single-objective problem. Upon closer examination, we found only three papers that considered a multi-objective problem. Lee *et al.* [10] developed a multi-objective model for single-product single-period CLL. Two objective functions were considered: (1) to maximize the quantity of the returned products and (2) to minimize the total cost. A fuzzy goal programming approach was applied to determine the compromise solution for the multi-objective model. A genetic algorithm (GA) with two sub-algorithms was then developed to solve the problem. Pishvaee *et al.* [15] used mixed integer linear programming to develop a multi-objective single-product single-period model that included minimizing the total costs and maximizing the responsiveness of a logistics network. To solve the proposed model, a memetic algorithm with a dynamic local search mechanism was designed to find the non-dominated set of solutions. Pishvaee and Torabi [17] proposed a possibilistic mixed integer programming model to address a single-product multi-period (only for the demand parameter) CLL under uncertainty. The primary objectives included a total cost minimization, and the minimization of the total tardiness of the delivered products. To solve the proposed model, an interactive fuzzy solution approach was developed by combining a number of efficient solution approaches from the recent literature.

Based on the aforementioned considerations, this paper proposes a multi-product multi-period model for a multi-objective CLL network, along with the possibility of facility expansion. In particular, instead of using the initial priority-based encoding method commonly used in GAs, we develop a new priority-based encoding GA with its corresponding decoding method to enhance the performance of the solution approach.

The remainder of this paper is organized as follows. First, we develop a generalized mixed integer non-linear programming formulation to design the multi-period, multi-product, and multi-objective CLL network. The model is then linearized and a straight priority-based GA (SPGA) is designed to solve the model. Then, a computational experiment is conducted to compare the results obtained using the initial priority-based genetic algorithm (IPGA), the linear relaxation, and the SPGA. Finally, conclusions and further research are discussed.

## 2. Research problem

### 2.1. Problem definition

The integrated forward/reverse logistics network discussed in this research problem is a multi-stage logistics network that includes plants, retailers, distribution, collection, recovery, and recycling centers.

Figure 1 shows that, in the forward flow, a certain number of new products (QI, QJ) are shipped from plants to retailers through distribution centers to meet the demand of each retailer. Plants and retailers are assumed to be predetermined and fixed. In the reverse flow, a certain number of returned products (QK) are collected in collection centers and, after testing, the recoverable products (QL) are shipped to recovery centers. Scrapped products (QS) are shipped to recycling centers. The recovery process is performed in recovery centers, and recovered products (QR) are inserted in the forward network and are considered to be identical to new products. By means of this strategy, the excessive transportation of returned products (especially scrapped products) is prevented, and the returned products can be shipped directly to the appropriate centers. Thus, the network is a CLL network. In such an integrated logistics network, hybrid centers offer potential cost savings compared to separate distribution and collection centers. The network, therefore, considers a hybrid distribution-collection center in which both distribution and collection centers are established at the same location. This model, unlike the existing location models, considers facility expansion over time in order to manage the network based on the trade-offs for various situations. By means of this change, we can increase the utilization rate of facilities and decrease the total cost, in addition to making our problem closer to real life.

We consider a decision horizon that includes multiple-periods and multiple-products in the proposed model. The flow quantities during each period between facilities that belong to different echelons are determined according to demand, returns and other periodic-based parameters. As such, this paper assumes that the demand for the products, the number of returned products, and the number of unrecoverable used products are known over the planning time horizon.

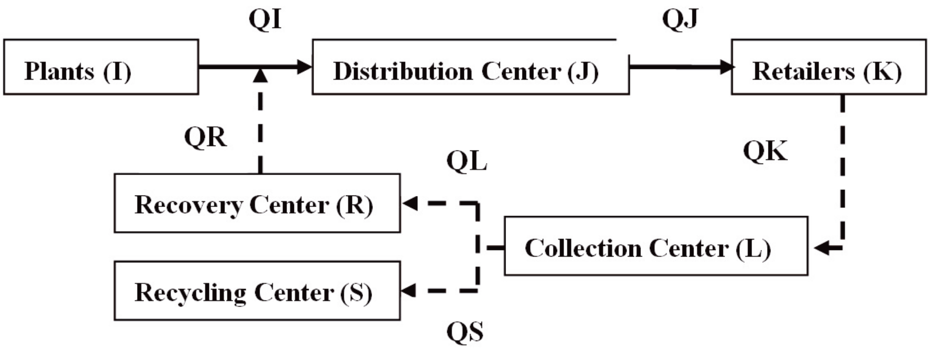



FIGURE 1. Model flow of a closed-loop logistics network.

The other main assumptions used in this problem formulation are as follows:

a) All products returned by the retailers must be collected, and all of the demand of the retailers must be satisfied.
b) Products are shipped through a pull mechanism and returned products are shipped through a push mechanism in the forward and reverse sides of the network, respectively.
c) A recycling center is a storage place for scrapped products. Based on our research teams experience, any processing costs in such centers are very small compared with other costs. Therefore, we do not consider any processing costs (such as the storage cost) for this type of facility.
d) There are no missing products in the forward logistics process.
e) We assume that the rates of returned and recoverable products are not constant for various reasons such as product quality, or the market situation in each period.
f) Fixed savings cost happens when a distribution center and a collection center are opened at the same location in the same period (hybrid facility).

To design the CLL network, two objective functions are considered: (1) to minimize the total cost and (2) to minimize the total delivery and collection time. The first objective is related to supply chain network efficiency and the second to network responsiveness. The second objective enables the supply chain to satisfy the customers expected delivery and collection times. These two objective functions conflict with each other. Optimizing the network involves trade-offs between these two objectives.

### 2.2. MODEL FORMULATION

The following notation is used in the formulation of the CLL problem.

$I, i\backslash J, j\backslash K, k\backslash L, l\backslash R, r\backslash S, s\backslash P, p\backslash T, t$: set and index of plants \distribution centers \retailers \collection centers \recovery centers \recycling centers \products \time periods;

$AC_p$: per unit storage space requirement for product $p$;
$AR^t_{\mathrm{kp}}$: percentage of product $p$ returned by retailer $k$ in period $t$;
$AS^t_p$: unrecoverable percentage of product $p$ in period $t$;
$CD\backslash CC$: cost of a delay in product delivery \collection per product per unit of time;
$CI_{ip}$: maximum production capacity of plant $i$ for product $p$;
$CJ_j\backslash CL_l\backslash CR_r\backslash CS_s$: maximum capacity of distribution center $j$ \collection center $l$ \recovery center $r$ \recycling center $s$;
$DP^t_{\mathrm{kp}}$: demand for product $p$ from retailer $k$ in period $t$;
$EC^t_{\mathrm{kp}}$: expected collection time of product $p$ from retailer $k$ in period $t$;
$ED^t_{\mathrm{kp}}$: expected delivery time of product $p$ to retailer $k$ in period $t$;
$EJ^t_j\backslash EL^t_l\backslash ER^t_r$: operating cost for standard expansion on distribution center $j$ \collection center $l$ \recovery center $r$ in period $t$;
$FH^t_h$: fixed savings cost associated with opening a distribution center and a collection center at location $h$ in period $t$, $h \in H, H \subset J, H \subset L$;
$FJ^t_j\backslash FL^t_l\backslash FR^t_r\backslash FS^t_s$: fixed cost of opening distribution center $j$ \collection center $l$ \recovery center $r$ \recycling center $s$ in period $t$;
$GJ_j\backslash GL_l\backslash GR_r$: standard expansion size of distribution center $j$ \collection center $l$ \recovery center $r$;
$MJ^t_j\backslash ML^t_l\backslash MR^t_r$: maximum number of times in period $t$ for which $GJ_j\backslash GL_l\backslash GR_r$ can occur;
$PI_{ip}$: manufacturing cost per unit of product $p$ at plant $i$;
$PJ_{jp}\backslash PL_{lp}$: processing cost per unit of product $p$ at distribution center $j$ \collection center $l$;
$PR_{rp}$: remanufacturing cost per unit of product $p$ at recovery center $r$;
$TC_{\mathrm{klp}}$: collection time of product $p$ from retailer $k$ by collection center $l$;
$TD_{\mathrm{jkp}}$: delivery time of product $p$ from distribution center $j$ to retailer $k$;
$D^t =$ $\{j\,|TD_{\mathrm{jkp}} \geq ED_{\mathrm{kp}}\}$ and $C^t = \{l\,|TC_{\mathrm{klp}} \geq EC_{\mathrm{kp}}\}$ at period $t$;
$TI_{\mathrm{ijp}}\backslash TJ_{\mathrm{jkp}}\backslash TK_{\mathrm{klp}}\backslash TL_{\mathrm{lrp}}\backslash TS_{\mathrm{lsp}}\backslash TR_{\mathrm{rjp}}$: transportation cost per unit of product $p$ from $i$ to $j$ \$j$ to $k$ \$k$ to $l$ \$l$ to $r$ \$l$ to $s$ \$r$ to $j$.

**Decision variables**:

$QI^t_{\mathrm{ijp}}$: quantity of product $p$ shipped from plant $i$ to distribution center $j$ in period $t$;
$QJ^t_{\mathrm{jkp}}$: quantity of product $p$ shipped from distribution center $j$ to retailer $k$ in period $t$;
$QK^t_{\mathrm{klp}}$: quantity of product $p$ shipped from retailer $k$ to collection center $l$ in period $t$;
$QL^t_{\mathrm{lrp}}$: quantity of product $p$ shipped from collection center $l$ to recovery center $r$ in period $t$;

$QS^t_{\mathrm{lsp}}$: quantity of product $p$ shipped from collection center $l$ to recycling center $s$ in period $t$;

$QR^t_{\mathrm{rjp}}$: quantity of product $p$ shipped from recovery center $r$ to distribution center $j$ in period $t$;

$XJ^t_j = 1$ if a distribution center is opened at location $j$ in period $t$, zero otherwise;

$XL^t_l = 1$ if a collection center is opened at location $l$ in period $t$, zero otherwise;

$XR^t_r = 1$ if a recovery center is opened at location $r$ in period $t$, zero otherwise;

$XS^t_s = 1$ if a recycling center is opened at location $s$ in period $t$, zero otherwise;

$ZJ^t_j \backslash ZL^t_l \backslash ZR^t_r$: number of standardized expansions to distribution center $j$ \collection center $l$ \recovery center $r$ in period $t$;

The CLL problem can be formulated as follows:

$$\text{Min } Z1 = \text{Opening cost} + \text{Expansion cost} + \text{Transportation cost} + \text{Processing cost}$$

$$\begin{aligned}
&= \sum\nolimits_j FJ^1_j XJ^1_j + \sum\nolimits_{t\ge 2}\sum\nolimits_j FJ^t_j XJ^t_j \left(1 - XJ^{t-1}_j\right) \\
&\quad + \sum\nolimits_l FL^1_l XL^1_l + \sum\nolimits_{t\ge 2}\sum\nolimits_l FL^t_l XL^t_l \left(1 - XL^{t-1}_l\right) \\
&\quad + \sum\nolimits_r FR^1_r XR^1_r + \sum\nolimits_{t\ge 2}\sum\nolimits_r FR^t_r XR^t_r \left(1 - XR^{t-1}_r\right) \\
&\quad + \sum\nolimits_s FS^1_s XS^1_s + \sum\nolimits_{t\ge 2}\sum\nolimits_s FS^t_s XS^t_s \left(1 - XS^{t-1}_s\right) \\
&\quad - \sum\nolimits_{h=j=l} FH^1_h XJ^1_h XL^1_h \\
&\quad - \sum\nolimits_{t\ge 2}\sum\nolimits_{h=j=l} FH^t_h XJ^t_h XL^t_h \left(1 - XJ^{t-1}_h XL^{t-1}_h\right) \\
&\quad + \sum\nolimits_t\sum\nolimits_j EJ^t_j ZJ^t_j + \sum\nolimits_t\sum\nolimits_l EL^t_l ZL^t_l + \sum\nolimits_t\sum\nolimits_r ER^t_r ZR^t_r \\
&\quad + \sum\nolimits_t\sum\nolimits_p\sum\nolimits_j\sum\nolimits_i TI_{\mathrm{ijp}} QI^t_{\mathrm{ijp}} + \sum\nolimits_t\sum\nolimits_p\sum\nolimits_k\sum\nolimits_j TJ_{\mathrm{jkp}} QJ^t_{\mathrm{jkp}} \\
&\quad + \sum\nolimits_t\sum\nolimits_p\sum\nolimits_l\sum\nolimits_k TK_{\mathrm{klp}} QK^t_{\mathrm{klp}} + \sum\nolimits_t\sum\nolimits_p\sum\nolimits_r\sum\nolimits_l TL_{\mathrm{lrp}} QL^t_{\mathrm{lrp}} \\
&\quad + \sum\nolimits_t\sum\nolimits_p\sum\nolimits_s\sum\nolimits_l TS_{\mathrm{lsp}} QS^t_{\mathrm{lsp}} + \sum\nolimits_t\sum\nolimits_p\sum\nolimits_j\sum\nolimits_r TR_{\mathrm{rjp}} QR^t_{\mathrm{rjp}} \\
&\quad + \sum\nolimits_t\sum\nolimits_p\sum\nolimits_j\sum\nolimits_i PI_{ip} QI^t_{\mathrm{ijp}} + \sum\nolimits_t\sum\nolimits_p\sum\nolimits_k\sum\nolimits_j PJ_{jp} QJ^t_{\mathrm{jkp}} \\
&\quad + \sum\nolimits_t\sum\nolimits_p\sum\nolimits_r\sum\nolimits_l PL_{lp} Ql^t_{\mathrm{lrp}} + \sum\nolimits_t\sum\nolimits_p\sum\nolimits_s\sum\nolimits_l PL_{lp} QS^t_{\mathrm{lsp}} \\
&\quad + \sum\nolimits_t\sum\nolimits_p\sum\nolimits_j\sum\nolimits_r PR_{rp} QR^t_{\mathrm{rjp}} \qquad (2.1)
\end{aligned}$$

$$\text{Min } Z2 = \text{Delivery time} + \text{Collection time}$$

$$\begin{aligned}
&= CD\sum\nolimits_t\sum\nolimits_p\sum\nolimits_k\sum\nolimits_{j\in D^t} (TD_{\mathrm{jkp}} - ED^t_{\mathrm{kp}}) QJ^t_{\mathrm{jkp}} \\
&\quad + CC\sum\nolimits_t\sum\nolimits_p\sum\nolimits_k\sum\nolimits_{l\in C^t} (TC_{\mathrm{klp}} - EC^t_{\mathrm{kp}}) QK^t_{\mathrm{klp}} \qquad (2.2)
\end{aligned}$$

Subject to:

$$\sum\nolimits_j QJ^t_{\mathrm{jkp}} \geq DP^t_{\mathrm{kp}} \quad \forall t,p,k \tag{2.3}$$

$$\sum\nolimits_l QK^t_{\mathrm{klp}} \geq AR^t_{\mathrm{kp}} DP^t_{\mathrm{kp}} \quad \forall t,p,k \tag{2.4}$$

$$\sum\nolimits_i QI^t_{\mathrm{ijp}} + \sum\nolimits_r QR^t_{\mathrm{rjp}} = \sum\nolimits_k QJ^t_{\mathrm{jkp}} \quad \forall t,p,j \tag{2.5}$$

$$(1 - AS^t_p) \sum\nolimits_k QK^t_{\mathrm{klp}} = \sum\nolimits_r QL^t_{\mathrm{lrp}} \quad \forall t,p,l \tag{2.6}$$

$$AS^t_p \sum\nolimits_k QK^t_{\mathrm{klp}} = \sum\nolimits_s QS^t_{\mathrm{lsp}} \quad \forall t,p,l \tag{2.7}$$

$$\sum\nolimits_l QL^t_{\mathrm{lrp}} = \sum\nolimits_j QR^t_{\mathrm{rjp}} \quad \forall t,p,r \tag{2.8}$$

$$\sum\nolimits_j QI^t_{\mathrm{ijp}} \leq CI_{ip} \quad \forall t,p,i \tag{2.9}$$

$$\sum\nolimits_p AC_p \Big( \sum\nolimits_i QI^t_{\mathrm{ijp}} + \sum\nolimits_r QR^t_{\mathrm{rjp}} \Big) \leq CJ_j XJ^t_j + \sum_{\theta=1}^{t} (GJ_j ZJ^\theta_j) \quad \forall t,j \tag{2.10}$$

$$\sum\nolimits_p AC_p \sum\nolimits_k QK^t_{\mathrm{klp}} \leq CL_l XL^t_l + \sum_{\theta=1}^{t} (GL_l ZL^\theta_l) \quad \forall t,l \tag{2.11}$$

$$\sum\nolimits_p AC_p \sum\nolimits_l QL^t_{\mathrm{lrp}} \leq CR_r XR^t_r + \sum_{\theta=1}^{t} GR_r ZR^\theta_r \quad \forall t,r \tag{2.12}$$

$$\sum\nolimits_p AC_p \sum\nolimits_l QS^t_{\mathrm{lsp}} \leq CS_s XS^t_s \quad \forall t,s \tag{2.13}$$

$$XJ^{t+1}_j \geq XJ^t_j \quad \forall t < |T|, j \tag{2.14}$$

$$XL^{t+1}_l \geq XL^t_l \quad \forall t < |T|, l \tag{2.15}$$

$$XR^{t+1}_r \geq XR^t_r \quad \forall t < |T|, r \tag{2.16}$$

$$XS^{t+1}_s \geq XS^t_s \quad \forall t < |T|, s \tag{2.17}$$

$$ZJ^t_j \leq M * XJ^t_j \quad \forall t,j \tag{2.18}$$

$$ZL^t_l \leq M * XL^t_l \quad \forall t,l \tag{2.19}$$

$$ZR^t_r \leq M * XR^t_r \quad \forall t,r \tag{2.20}$$

$$ZJ^t_j \leq MJ^t_j \quad \forall t,j \tag{2.21}$$

$$ZL^t_l \leq ML^t_l \quad \forall t,l \tag{2.22}$$

$$ZR^t_r \leq MR^t_r \quad \forall t,r \tag{2.23}$$

$$XJ_j^t, XL_l^t, XR_r^t, XS_s^t \in \{0,1\} \quad \forall t, j, r, l, s \tag{2.24}$$

$$QI_{\text{ijp}}^t, QJ_{\text{jkp}}^t, QK_{\text{klp}}^t, QL_{\text{lrp}}^t, QS_{\text{lsp}}^t, QR_{\text{rjp}}^t \geq 0 \quad \forall t, p, i, j, k, l, r, s \tag{2.25}$$

$$ZJ_j^t, ZL_l^t, ZR_r^t \text{ integer} \quad \forall t, j, l, r. \tag{2.26}$$

Constraint (2.3) ensures that the demands of all customers are satisfied. Constraint (2.4) ensures that all of the returned products from all of the customers are collected. Constraints (2.5)−(2.8) ensure the balance of quantity flow at the distribution, collection, recovery and recycling centers. Constraints (2.9)−(2.13) are capacity constraints on facilities, including expansion size across the time period, and also prohibit units of products, returned products, recoverable and recyclable products from being transferred to facilities that have not yet been opened. Constraints (2.14)−(2.17) guarantee that the open facilities cannot be closed during the following periods. Constraints (2.18)−(2.20) ensure that the expansion of a facility is only possible if that facility has already been opened. Constraints (2.21)−(2.23) impose a maximum number of standardized expansions for each type of facility in each period of time. Finally, Constraints (2.24)−(2.26) enforce binarity, nonnegativity or integrality on the decision variables.

In the objective function, there are six nonlinear terms to be considered, dealing with the fixed costs of opening distribution, collection, recovery, and recycling centers and the fixed savings cost of a hybrid facility (two nonlinear terms). Each of them involves the multiplication of two binary variables $(XJ_j^t, XJ_j^{t-1})$, $(XL_l^t, XL_l^{t-1})$, $(XR_r^t, XR_r^{t-1})$, $(XS_s^t, XS_s^{t-1})$, and $(XJ_h^t, XL_h^t)$ respectively. Therefore, the above model is linearized by defining new variables as follows:

First, using $X'J_j^t = XJ_j^t\left(1 - XJ_j^{t-1}\right)$, the following constraints are added into the model:

$$XJ_j^t + XJ_j^{t-1} + X'J_j^t \leq 2 \qquad \forall t \geq 2, j \tag{2.27}$$

$$XJ_j^t + XJ_j^{t-1} - X'J_j^t \geq 0 \qquad \forall t \geq 2, j \tag{2.28}$$

$$2XJ_j^t - XJ_j^{t-1} - X'J_j^t \leq 1 \qquad \forall t \geq 2, j \tag{2.29}$$

$$-2XJ_j^t + XJ_j^{t-1} + X'J_j^t \leq 1 \qquad \forall t \geq 2, j. \tag{2.30}$$

Constraint (2.27) ensures that, if $XJ_j^t = 1$ and $XJ_j^{t-1} = 1$, then $X'J_j^t$ is zero; constraint (2.28) ensures that, if $XJ_j^t = 0$ and $XJ_j^{t-1} = 0$, then $X'J_j^t$ is zero; constraint (2.29) guarantees that, if $XJ_j^t = 1$ and $XJ_j^{t-1} = 0$, then $X'J_j^t$ is equal to one, and constraint (2.30) ensures that, if $XJ_j^t = 0$ and $XJ_j^{t-1} = 1$, then $X'J_j^t$ is zero.

Second, using $X'L_l^t = XL_l^t\left(1 - XL_l^{t-1}\right)$, $X'R_r^t = XR_r^t\left(1 - XR_r^{t-1}\right)$, and $X'S_s^t = XS_s^t\left(1 - XS_s^{t-1}\right)$, based on the same logic as was applied for the fixed cost of opening a distribution center, the following constraints are added to the model:

$$XL_l^t + XL_l^{t-1} + X'L_l^t \le 2 \qquad \forall t \ge 2, l \tag{2.31}$$

$$XL_l^t + XL_l^{t-1} - X'L_l^t \ge 0 \qquad \forall t \ge 2, l \tag{2.32}$$

$$2XL_l^t - XL_l^{t-1} - X'L_l^t \le 1 \qquad \forall t \ge 2, l \tag{2.33}$$

$$-2XL_l^t + XL_l^{t-1} + X'L_l^t \le 1 \qquad \forall t \ge 2, l \tag{2.34}$$

$$XR_r^t + XR_r^{t-1} + X'R_r^t \le 2 \qquad \forall t \ge 2, r \tag{2.35}$$

$$XR_r^t + XR_r^{t-1} - X'R_r^t \ge 0 \qquad \forall t \ge 2, r \tag{2.36}$$

$$2XR_r^t - XR_r^{t-1} - X'R_r^t \le 1 \qquad \forall t \ge 2, r \tag{2.37}$$

$$-2XR_r^t + XR_r^{t-1} + X'R_r^t \le 1 \qquad \forall t \ge 2, r \tag{2.38}$$

$$XS_s^t + XS_s^{t-1} + X'S_s^t \le 2 \qquad \forall t \ge 2, s \tag{2.39}$$

$$XS_s^t + XS_s^{t-1} - X'S_s^t \ge 0 \qquad \forall t \ge 2, s \tag{2.40}$$

$$2XS_s^t - XS_s^{t-1} - X'S_s^t \le 1 \qquad \forall t \ge 2, s \tag{2.41}$$

$$-2XS_s^t + XS_s^{t-1} + X'S_s^t \le 1 \qquad \forall t \ge 2, s. \tag{2.42}$$

Finally, the nonlinear terms, for the fixed savings cost of a hybrid facility, are linearized using the following two steps:

First, a new variable $XH_{h=j=l}^t = XJ_j^t XL_l^t$ is defined as follows:

$XH_{h=j=l}^t = 1$: if a distribution center and a collection center are opened at location $h$ in period $t$, zero otherwise.

Substituting the new variable, the transformed term is:

$$\sum\nolimits_{h=j=l} FH_h^1 XH_h^1 - \sum\nolimits_{t\ge 2} \sum\nolimits_{h=j=l} FH_h^t XH_h^t \left(1 - XH_h^{t-1}\right).$$

However, as the objective function minimizes cost, it has a tendency to set the value of $XH_h^t$ to 1. We should, thus, set the value of $XH_h^t$ to zero when at least one of $XJ_j^t$ and $XL_l^t$ is equal to zero. This restriction can be achieved by adding the following constraints to the model.

$$2XH_{h=j=l}^t \le XJ_j^t + XL_l^t \quad \forall t, j, l. \tag{2.43}$$

$$-XH_{h=j=l}^t + XJ_j^t + XL_l^t \le 1 \quad \forall t, j, l. \tag{2.44}$$

Second, using $X'H_h^t = XH_h^t\left(1 - XH_h^{t-1}\right)$, and the same logic as was applied for the fixed costs of opening other centers, the following constraints are added to the model:

$$XH_h^t + XH_h^{t-1} + X'H_h^t \leq 2 \qquad \forall t \geq 2, h \tag{2.45}$$

$$XH_h^t + XH_h^{t-1} - X'H_h^t \geq 0 \qquad \forall t \geq 2, h \tag{2.46}$$

$$2XH_h^t - XH_h^{t-1} - X'H_h^t \leq 1 \qquad \forall t \geq 2, h \tag{2.47}$$

$$-2XH_h^t + XH_h^{t-1} + X'H_h^t \leq 1 \qquad \forall t \geq 2, h. \tag{2.48}$$

## 3. Solution approach

In this paper, the proposed CLL network includes three problems – the capacitated facility location problem, the flow optimization problem, and the reverse logistics problem. As proven by Krarup and Pruzan [9], the capacitated facility location problem is an NP-complete problem; therefore, the extended problem is an NP-hard problem. Solving this problem on a large scale with an exact algorithm is computationally intractable. As one of the evolutionary algorithms, the GA is a powerful and broadly applicable stochastic search and optimization technique based on principles derived from natural evolution and genetics [6].

The workability of genetic algorithms is based on Darwinians theory of survival of the fittest. Genetic algorithms may require the definition of chromosomes, genes, sets of population, fitness functions, as well as mutation and selection mechanisms. Genetic algorithms begin with a set of solutions represented by chromosomes, called population. Solutions from one population are taken and used to form a new population, which is motivated by the possibility that the new population will be better than the old one. To achieve that, solutions are selected according to their fitness to form new solutions, that is, from the offspring chromosomes. The above process is repeated until some condition is satisfied.

### 3.1. Straight-based encoding scheme

Usually, different problems have different genetic representations. During the last 20 years, various encoding methods have been developed to provide effective implementation of GAs. Tree-based representation is one way of representing network problems. In 1991, Michalewicz *et al.* [13] used matrix-based representation for solving linear and nonlinear transportation/distribution problems. If we consider $|I|$ and $|J|$ as the number of sources and depots, respectively, the dimension of the matrix will be $|I| \times |J|$. Gen and Li [7] employed the Prüfer number for encoding a spanning tree. This method belongs to vertex-based encoding and needs $|I| + |J| - 2$ genes to represent a candidate solution for a transportation tree. In 2006, Gen *et al.* [5] developed a priority-based GA for a single-product two-stage transportation problem that we call the initial priority-based GA or IPGA. This method, which also belongs to vertex-based encoding, was applied to a single-product, single-source, three-stage supply chain network [2], and to a multi-product, three-stage supply chain network problem as a steady-state GA [1]. A gene in a chromosome is characterized by the position of the gene within the structure of the chromosome and the value the gene takes. In the IPGA, the position of a gene is used not only to represent a node (source/depot in a transportation network) but also the product type, and the value is used to represent the priority of the corresponding node for constructing a tree among all the candidate nodes.

When the IPGA is applied to a multi-product transportation problem in which $P$ is the set of products, a stage of chromosome is generated that consists of $|P|$ parts, and the length of each stage is $|P|(|I| + |J|)$, but in the straight priority-based GA (SPGA), the length of each stage is equal to $|I| + |P||J|$. This means that, in the IPGA and the SPGA, the maximum gene values of the chromosome are $|P|(|I| + |J|)$ and $|I| + |P||J|$, respectively. In the SPGA, when the highest priority in a stage belongs to a source, the minimum transportation cost should be considered for all products in order to determine a depot. However, under the IPGA, the highest priority determines not only a source but also the product type, simultaneously. This means that, when the highest priority in a stage belongs to a source, the IPGA considers minimum transportation cost in the case of the determined product in order to determine the depot. In other words, the IPGA decreases the number of choices required to find the minimum cost for all products by predetermining a product.

Let $T$ be the set of periods. To consider a multi-period method, $|T| - 1$ rows should be added to the above chromosomes. Thus, in the IPGA and the SPGA, solutions are encoded as $|T| \times |P|(|I| + |J|)$ and $|T|(|I| + |P||J|)$ matrices, respectively. Figure 2 represents a comparison between these two chromosomes for two sources, three depots, three products, and two periods.

According to the SPGA, our problem is presented as a $|T| \times (|I| + |J| + |K| + 2|L| + |R| + |P|(2|J| + |K| + |L| + |R| + |S|))$ matrix. The chromosome consists of six stages, in which each stage is related to one echelon of our problem (Fig. 3). In our model, the sequence of stages in the decoding method is $2 \to 3 \to 4, 5 \to 6 \to 1$. The straight priority-based decoding method for the multi-period, multi-product CLL problem is given below.

**Input:**

$I \backslash J \backslash P \backslash T$: set of sources \depots \products \periods respectively

$d^t_{jp}$: demand for product $p$ from depot $j$ in period $t$, $\forall j \in J, p \in P, t \in T$

$s^t_i$: capacity of source $i$ in period $t$, $\forall i \in I, t \in T$

$c^t_{\mathrm{ijp}}$: unit transportation cost for taking product $p$ from source $i$ to depot $j$ in period $t$, $\forall i \in I, j \in J, p \in P, t \in T$

$v^t_r\,(|T|\,(|I| + |P|\,|J|))$: The chromosome that needs to be decoded $\forall i \in I, j \in J, p \in P, t \in T, r \in |T|\,(|I| + (|P|\,|J|))$

| | First Stage | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | First Product | | | | | Second Product | | | | | Third Product | | | | |
| | Source | | Depot | | | Source | | Depot | | | Source | | Depot | | |
| | 1 | 2 | 1 | 2 | 3 | 1 | 2 | 1 | 2 | 3 | 1 | 2 | 1 | 2 | 3 |
| Period1 | | | | | | | | | | | | | | | |
| Period2 | | | | | | | | | | | | | | | |

(a)

| | First Stage | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Source | | First Product Depot | | | Second Product Depot | | | Third Product Depot | | |
| | 1 | 2 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| Period1 | | | | | | | | | | | |
| Period2 | | | | | | | | | | | |

(b)

FIGURE 2. A comparison between the (a) initial and (b) straight chromosomes.

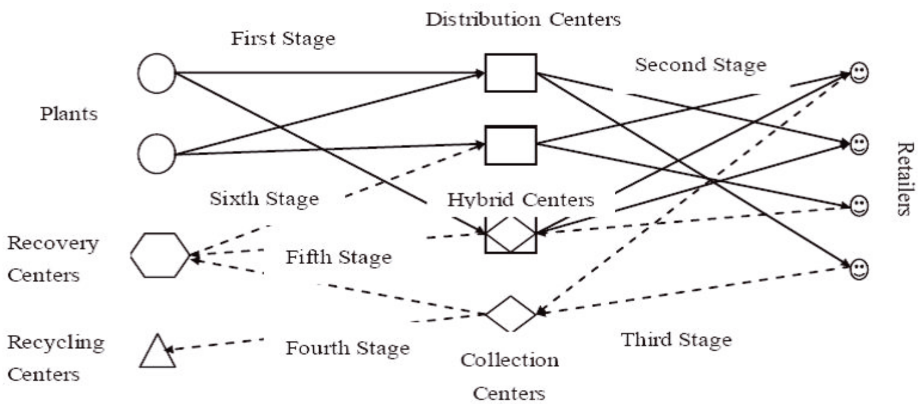



FIGURE 3. An illustration of our model.

$a_p$: per unit storage space requirement for product $p$, $\forall p \in P$ and $\sum_p a_p \sum_j d^t_{jp} \leq \sum_i s^t_i$, $\forall t \in T$

**Local variable:**

$v^t_i, v^t_{jp}$: gene, $\forall i \in I, j \in J, p \in P, t \in T$

**Output:**

$q^t_{\mathrm{ijp}}$: quantity of product $p$ shipped from source $i$ to depot $j$ in period $t$, $\forall i \in I, j \in J, p \in P, t \in T$

**Step 1.** $q^t_{\text{ijp}} = 0, \forall i \in I, j \in J, p \in P, t \in T$.

**Step 2.** $L \leftarrow \arg\ \max\{v^t_r, t \in T, r \in |T|\,(|I| + (|P|\,|J|))\}$.

**Step 3.** **If** $L$ belongs to the period $t$, then $t^* = t$.

**Step 4.** **If** $L \leq |I|$ for period $t^*$, then $i^* = L$,
$p^*, j^* = \arg\ \min\left\{c^{t^*}_{i^*jp}\,\middle|v^{t^*}_r \neq 0, r > |I|\right\}$,
**else** $L > |I|$ for period $t^*$, then $p^* = \left\lceil \frac{L-|I|}{|J|} \right\rceil$,
$j^* = L - |I| - ((p^* - 1)\,|J|)$,
$i^* = \arg\ \min\left\{c^{t^*}_{ij^*p^*}\,\middle|v^{t^*}_r \neq 0, r \leq |I|\right\}$.

**Step 5.** $q^{t^*}_{i^*j^*p^*} = \min\left\{d^{t^*}_{j^*p^*}, \left\lfloor \frac{s^{t^*}_{j^*}}{a_{p^*}} \right\rfloor\right\}$.

**Step 6.** $s^{t^*}_{i^*} = s^{t^*}_{i^*} - q^{t^*}_{i^*j^*p^*} \,.\, a_{p^*}$ and $d^{t^*}_{j^*p^*} = d^{t^*}_{j^*p^*} - q^{t^*}_{i^*j^*p^*}$.

**Step 7.** **If** $s^{t^*}_{i^*} = 0$ , then $v^{t^*}_{i^*} = 0$,
**If** $d^{t^*}_{j^*p^*} = 0$ , then $v^{t^*}_{j^*p^*} = 0$.

**Step 8.** **If** $v^t_{jp} = 0 \quad \forall j \in J, p \in P, t \in T$, then calculate transportation cost for the stage, **else** go to **step 2**.

Step 2 consists in looking for a node with the highest priority ($L$) in the current chromosome. Step 3 specifies a period ($t^*$) based on the highest priority node ($L$). Step 4 shows that if $L$ belongs to the source segment, then $L$ specifies a source ($i^*$) and finally, a product ($p^*$) and depot ($j^*$) are identified based on the lowest cost simultaneously. Otherwise, firstly, a product ($p^*$) and depot ($j^*$) are identified, then a source ($i^*$) is selected based on the lowest cost. Step 5 assigns the available amount of product $p^*$ that is shipped from source $i^*$ to depot $j^*$ in period $t^*$. Step 6 insures that the capacity of source $i^*$ and the demand of depot $j^*$ are updated. In step 7, the first part enforces the capacity restriction at source $i^*$, and the second part guarantees that the demand of depot $j^*$ is satisfied. Finally, step 8 insures that the demands of all depots are satisfied.

As it is seen in Figure 4c, at the first step of decoding procedure, an arc between Source 3 and Product 1 of Depot 1 is added to transportation tree since Product 1 of Depot 1 has highest priority in the chromosome and the lowest cost is between Product 1 of Depot 1 and Source 3 ($c_{311} = 1$). After determining the amount of shipment that is $q_{311} = \min\{30, 150\} = 30$, capacity of source and demand of depot are updated as $s_3 = 150 - 30 = 120$, $d_{11} = 30 - 30 = 0$, respectively. Since $d_{11} = 0$, the priority of Product 1 of Depot 1 is set to 0, and Product 2 of Depot 1 with next highest priority is selected. After adding arc between Product 2 of Depot 1 and Source 3 ($c_{312} = 2$), the amount of shipment between them is determined and their capacity and demand are updated as it is explained above, and this process repeats until demands of all depots are met.

Now, we compare the IPGA and the SPGA for a common transportation tree of a two-product three-source four-depot example, as shown in Figure 4 and Table 1, where it can be seen that the SPGA produces a better final solution in all scenarios.

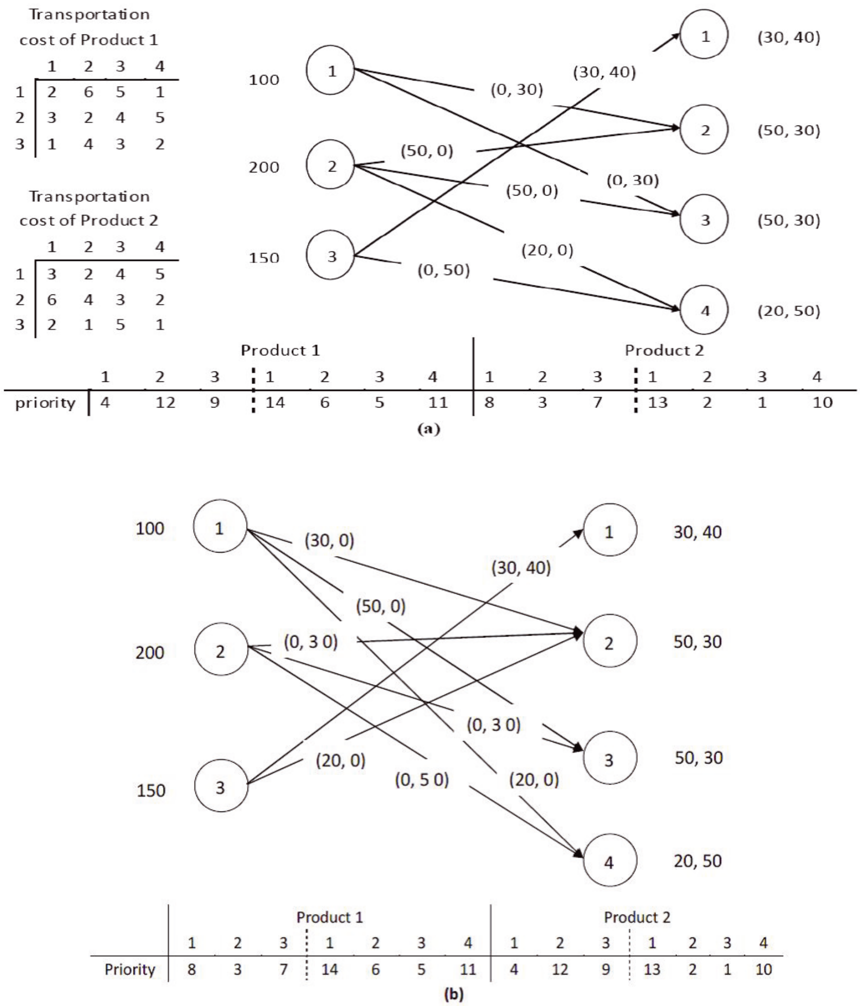
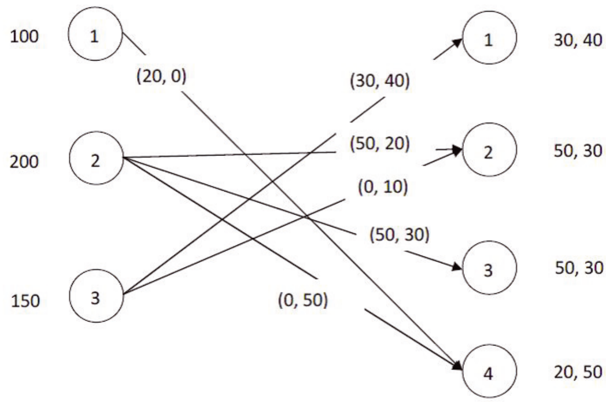



FIGURE 4. A sample transportation tree for the (a and b) initial and (c and d) straight GAs.

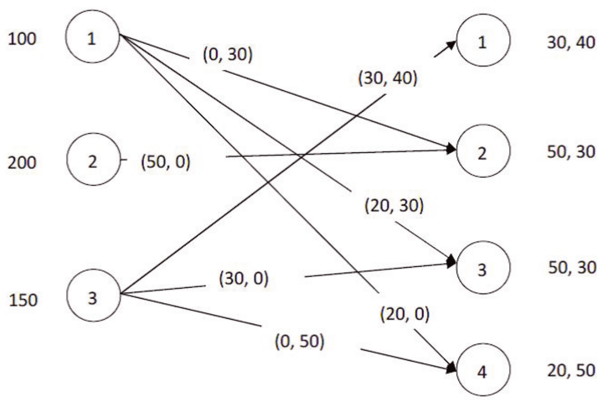
## 3.2. SELECTION

In multi-objective optimization, the computation of the fitness value of a chromosome is a key issue. It often results from a combination of the $n$ objective function-values, each denoted $z_i$ hereafter, and if necessary after the normalization of these values. One of the simplest methods to combine $n$ objective functions values into a scalar fitness value $z$ was presented by Murata *et al.* [14]. It is achieved




| | | | | Product 1 | | | | Product 2 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Priority | 3 | 9 | 6 | 11 | 5 | 4 | 8 | 10 | 2 | 1 | 7 |

**(c)**




| | | | | Product 1 | | | | Product 2 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Priority | 7 | 3 | 6 | 11 | 5 | 4 | 9 | 10 | 2 | 1 | 8 |

**(d)**

FIGURE 4. Continued.

using the following equations:

$$w_i = r_i / \sum\nolimits_{i=1}^{n} r_i \qquad i = 1, 2, \ldots, n \tag{3.1}$$

$$Z = \sum\nolimits_{i=1}^{n} w_i z_i. \tag{3.2}$$

Table 1. Total Transportation Cost.

| Algorithm | Scenario | Total Transportation Cost |
|---|---|---|
| IPGA | a | 740 |
| | b | 950 |
| SPGA | c | 710 |
| | d | 650 |

Where each $r_i$ is a non-negative number randomly chosen. We assign a random real number $r_i$ to equations (3.1) to calculate the weights $w_i$ and then a fitness value of each chromosome is calculated by equation (3.2). In equation (3.2), the constant weights $w_i$ make the search direction in GAs constant as well. Therefore, we use variable weights $w_i$ in the fitness procedure for various search directions.

The weighted sum $z_i$ in equation (3.2) is used for determining the selection probability of each chromosome. In this paper, roulette wheel selection is used as a selection mechanism. It is based on the selection or survival probability that is determined for each chromosome, proportional to the fitness value. The selection process is based on spinning the wheel a number of times, where the number of spins is equal to the difference between the population size and the number of elite solutions. In this method, with each spin, a single chromosome is selected for the new population. The above procedure is iterated $N$ (number of selection in each generation) times in each generation for selecting $N$ pairs of parents.

### 3.3. Elitism

Rudolph (1996) proved that GAs converge to the global optimal solution of some functions in the presence of elitism [3]. A tentative set of Pareto optimal solutions is preserved in the execution of our multi-objective GA. According to the fitness values, a tentative set of Pareto optimal solutions (selected from among the current population) is stored and carried over to the new population for elite protection. The other members of the new population are selected from among the individuals that are generated through crossover and mutation operations.

### 3.4. Crossover and mutation

In a crossover operation, two parents are picked from the last generation at random and some proportions of parents are interchanged to reproduce two offspring chromosomes. In this paper, a stage-based crossover operation is used. Under this operation, each stage of the offspring chromosome is randomly selected with equal probability from among the corresponding stages of parents. Each time a crossover operation is employed, two offspring chromosomes are generated which means they are complementary to each other.

Similar to crossover, mutation is done to prevent the premature convergence and explores a new solution [3]. It is usually done by modifying the genes within a chromosome. As in the case of crossover operations, stage-based mutation is used. In this operator, firstly, a decision about which segments and periods will be mutated is given with probably of 0.5 (*i.e.* using a binary mask), and then selected segments are mutated by swap operator. Swap operator selects two genes from the corresponding segment and period and then it exchanges their places.

### 3.5. Termination

The number of times the whole process (iteration) needs to be repeated will depend on the size of the problem. Therefore, the termination criterion is calculated based on the size of the problem.

The overall procedure for solving the multi-period, multi-product CLL problem is shown below.

**Step 0:** Initialization

- Generate an initial population (population size = 200).

**Step 1**: Evaluation

- Calculate the values of the objective functions for the generated chromosomes.
- Update the tentative set of Pareto optimal solutions.

**Step 2**: Selection

- Generate $n$ non-negative random real numbers ($n = 2$).
- Calculate the weights $w_i$ using equations (3.1).
- Calculate the fitness value $Z$ of each chromosome using equation (3.2).
- Calculate the selection probability based on the fitness values for each individual in the current population.
- Select a pair of parents from the current population using the roulette wheel selection method.

Note: Repeat **step 2** to select $N$ pairs of parents ($N = 50$).
**Step 3**: Crossover

- Apply the crossover operation to the selected parents to generate two offspring chromosomes.

**Step 4**: Mutation

- Apply the mutation operation to each chromosome generated by the crossover operation.

**Step 5**: Elitism

- Randomly, remove $M$ ($M = 10$) offspring chromosomes from the current population generated by the above operations.
- Randomly, add the same number of chromosomes (10 elite solutions) from the tentative set of Pareto optimal solutions to the current population.

TABLE 2. Test scenarios sizes.

| Scenario number | No. products | No. periods | No. plants | No. distribution centers | No. retailers | No. collection centers | No. recovery centers | No. recycling centers |
|---|---|---|---|---|---|---|---|---|
| 1st/2nd/ | 2/2/ | 2/2/ | 2/2/ | 2/3/ | 3/8/ | 2/3/ | 2/2/ | 2/2/ |
| 3rd/4th/5th/ | 2/3/3/ | 3/3/3/ | 2/2/3/ | 4/5/8/ | 12/15/20/ | 4/5/8/ | 3/3/4/ | 2/2/2/ |
| 6th/7th/ | 4/3/ | 3/3/ | 3/5/ | 12/15/ | 35/50/ | 12/15/ | 5/4/ | 3/2/ |
| 8th/9th/10th | 5/4/4 | 2/3/2 | 5/5/7 | 12/10/20 | 55/65/80 | 12/10/20 | 3/4/5 | 2/3/3 |

TABLE 3. Values of parameters used in the test scenarios.

| Parameter | Value | Parameter | Value | Parameter | Value |
|---|---|---|---|---|---|
| $\mathrm{DP}^t_{\mathrm{kp}}$ | U(80, 190) | $\mathrm{AC}_p$ | U(0.8, 1) | $\mathrm{TI}_{\mathrm{ijp}}$, $\mathrm{TJ}_{\mathrm{jkp}}$, $\mathrm{TK}_{\mathrm{klp}}$ | U(4, 10) |
| $\mathrm{AR}^t_{\mathrm{kp}}$ | U(0.6, 0.7) | $\mathrm{CI}_{ip}$ | U(500, 750) | $\mathrm{TL}_{\mathrm{lrp}}$, $\mathrm{TS}_{\mathrm{lsp}}$, $\mathrm{TR}_{\mathrm{rjp}}$ | U(4, 10) |
| $\mathrm{AS}^t_p$ | U(0.15, 0.20) | $\mathrm{CJ}_j$, $\mathrm{CL}_l$ | U(250, 350) | $\mathrm{TD}_{\mathrm{jkp}}$, $\mathrm{TC}_{\mathrm{klp}}$ | U(5, 8) |
| $\mathrm{FJ}^t_j$, $\mathrm{FL}^t_l$ | U(180 000, 260 000) | $\mathrm{CR}_r$ | U(200, 350) | $\mathrm{ED}^t_{\mathrm{kp}}$, $\mathrm{EC}^t_{\mathrm{kp}}$ | U(4, 6) |
| $\mathrm{FR}^t_r$ | U(300 000, 400 000) | $\mathrm{CS}_s$ | U(80, 150) | $\mathrm{PR}_{rp}$ | U(2, 4) |
| $\mathrm{FS}^t_s$ | U(150 000, 220 000) | $\mathrm{PI}_{ip}$ | U(3, 5) | CD, CC | 1 |
| $\mathrm{FH}^t_h$ | U(60 000, 100 000) | $\mathrm{PJ}_{jp}$, $\mathrm{PL}_{lp}$ | U(1.5, 3) | $\mathrm{MJ}^t_j$, $\mathrm{ML}^t_l$, $\mathrm{MR}^t_r$ | U(1–5) |
| $\mathrm{EJ}^t_j$, $\mathrm{EL}^t_l$ | U(20 000–50 000) | $\mathrm{ER}^t_r$ | U(30 000–70 000) | $\mathrm{GJ}_j$, $\mathrm{GL}_l$, $\mathrm{GR}_r$ | U(50–100) |
| $\mathrm{D}^t$ | $\{j \mid TD_{\mathrm{jkp}} \geq ED_{\mathrm{kp}}\}$ | $\mathrm{C}^t$ | $\{l \mid TC_{\mathrm{klp}} \geq EC_{\mathrm{kp}}\}$ | | |

**Step 6**: Termination

- **If** the evolution loop repeats $= 20 + \left\lfloor \frac{(I+J+K+L+R+S)(P+T)}{50} \right\rfloor$ times, then stop the run and move to the next step, **else**, return to Step 1.

**Step 7**: User selection

- The multi-objective GA provides the decision makers with the final set of Pareto optimal solutions.

## 4. Computational experiments

To compare the performance of the SPGA and the IPGA, the lower bound of the objective function in the CLL network is obtained using linear relaxation. Both GAs are coded in Java 1.6. CPLEX 12.2 optimization software is used to calculate the lower bound. They are tested on ten test scenarios of different sizes, as shown in Table 2. Other parameters are randomly generated using a uniform distribution, as specified in Table 3. Five instances are generated randomly for each size of test scenario. All the tests are carried out on a PC with 2.3 GHz CPU times and 1 GB of RAM.

Table 4 and Figure 5 show that the CPU times for the IPGA and the SPGA carried out for this paper are fairly consistent for a given problem size, and increase reasonably with increasing problem size. On the other hand, the CPU times using

TABLE 4. CPU time(s) for test results.

| Scenario | IPGA | | | Lower bound | SPGA | | |
|---|---|---|---|---|---|---|---|
| | Min | Average | Max | | Min | Average | Max |
| 1st | 22 | 22 | 22 | 18 | 20 | 20 | 20 |
| 2nd | 47 | 47 | 47 | 51 | 45 | 46 | 47 |
| 3rd | 114 | 116 | 117 | 814 | 112 | 116 | 120 |
| 4th | 242 | 244 | 246 | 1931 | 239 | 254 | 275 |
| 5th | 456 | 462 | 467 | 3417 | 522 | 551 | 582 |
| 6th | 1420 | 1438 | 1461 | 10 215 | 1659 | 1710 | 1779 |
| 7th | 1522 | 1585 | 1612 | 12 122 | 1756 | 1880 | 1920 |
| 8th | 2088 | 2121 | 2177 | 17 801 | 2380 | 2502 | 2598 |
| 9th | 2478 | 2510 | 2621 | 19 420 | 2881 | 2996 | 3102 |
| 10th | 2841 | 2980 | 3038 | 22480 | 3408 | 3590 | 3760 |

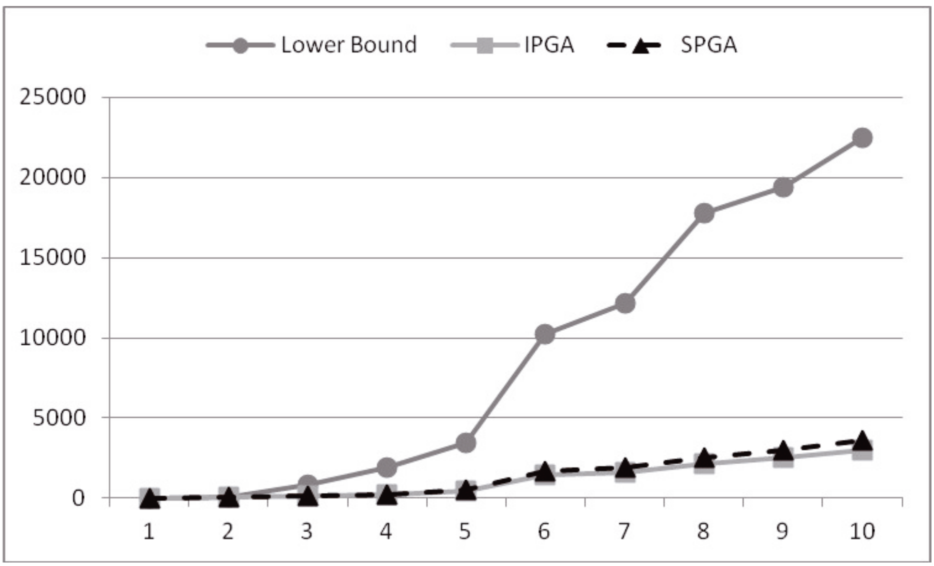



FIGURE 5. A comparison between the lower bound, the IPGA and the SPGA based on average CPU times.

CPLEX for the lower bounds deteriorate substantially as the problem size becomes large. For small-scale problems, the SPGA needs less CPU time than the IPGA. As the number of facilities increases, the growth in the CPU time for the IPGA is less than for the SPGA.

The gap between the average final solutions obtained by the two GAs and the lower bounds for each scenario are illustrated in Table 4 and Figure 6. It can be seen that, for the small-scale problems, the average gap between the SPGA and

Table 5. Final solutions for test results.

| Scenario | IPGA | | | Lower bound | SPGA | | |
|---|---|---|---|---|---|---|---|
| | Min | Average | Max | | Min | Average | Max |
| 1st | 8283 | 8378 | 8519 | 7637 | 7598 | 7659 | 7703 |
| 2nd | 17 546 | 17 792 | 18 483 | 17 405 | 17 337 | 17 713 | 18 182 |
| 3rd | 42 438 | 43 476 | 44 223 | 39 921 | 39 739 | 40 541 | 41 535 |
| 4th | 70 564 | 73 167 | 76 171 | 68 552 | 68 744 | 70 866 | 72 878 |
| 5th | 118 930 | 121 138 | 125 102 | 101 283 | 103 020 | 105 187 | 107 802 |
| 6th | 271 508 | 277 726 | 289 481 | 239 319 | 243 989 | 250 946 | 255 305 |
| 7th | 275 992 | 289 311 | 303 801 | 246 880 | 259 398 | 264 440 | 269 981 |
| 8th | 466 512 | 475 360 | 490 221 | 408 602 | 425 482 | 436 591 | 451 008 |
| 9th | 551 720 | 575 800 | 588 345 | 481 466 | 510 121 | 526 906 | 548 328 |
| 10th | 803 114 | 845 410 | 879 890 | 711 500 | 745 720 | 783 260 | 800 651 |

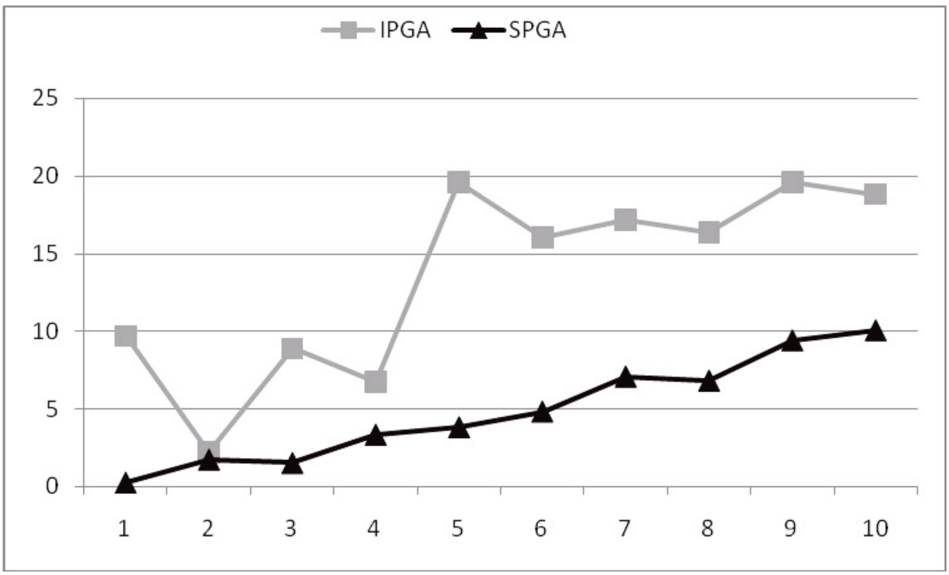



Figure 6. A comparison between the percentage gap of the IPGA and the SPGA to the lower bound based on the average final solutions.

the lower bound is less than 4%. For the other problems, the average gap ranges from 4% to 11%. However, the average final solution gap between the IPGA and the lower bound varies from 2% to 20% for all problems.

Comparing these two GAs with regard to the average final solutions, clearly, the SPGA outperforms the IPGA in all scenarios. Since the lower bound of the objective function is obtained by linear relaxation, there exists an inherent difference between the optimal solution and the lower bound. Therefore, the gap between the final solution and the lower bound is acceptable, which demonstrates the good quality of the final solution obtained by the SPGA.

## 5. Conclusions

This paper has proposed a deterministic mathematical model for a multi-period multi-product closed-loop logistics (CLL) problem. The problem is a strategic problem as it deals with facility location-allocation issues. We consider the issue of balancing cost against delivery/collection times by considering a multi-objective model. Moreover, the model supports facility expansion for each facility except for the plants and recycling centers and also considers cost savings associated with hybrid centers.

We have proposed the straight priority-based genetic algorithm (SPGA) to solve the model. The effectiveness of the SPGA has been investigated in detail by a comparison of its results against those obtained using the initial priority-based genetic algorithm (IPGA) and linear relaxation. The numerical results show that the SPGA outperformed the IPGA in our tests, at least in terms of the final solutions. Also, the comparison between the SPGA and the lower bound shows that the quality of solutions obtained by the SPGA for CLL problems is reasonable.

In our model, we have assumed parameters such as demand and capacity to be unchanged; future research may be required to include uncertainty in these parameters. When a single-objective solution method is applied to a multi-objective optimization problem, multiple objective functions should be combined into a scalar fitness function as we do in our solution method. Hence, identifying a solution method that will find all possible trade-offs among multiple objective functions (which are usually conflicting) could also be an interesting research direction.

## References

[1] F. Altiparmak, M. Gen, L. Lin and I. Karaoglan, A steady-state genetic algorithm for multi-product supply chain network design. *Comput. Ind. Engrg.* **56** (2009) 521–537.

[2] F. Altiparmak, M. Gen, L. Lin and T. Paksoy, A genetic algorithm approach for multi-objective optimization of supply chain networks. *Comput. Ind. Engrg.* **51** (2006) 196–215.

[3] K. Deb et al. *Multi-objective optimization using evolutionary algorithms*, Vol. 2012. John Wiley & Sons Chichester (2001).

[4] M. El.-Sayed, N. Afia and A. El.-Kharbotly, A stochastic model for forward–reverse logistics network design under risk. *Comput. Ind. Engrg.* **58** (2010) 423–431.

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

[6] M. Gen and R. Cheng, *Genetic algorithms and engineering optimization*, Vol. 7. John Wiley & Sons (2000).

[7] M. Gen and Y. Li, Spanning tree-based genetic algorithm for the bicriteria fixed charge transportation problem, in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, Vol. 3. IEEE (1999).

[8] V. Jayaraman, V.D.R. Guide Jr and R. Srivastava, A closed-loop logistics model for remanufacturing. *J. Oper. Res. Soc.* (1999) 497–508.

[9] Jakob Krarup and Peter Mark Pruzan. The simple plant location problem: survey and synthesis. *Eur. J. Oper. Res.* **12** (1983) 36–81.

[10] D.-Horng Lee, W. Bian and M. Dong, Multiobjective model and solution method for integrated forward and reverse logistics network design for third-party logistics providers. *Transp. Res. Record: Journal of the Transportation Research Board* **2032** (2007) 43–52.

[11] O. Listeş, A generic stochastic model for supply-and-return network design. *Comput. Oper. Res.* **34** (2007) 417–442.

[12] A. Marín and B. Pelegrín, The return plant location problem: Modelling and resolution. *Eur. J. Oper. Res.* **104** (1998) 375–392.

[13] Z. Michalewicz, G.A. Vignaux and M. Hobbs, A nonstandard genetic algorithm for the nonlinear transportation problem. *ORSA J. Comput.* **3** (1991) 307–316.

[14] Tadahiko Murata, Hisao Ishibuchi and Hideo Tanaka. Multi-objective genetic algorithm and its applications to flowshop scheduling. *Comput. Ind. Engrg.* **30** (1996) 957–968.

[15] M. Saman Pishvaee, R. Zanjirani Farahani and W. Dullaert, A memetic algorithm for bi-objective integrated forward/reverse logistics network design. *Comput. Oper. Res.* **37** (2010) 1100–1112.

[16] M. Saman Pishvaee, M. Rabbani and S. Ali Torabi, A robust optimization approach to closed-loop supply chain network design under uncertainty. *Appl. Math. Modell.* **35** (2011) 637–649.

[17] M.S. Pishvaee and S.A. Torabi, A possibilistic programming approach for closed-loop supply chain network design under uncertainty. *Fuzzy Sets Syst.* **161** (2010) 2668–2683.