RAIRO-Oper. Res. 45 (2011) 153–178
DOI: 10.1051/ro/2011105

RAIRO Operations Research
www.rairo-ro.org

A HYBRID GENETIC ALGORITHM FOR A DYNAMIC LOGISTICS NETWORK WITH MULTI-COMMODITIES AND COMPONENTS

Peng-Sheng You[1], Yi-Chih Hsieh[2] and Hisn-Hung Chen[1]

Abstract. Various topics related to reverse logistics have been discussed over the years. Most of them have assumed that facilities are kept open once they are established, and no returned products or recovery parts are stocked in intermediate recycling stations. However, firms may have the right to repeatedly open or close their facilities according to their economic benefits if they can acquire their facilities by lease. It also turns out that intermediate recycling stations like collection centers and disassembly centers usually stock returned products or parts in their facilities. By simultaneously relaxing these two assumptions, this study explores a logistics system with multiple items, each of which consists of some components among a variety of spare parts. The purpose is to maximize the total logistics costs by establishing a production schedule and reverse logistics framework over finite time periods for a logistics system. The mathematical model established in this study is a constrained linear integer programming problem. A genetic based algorithm is developed with the help of linear programming to find solutions to this problem. Limited computational experiments show that the proposed approach can produce better feasible solutions than the well-known CPLEX 10.0 software.

Keywords. Reverse logistics, genetic algorithm, constrained integer programming, production schedule, inventory.

Mathematics Subject Classification. 90-XX, 90-08.

Received September 30, 2008. Accepted May 24, 2011.

[1] Graduate Institute of Marketing & Logistics/Transportation, National Chia-Yi University, Taiwan. psyuu@mail.ncyu.edu.tw; goodjob8866@gmail.com

[2] Department of Industrial Management, National Formosa University, Taiwan. yhsieh@nfu.edu.tw

Article published by EDP Sciences

1. Introduction

With progress in science and technology, many new products have been developed and have become our cherished possessions. These products play important roles for people since they make people's lives more convenient. For all their importance, some of these products are quickly discarded when the latest models become available. For example, fashionable devices such as cell phones, personal digital assistants (PDAs) and computers are discarded when new versions become available. The incorporation of new products usually leads to the disposal of older products. However, some of the products discarded are still in good condition and their parts may be reusable. If these recyclable items are not put to good use, they will quickly lead to the accumulation of waste and bring about considerable negative impacts on the environment. To minimize the negative impact of waste, the concept of reuse has become a trend today.

Putting the concept of reuse into practice, many firms have incorporated reverse logistics systems into their manufacturing systems. This study deals with a multi-commodity inventory problem in which a firm can obtain its materials/components from outsourcing and its reverse system. Usually, reverse logistics systems involve the manufacturer, distribution center, customer, collector, asset recovery and recycler. In this study, the reverse logistics system focuses on scenario of product return, asset recovery, distributions, inventory and use. The product return involves the collection, product mixing and transfer of returned products at collection points, the asset recovery of returned products through disassembly, repair and refurbishing at disassembly centers, and remanufacturing by the manufacturer. Holding products or materials incurs inventory holding costs while handling returned products incurs not only inventory holding costs but also processing costs such as collections, refurbishments and repairs, transportation costs of transferring returned products from collection centers to disassembly centers, as well as the transportation cost of transferring refurbished parts from disassembly centers to the manufacturer.

To minimize logistics costs, enterprisers have to carefully determine production planning, facility location and distribution planning. In this study, the production planning problem is concerned with how many components should be purchased and how many products should be produced in each planning period. The facility location problem is concerned with where and when to set the facilities. The distribution planning problem is concerned with what amount of items should be shipped between facilities for each period of time.

Over the years, a number of studies have dealt with production planning problems. Readers are referred to the textbook by Silver *et al.* [28] and the recent survey paper by Jans and Degraeve [8]. In addition, Sodhi and Reimers' [29] study of production planning problem considered product recovery. The facility location and distribution planning problems studied here can be found in logistics or supply chain works. Roughly, the designs of facilities in logistics systems can be divided into static and dynamic structures. In the former structure, no change is incurred throughout the whole planning periods once a facility is established or opened. In the latter structure, closing or reopening of facilities is allowed during planning periods. A variety of relevant studies have discussed static structure. Among them, studies by Melachrinoudis *et al.* [19], Fleischmann *et al.* [6], Jayaraman *et al.* [9], Lieckens and Vandaele [17], Lu and Bostel [18], Salema *et al.* [27], Lee and Dong [16], Kusumastuti *et al.* [14].

Melachrinoudis *et al.* [19] developed a multiple period mixed integer programming model for a landfill facility problem. In their model, a facility can be opened at the beginning of any period while it cannot be closed once it has been opened. Lieckens and Vandaele [17] investigated a single product reverse logistics network in which the surplus of the supply of returned goods over demand is discarded. This problem was formulated as a mixed integer nonlinear program and a genetic algorithm was developed to deal with the problem. Lee and Dong [16] developed a deterministic programming model to deal with an integrated network with forward and reverse logistics. Their integrated network includes a single manufacturer and multiple hybrid processing facilities. Kusumastuti *et al.* [14] discussed a repair network for a computer manufacturer. The objective of this study is to determine the optimal locations of local sub-hubs and regional distribution centers, as well as the allocations of components among them, in order to minimize the total cost when the locations of service providers, original equipment manufacturers and the third-party repair vendors are known.

In addition to static models, a number of studies have also discussed dynamic structure. These include studies by Canel *et al.* [2], Chau [3], Ko *et al.* [12], Min *et al.* [23], Dias *et al.* [4], Ko and Evans [11], Hinojosa *et al.* [7], Min and Ko [22], etc.

The facility location problems are categorized as NP-hard problems [10,11]. Due to the computational complexity, finding an optimal solution in polynomial time for these problems are intractable. Thus, many heuristic approaches have been developed, which deal with variant location problems, including forward logistics problems [2,4,7,12,20], reverse logistics problems [14,19,22,23,25,31], and problems integrated with them [11,16]. Chau [3] used the genetic algorithm to deal with a construction facility allocation problem. A two stage solution approach was used to deal with the problem. The linear program was used in the first stage and a genetic algorithm was used in the second stage. Min *et al.* [23] employed the genetic algorithm approach to deal with a reverse logistics system with one centralized return center and multiple collection points. In their model, the firm can repeatedly open or close its collection points. Ko and Evans [11] proposed a genetic based heuristic to deal with a dynamic integrated distribution network. The integrated distribution network simultaneously contains a forward and a reverse logistics. In the forward flow, manufacturers produce products and sell them to customers through third party logistics providers. In the reverse flow, returned products are inspected and separated at repair centers and the collected products are shipped to manufacturers. In their study, facility opening and closing decisions are dynamic.

Hinojosa *et al.* [7] dealt with a facility location problem in which outsourcing and holding of inventory over consecutive time periods are allowed. Min and

Ko [22] employed a genetic based algorithm to deal with a reverse logistics design problem. The purpose of their study is to find the optimal location, number, and size of repair facilities under capacity limits and service requirements. In addition, Spengler *et al.* [30] dealt with a production planning problem that considered reverse logistics. Prahinski and Kocabasoglu [26] provided a literature review on reverse supply chains and suggested 10 research propositions.

To recapture the economic value of their products, many firms are encouraged to engage in product recovery. Some of these firms recover their products from collection sites, and then disassemble them totally or partially at disassembly sites. The recovered components are reused in new products, sold in secondary markets or recycled for other purposes. Langella [15] developed a heuristic to deal with a demand-driven disassembly planning problem for items with common component design. Their study did not consider facility location decisions. Melo *et al.* [20] dealt with a multi-commodity supply chain network with dynamic location and relocation of facilities. In their work, reverse logistics was not considered.

Problems associated with a logistics chain may arise during the distribution, production planning, or inventory stage. There have been several studies in the literature about the distribution and production planning stages. However, most studies on reverse logistics have begun with the assumption of transference of all recycled items to a next intermediate recycling stage. This assumption implies that there are no recycled items in stock in intermediate recycling stations such as collection centers and treatment centers. In fact, as far as an enterprise is concerned, intermediate recycling stations may have recycled items in stock. Therefore, inventory costs at intermediate recycling stations should be taken into consideration.

Reverse logistics systems with common components (*e.g.* [15]) and multi-commodity forward logistics systems with dynamic location and relocation of facilities (*e.g.* [20]) have been discussed. However, no work has been done on reverse logistics systems which are simultaneously compose of multi-product and multi-component, dynamic location and relocation of facilities, and inventory at intermediate recycling stations. Since this problem cannot be ignored, it is worthwhile to study a logistics system that is multi-product and multi-component with a dynamic reverse facility location. This study deals with a multi-product and multi-component with dynamic reverse facility location problem in relation to the outsourcing, inventory, and production planning stages. The decisions include (1) the production decision, (2) the component purchasing decision, (3) the dynamic facility location decision, (4) the returned products' distribution decision, and (5) the reusable components' distribution decision. The mathematical model established in this study is a constrained linear mixed integer planning model.

The facility location problems are categorized as NP-hard problems. This study allows analysis not only of dynamic facility location but also of outsourcing, inventory, and production planning. The additional allowance increases the computational complexity. Thus, the investigated problem also belongs to the class of NP-hard problems and cannot necessarily be solved optimally using either the traditional optimization method or any existing commercial optimization software

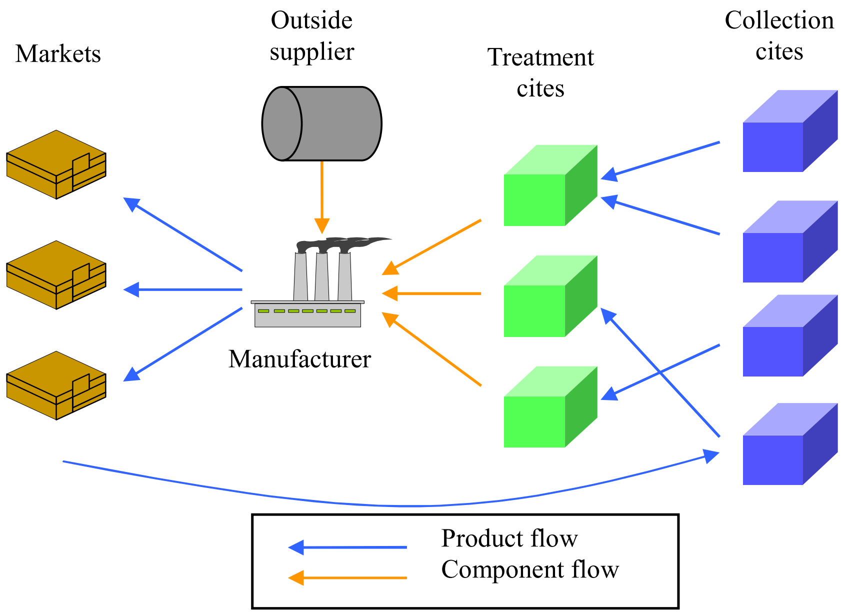

FIGURE 1. Location problem of a manufacturing system.

in polynomial time. This study develops a hybrid genetic algorithm to find feasible solutions to the mathematical model. The rest of this paper is organized as follows: Section 2 outlines all assumptions made and formulates the supply chain problem a constrained integer linear programming model. Section 3 then presents the solution methodology. Section 4 tests the performance of the proposed heuristic, using numerical examples to compare with the well-known commercial software, CPLEX 10.0. Conclusions are finally drawn in Section 5.

2. Model assumptions and description

The main differences of this model as compared to existing location models is that it simultaneously allows for production planning, multiple commodities and components, stocks in intermediate recycling stations and dynamic facility locations. The network structure of this model is depicted in Figure 1. The network consists of one manufacturer, N market regions, C potential collection-cites and D potential treatment cites. In the forward flow, the manufacturer produces J types of commodities and sells them to N market regions. These commodities are composed of some components among M types of components. The number of component-m required by commodity-j is n_{jm}. The manufacturer can get each type of component by outsourcing or from its reverse logistics system. Demand for commodity-j in market region-n in period t is assumed to be a constant value of d_{jnt} and all demands must be satisfied. In the reverse flow, the reverse logistics system starts with returned products from customers to collection-centers. The returned products from market region-n to candidate collection-site c are separated into returnable and un-returnable products. The fractions of the returnable and un-returnable returned products from market region-n to candidate collection-site c are assumed to be a constant value of r_n^c and a constant value of b_n^c, respectively. Symbols dr_{jt}^c and db_{jt}^c are respectively used to represent the amounts of returnable and un-returnable products collected from all market regions to collection-site c. Since both dr_{jt}^c and db_{jt}^c are integers, we make a simple assumption that dr_{jt}^c and db_{jt}^c are given by $\sum_{n=1}^{N} \llcorner d_{jnt} r_n^c \lrcorner$ and $\sum_{n=1}^{N} \lfloor d_{jnt} b_n^c \rfloor$, respectively. The disposal cost for commodity j at collection center c is τ_j^c per unit. To get components from returned products, the firm should ship the returnable commodities to treatment-centers for clearing, disassembling and refurbishing. All returnable commodities shipped to treatment-centers are transformed into reusable components. These components are viewed as new components and can be stocked in treatment-centers or shipped to the manufacturer to produce products.

In addition, we make the following assumptions. The locations of the collection-centers should be selected from C potential sites and the locations of the treatment-centers should be selected from D potential sites. For managerial purposes, it is assumed that a collection center is a dedicated facility to a treatment-center only. That is, the opening of a collection center only can be assigned to one treatment-center. No inventory is allowed in hand when a facility is closed. Inventory costs are based on an average inventory basis. Moreover, we assume that there are constraints on production and stock capacities. The production limit per period for commodity-j at manufacturer is $prod_j^F$, the warehousing limit for commodity-j at manufacturer is cap_j^F, the warehousing limitation for commodity-j at collection-center c is cap_j^c and the warehousing limitation for component-m at treatment-center d is cap_m^d.

The manufacturer aims to minimize total logistics costs over T planning periods by determining the reverse logistics structure, production planning, and the distribution amounts between facilities. The logistics costs are composed of the fixed operation and start-up cost for all collection centers, CSC, the total fixed operation and start-up cost for all treatment-centers, CSD, the operation and disposal costs incurred at all collection-centers, COC, the operation costs incurred at all treatment-centers, COD, the cost of shipping returned commodities from collection-centers to treatment-centers, $CTcd$, the cost of shipping components from treatment-centers to the manufacturer, $CTdF$, the commodity inventory cost at the manufacturer, HF, and the component inventory cost at the manufacturer, HP, the returned-commodity inventory cost for all collection-centers, HC, the component inventory costs for all treatment-centers, HD and outsourcing costs CO.

The decision process is addressed as follows. At the beginning of the each period, the decisions made by the manufacturer include (1) the production decision, which specifies the producing amount of each product, (2) the purchasing decision, which specifies the purchasing amount of each component from external suppliers, (3) the facility location decision, which determines whether to open/close closed/opened collection-centers, and determines whether to open/close closed/opened treatment-centers, (4) the returned products' distribution decision, which specifies the delivery amount of each returnable product between collection-centers and treatment-centers, and (5) the reusable components' distribution decision, which specifies the delivery amount of the reusable components between the treatment-centers and the manufacturer.

In addition to the notation denoted previously. We denote additional notation and summarize the notation denoted previously as follows.

Notation:

C: total number of potential collection-centers,

D: total number of potential treatment-centers,

N: total number of market regions,

J: total number of commodity types,

M: total number of component types,

T: total number of planning periods,

d_{jnt}: the demand amount of commodity-j at market region-n in period t,

dr_{jt}^{c}: the amount of commodity-j returned to collection-center c in period t,

c^c: the fixed operating cost per period for collection-center c,

c^d: the fixed operating cost per period for treatment-center d,

v^c: the start-up cost incurred by opening or reopening a collection-center at candidate collection-site c,

v^d: the start-up cost incurred by opening or reopening a treatment-center at candidate treatment-site d,

g_j^c: the variable cost incurred by the collection of one unit of product-j by collection-center c,

g_j^d: the variable cost incurred by the treatment of one unit of returned product-j by treatment-center d,

τ_j^c: unit disposed cost for returned commodity-j at collection-center c,

c_m: the unit outsourcing cost of component-m,

h_j^F: the unit inventory holding cost per period for commodity-j at manufacturer,

h_m^p: the unit inventory holding cost per period for component-m at manufacturer,

h_j^c: the unit inventory holding cost per period for returned commodity-j at collection-center c,

h_m^d: the unit inventory holding cost per period for component-m at treatment-center d,

z_{cj}^d: the shipping cost per unit of commodity-j per unit length from collection-center c to treatment-center d,

z_{dm}^F: the shipping cost per unit of component-m per unit length from treatment-center d to manufacturer,

w_c^d: the distance between collection-center c and treatment-center d,

w_d^F: the distance between treatment-center d and the manufacturer,

n_{jm}: the amount of component-m used by producing one unit of commodity-j.

$prod_j^F$: production capacity per period for commodity-j,

cap_j^F : stock capacity for commodity-j at manufacturer,

cap_j^c : stock limitation for commodity-j at collection-center c,

cap_m^d : stock limitation for component m at treatment-center d,

B: a very large number.

The decision variables of the problem are denoted as follows.

x_t^c: 1 if collection-center c is opened or reopened during period t, 0 otherwise,

$\tilde{x}_t^c$: a start up variable, 1 if collection-center c is reopened in period t, 0 otherwise,

s_t^d: 1 if treatment-center d is opened or reopened during period t, 0 otherwise,

$\tilde{s}_t^d$: a start up variable, 1 if treatment-center d is reopened in period t, 0 otherwise,

u_{ct}^d: 1 if both collection-center c and treatment-center d are opened and collection-center c is assigned to treatment-center d in period t, 0 otherwise,

Y_{cjt}^d: the quantity of commodity j that is shipped from collection-center c to treatment-center d at in period t,

Y_{dmt}^F: the quantity of component-m that is shipped from treatment-center d to manufacturer in period t,

Q_{jt}^F: the quantity of commodity-j produced by the manufacturer in period t,

O_{mt}: the quantity of component-m purchased from outside suppliers at the beginning of period t.

To develop the formulation for our model, we will first develop the inventory functions. Let I_{jt}^F, I_{mt}^p, I_{jt}^c and I_{mt}^d be the amount of commodity-j held in stock by the manufacturer at the end of period t, the amount of component-m held in stock by the manufacturer at the end of period t, the amount of returned commodity-j held by collection-center c at the end of period t, and the amount of component-m held by treatment-center d at the end of period t, respectively. Note that the initial inventory plus the goods input minus the goods output is equal to the ending inventory. Using the conditions of $I_{j0}^F = 0$, $I_{m0}^p = 0$, $I_{j0}^c = 0$

and $I^d_{m0} = 0$, and similar approaches used by Brahimi *et al.* [1], Eksioglu [5] and Jans and Degraeve [8], we can express the inventories as follows.

$$I^F_{jt} = \sum_{k=1}^{t} Q^F_{jk} - \sum_{n=1}^{N}\sum_{k=1}^{t} d_{jnk}, \ \forall\ j,t, \tag{2.1}$$

$$I^F_{mt} = \sum_{d=1}^{D}\sum_{k=1}^{t} Y^F_{dmk} - \sum_{j=1}^{J}\sum_{k=1}^{t} Q^F_{jk} n_{jm} + \sum_{k=1}^{t} O_{mk}, \ \forall\ m,t, \tag{2.2}$$

$$I^c_{jt} = \sum_{k=1}^{t} dr^c_{jk} x^c_k - \sum_{d=1}^{D}\sum_{k=1}^{t} Y^d_{cjk}, \ \forall\ c,j,t, \tag{2.3}$$

$$I^d_{mt} = \sum_{c=1}^{C}\sum_{j=1}^{J}\sum_{k=1}^{t} Y^d_{cjk} n_{jm} - \sum_{k=1}^{t} Y^F_{dmk}, \ \forall\ d,m,t. \tag{2.4}$$

Then, the formulation of the problem can be formulated as Problem $\mathbf{P}_1$.
Problem P_1.

$$\begin{aligned}\text{Min}\ :\ & CSC + CSD + COC + COD + CTcd + CTdF + HF + HP \\ & + HC + HD + CO\end{aligned} \tag{2.5}$$

where

$$CSC = \sum_{c=1}^{C}\sum_{t=1}^{T} c^c x^c_t + \sum_{c=1}^{C}\sum_{t=1}^{T} v^c \tilde{x}^c_t, \tag{2.6}$$

$$CSD = \sum_{d=1}^{D}\sum_{t=1}^{T} c^d s^c_t + \sum_{d=1}^{D}\sum_{t=1}^{T} v^d \tilde{s}^d_t, \tag{2.7}$$

$$COC = \sum_{c=1}^{C}\sum_{t=1}^{T} x^c_t \sum_{j=1}^{J} \Big((dr^c_{jt} + db^c_{jt}) g^c_j + db^c_{jt}\tau^c_j\Big), \tag{2.8}$$

$$COD = \sum_{c=1}^{C}\sum_{d=1}^{D}\sum_{j=1}^{J}\sum_{t=1}^{T} Y^d_{cjt} g^d_j, \tag{2.9}$$

$$CTcd = \sum_{c=1}^{C}\sum_{d=1}^{D}\sum_{j=1}^{J}\sum_{t=1}^{T} w_c^d Y_{cjt}^d z_{cj}^d, \tag{2.10}$$

$$CTdF = \sum_{d=1}^{D}\sum_{m=1}^{M}\sum_{t=1}^{T} w_d^F Y_{dmt}^F z_{dm}^F, \tag{2.11}$$

$$\begin{aligned} HF &= \sum_{j=1}^{J}\sum_{t=1}^{T} 0.5(I_{jt}^F + I_{j,t-1}^F)h_j^F \\ &= \sum_{j=1}^{J}\sum_{t=1}^{T}(T-t+0.5)Q_{jt}^F h_j^F - \sum_{j=1}^{J}\sum_{n=1}^{N}\sum_{t=1}^{T}(T-t+0.5)d_{jnt}h_j^F, \end{aligned} \tag{2.12}$$

$$\begin{aligned} HP &= \sum_{m=1}^{M}\sum_{t=1}^{T} 0.5(I_{mt}^p + I_{m,t-1}^p)h_m^p \\ &= \sum_{d=1}^{D}\sum_{m=1}^{M}\sum_{t=1}^{T}(T-t+0.5)Y_{dmt}^F h_m^p + \sum_{m=1}^{M}\sum_{t=1}^{T}(T-t+0.5)O_{mt}h_m^F \\ &\quad -\sum_{j=1}^{J}\sum_{m=1}^{M}\sum_{t=1}^{T}(T-t+0.5)Q_{jt}^F n_{jm} h_m^p, \end{aligned} \tag{2.13}$$

$$\begin{aligned} HC &= \sum_{c=1}^{C}\sum_{j=1}^{J}\sum_{t=1}^{T} 0.5(I_{jt}^c + I_{j,t-1}^c)h_j^c \\ &= \sum_{c=1}^{C}\sum_{j=1}^{J}\sum_{t=1}^{T}(T-t+0.5)dr_{jt}^c x_t^c h_j^c \\ &\quad -\sum_{c=1}^{C}\sum_{d=1}^{D}\sum_{j=1}^{J}\sum_{t=1}^{T}(T-t+0.5)Y_{cjt}^d h_j^c, \end{aligned} \tag{2.14}$$

$$\begin{aligned} HD &= \sum_{d=1}^{D}\sum_{m=1}^{M}\sum_{t=1}^{T} 0.5(I_{mt}^d + I_{m,t-1}^d)h_m^d \\ &= \sum_{c=1}^{C}\sum_{d=1}^{D}\sum_{j=1}^{J}\sum_{m=1}^{M}\sum_{t=1}^{T}(T-t+0.5)Y_{cjt}^d n_{jm} h_m^d \\ &\quad -\sum_{d=1}^{D}\sum_{m=1}^{M}\sum_{t=1}^{T}(T-t+0.5)Y_{dmt}^F h_m^d, \end{aligned} \tag{2.15}$$

$$CO = \sum_{m=1}^{M}\sum_{t=1}^{T} O_{mt} c_m, \tag{2.16}$$

subject to the following constraints:

$$\tilde{x}_t^c \geq x_t^c - x_{t-1}^c, \ \forall \ c, t, \tag{2.17}$$

$$\tilde{s}_t^d \geq s_t^d - s_{t-1}^d, \ \forall \ d, t, \tag{2.18}$$

$$\sum_{d=1}^{D} u_{ct}^d = x_t^c, \ \forall \ c, t, \tag{2.19}$$

$$\sum_{c=1}^{C} u_{ct}^d \geq s_t^d, \ \forall \ d, t, \tag{2.20}$$

$$u_{ct}^d \leq x_t^c, \ \forall \ c, d, t, \tag{2.21}$$

$$u_{ct}^d \leq s_t^d, \ \forall \ c, d, t, \tag{2.22}$$

$$Y_{cjt}^d \leq u_{ct}^d B, \ \forall \ c, d, j, t, \tag{2.23}$$

$$Y_{cjt}^d \geq u_{ct}^d, \ \forall \ c, d, j, t, \tag{2.24}$$

$$Y_{dmt}^F \leq s_t^d B, \ \forall \ m, d, t, \tag{2.25}$$

$$Y_{dmt}^F \geq s_t^d, \ \forall \ m, d, t, \tag{2.26}$$

$$\sum_{k=1}^{t} dr_{jk}^c x_k^c - \sum_{d=1}^{D} \sum_{k=1}^{t} Y_{cjk}^d \leq x_t^c B, \ \forall \ c, j, t, \tag{2.27}$$

$$\sum_{c=1}^{C} \sum_{j=1}^{J} \sum_{k=1}^{t} Y_{cjk}^d n_{jm} - \sum_{k=1}^{t} Y_{dmk}^F \leq s_t^d B, \ \forall \ d, m, t, \tag{2.28}$$

$$Q_{jt}^F \leq Prod_j, \ \forall \ j, t, \tag{2.29}$$

$$\sum_{k=1}^{t} Q_{jk}^F - \sum_{n=1}^{N} \sum_{k=1}^{t} d_{jnk} \leq cap_j^F, \ \forall \ j, t, \tag{2.30}$$

$$\sum_{d=1}^{D} \sum_{k=1}^{t} Y_{dmk}^F - \sum_{j=1}^{J} \sum_{k=1}^{t} Q_{jk}^F n_{jm} + \sum_{k=1}^{t} O_{mk} \leq cap_m^F, \ \forall \ m, t, \tag{2.31}$$

$$\sum_{k=1}^{t} dr_{jk}^c x_k^c - \sum_{d=1}^{D} \sum_{k=1}^{t} Y_{cjk}^d \leq cap_j^c, \ \forall \ c, j, t, \tag{2.32}$$

$$\sum_{c=1}^{C} \sum_{j=1}^{J} \sum_{k=1}^{t} Y_{cjk}^d xn_{jm} - \sum_{k=1}^{t} Y_{dmk}^F \leq cap_m^d, \ \forall \ d, m, t, \tag{2.33}$$

$$\sum_{k=1}^{t} Q_{jk}^{F} - \sum_{n=1}^{N}\sum_{k=1}^{t} d_{jnk} \geq 0, \ \forall\ j,t, \tag{2.34}$$

$$\sum_{d=1}^{D}\sum_{k=1}^{t} Y_{dmk}^{F} - \sum_{j=1}^{J}\sum_{k=1}^{t} Q_{jk}^{F} n_{jm} + \sum_{k=1}^{t} O_{mk} \geq 0, \ \forall\ m,t, \tag{2.35}$$

$$\sum_{k=1}^{t} dr_{jk}^{c} x_{k}^{c} - \sum_{d=1}^{D}\sum_{k=1}^{t} Y_{cjk}^{d} \geq 0, \ \forall\ c,j,t, \tag{2.36}$$

$$\sum_{c=1}^{C}\sum_{j=1}^{J}\sum_{k=1}^{t} Y_{cjk}^{d} xn_{jm} - \sum_{k=1}^{t} Y_{dmk}^{F} \geq 0, \ \forall\ d,m,t, \tag{2.37}$$

$$x_{0}^{c} = s_{0}^{d} = 0, \ \forall\ c,d, \tag{2.38}$$

$$x_{t}^{c}, s_{t}^{d}, U_{ct}^{d} \in \ \{0,1\}, \ \forall\ c,d,t. \tag{2.39}$$

In addition, the values of O_{mt}, Y_{cjt}^{d} and Y_{dmt}^{F} are required to be non-negative integers for all c, d, j, m and t. Note the value of 0.5 in HD represents the situation that the inventory holding cost is based on average inventory level.

Equations 2.17 and 2.18 guaranties that start-up cost are incurred if and only if a facility is opened or reopened. The reason is as follows. The possible combinations of (x_{t}^{c}, x_{t-1}^{c}) are (0,0), (0,1), (1,0) and (1,1). The value of $\tilde{x}_{t}^{c}$ should be one for the combination of (1,0) and should be zero for the combinations of (0,0), (0,1) and (1,1). From the constraint of $\tilde{x}_{t}^{c} \geq x_{t}^{c} - x_{t-1}^{c}$, we see that the value of $\tilde{x}_{t}^{c}$ must be one for the combination of (1,0) and can be one or zero for other combinations. Since the purpose is to minimize the total logistics costs, $\tilde{x}_{t}^{c} = 1$ increase the total cost. Thus, the value of $\tilde{x}_{t}^{c}$ must be zero for the combinations of (0,0), (0,1) and (1,1). Accordingly, constraint (2.17) is sufficient to control the value of $\tilde{x}_{t}^{c}$. Using the same logic, constraint (2.18) can be applied to determine the value of $\tilde{s}_{t}^{d}$. Equation (2.19) guarantees that collection center c must be assigned to exactly one treatment-center if collection-center c is opened; equation (2.20) guarantees that at least one collection-center is assigned to treatment-center d if treatment-center d is opened; equation (2.21) guarantees that collection center c should not assigned to any treatment-center if collection-center c is closed; equation (2.22) guarantees that no collection-center ships returned commodities to treatment-center d if treatment-center d is closed; equation (2.23) guarantees that no returned commodity is shipped from collection-center c to treatment-center d if collection-center c is not assigned to treatment-center d; equation (2.24) guarantees that the quantity of returned commodities shipped from collection-center c to treatment-center d must be larger then zero if collection-center c is assigned to treatment-center d; equation (2.25) guarantees that no component is shipped from treatment-center d to the manufacturer if treatment-center d is not open; equation (2.26) guarantees that the quantities of components shipped from treatment-center d to the manufacturer must be larger than zero if treatment-center d is open; equation (2.27) guarantees that no returned commodity is held in collection-center c if collection-center c is closed; equation (2.28) guarantees that no component is held in treatment-center d if treatment-center d is closed; equations (2.29) and (2.30) respectively represent the production limit and the warehousing limit for commodities at the manufacturer; equation (2.31) represents the warehousing limit for components at the manufacturer; equation (2.32) represents the warehousing limit for returned commodities at collection-centers; equation (2.33) represents the warehousing limit for components at treatment-centers; equations (2.34) to (2.37) ensures that the inventory levels are nonnegative; (2.34) also ensures that demands must be satisfied; equation (2.38) specifies the initial facility status; equation (2.39) specifies the binary nature of the parameters.

3. Solution methology

The model developed in this study is a generalization of a simple plant location problem, so it can be described as a NP-hard problem (Krarup and Pruzan [13]). We develop a hybrid genetic algorithm to deal with this problem.

3.1. Hybrid genetic algorithm

Problem $\mathbf{P}_1$ is a constrained integer linear programming model. Due to the computational complexity of the model, there is no guarantee that any approach can solve the problem optimally within a polynomial time. To overcome this difficulty, we have developed a solution approach to obtain a compromise within a reasonable CPU time. The solution approach is an iterative method in which the concepts of the genetic algorithm (GA) and linear programming are employed. GA is an iterative optimization procedure that mimics the process of natural evolution. GAs are a particular class of evolutionary algorithms, which generate exact or approximate solutions to optimization and search problems using techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover. For details, readers are referred to two textbooks by Onwubolu and Babu [24] and Michalewicz [21].

At each iteration, the values of x_t^c, s_t^d and u_{ct}^d are determined by GA. Then, this paper establishes **Problem** P_2 by substituting the values of x_t^c, s_t^d and u_{ct}^d into **Problem** P_1. Then, we solve **Problem** P_2 by relaxing the integer constraints. Finally, we round the relaxed decision variables to obtain feasible solutions of **Problem** P_1. The details of the heuristic are addressed as follows.

3.1.1. *Encoding*

Designing a suitable chromosome is the first step in implementing a successful GA. Each chromosome is represented as a single dimensional array with CT genes. Each gene is randomly generated by an integer number within the range of $[0, D]$. The value of the $(t-1)C+c$-th entry of a chromosome represents whether or not collection-center c is assigned to treatment-center d in period t. For example, if the number of candidate collection sites, the number of candidate treatment sites

TABLE 1. An encoding example.

$t = 1$			$t = 2$			$t = 3$		
1	0	2	0	1	4	0	0	1

TABLE 2. A description of crossover operator.

Individual	Chromosome								
	$t = 1$			$t = 2$			$t = 3$		
Parent 1	1	0	2	0	1	4	0	0	1
Parent 2	0	1	3	1	0	2	1	2	0
Offsprint 1	1	0	3	0	0	2	0	2	0
Offsprint 2	0	1	2	1	1	4	1	0	1

and the number of planning periods are $C = 3$, $D = 4$ and $T = 3$, respectively, then, the number of the total genes of a chromosome is $CT = 9$ and the possible value of each gene is within the range of 0 to 4. Table 1 is a description of the encoding operator.

3.1.2. *Genetic operators*

Four genetic operators are used in our proposed heuristic. These operators are described as follows.

1. Cloning operator:
   In our genetic algorithm, we first select the best K individuals and directly copy them to the next generation, and then produce the remaining $Num_{pop} - K$ individuals by parent selection, crossover and mutation operators.
2. Parent selection:
   Roulette-wheel selection is used to produce the mating pool to produce the remaining individuals.

3.1.3. *Crossover operator*

1. Sequentially choose two individuals from the mating pool for crossover.
2. For each period, generate a random integer number *pos* from the range of $[0, C - 1]$. The number *pos* indicates the position of the crossing point. Table 2 is a description of the crossover operator. In this table, we assume that the crossing points are 2, 1 and 1 for $t = 1$, $t = 2$ and $t = 3$, respectively.

3.1.4. *Mutation operator*

1. For each crossed individual, generate a random number r within the range $[0, 1]$. If $r < \rho$, generate two random integer numbers k_1 and k_2 within $[1, CT]$ and $[0, D]$, respectively, and replace the number at k_1th bit with the number k_2.

TABLE 3. A decoding operation.

x_t^c	c			s_t^d	d			
t	1	2	3	t	1	2	3	4
1	0	1	1	1	1	1	0	0
2	1	1	1	2	1	0	0	1
3	1	0	1	3	1	0	0	0

TABLE 4. A decoding operation.

u_{c1}^d	d				u_{c2}^d	d				u_{c3}^d	d			
c	1	2	3	4	c	1	2	3	4	c	1	2	3	4
1	0	0	0	0	1	1	0	0	0	1	1	0	0	0
2	1	0	0	0	2	1	0	0	0	2	0	0	0	0
3	0	1	0	0	3	0	0	0	1	3	1	0	0	0

2. Copy the individual to the next generation.

3.1.5. *Decoding*

The heuristic initializes a population with Num_{pop} chromosomes. Each chromosome has CT genes and each gene is randomly generated by an integer number E within the range of $[0, D]$. The integer number E may be a zero or a positive number. Integer number $E = 0$ means that no collection-center is located at candidate site c. That is, $x_t^c = 0$. Integer number $E > 0$ means that a collection-center is located at candidate site c, a treatment-center is located at candidate treatment site E and collection-center c is assigned to treatment-center E. That is, $x_t^c = 1$, $s_t^E = 1$ and $u_{ct}^E = 1$. Consider the previous example in Table 2. The chromosome of offspring 2 which is encoded as $(0, 1, 2, 1, 1, 4, 1, 0, 1)$ can be decoded as those in Tables 3 and 4.

Tables 3 and 4 show that in period 1, collection-centers 2 and 3 are opened and assigned to treatment-centers 1 and 2, respectively; in period 2, collection-centers 1, 2 and 3 are opened and assigned to treatment-centers 1, 1 and 4, respectively; in period 3, collection-centers 1 and 3 are opened and both of them are assigned to treatment-center 1.

3.1.6. *Fitness function*

To measure the goodness of a chromosome, we have to compute its fitness value. In problem $\mathbf{P}_1$, the values of x_t^c, s_t^d and u_{ct}^d are binary decision variables and the decision variables of O_{mt}, Q_{jt}, Y_{cjt}^d and Y_{dmt}^F are required to be integers. Substituting the values of x_t^c, s_t^d and u_{ct}^d into problem $\mathbf{P}_1$, making it a more simplified model. We refer to this model as Problem $\mathbf{P}_2$.

At each iteration, we relax the integer constraints on decision variables O_{mt}, Q_{jt}, Y_{cjt}^d and Y_{dmt}^F in Problem $\mathbf{P}_2$. We refer to such a problem as problem $\mathbf{P}_3$. It is clear that $\mathbf{P}_3$ is a linear programming problem and can be solved optimally.

TABLE 5. Problem types.

Type	C	D	J	M	N	T
1	3	3	3	3	3	3
2	5	5	5	5	5	5
3	5	5	5	5	5	10
4	8	8	8	8	8	8
5	10	10	10	10	10	10

However, the values of O_{mt}, Q_{jt}, Y^d_{cjt} and Y^F_{dmt} are not necessarily integers. Thus, the solution obtained from $\mathbf{P}_3$ may be an infeasible solution for problem $\mathbf{P}_2$. For the integer constraints to be satisfied, we round off the values of O_{mt}, Q_{jt}, Y^d_{cjt} and Y^F_{dmt}. However, this may lead to the problem that some constraints may be violated. To reduce the probability that chromosomes with infeasible solutions are copied to the next generation, the fitness function is evaluated by adding a penalty to the original objective function. The penalty function is assumed to be a linear function of the amount by which a constraint's boundary is violated. For each violation, the amount of the penalty is determined by the violation amount times a fixed penalty cost η_e where e is the equation number of an inequality constraint. For example, if the violation amount for constraint (6) is $\eta_6 = 2$ units and the unit penalty cost is set at 5, then the penalty is 10.

4. Computational study

In this section, the proposed approach was applied to solve some fictitious problems. Six problem types, which are described in Table 5, have been considered. For each of these problem types, the number of potential collection cites, C, the number of potential treatment cites, D, the number of market regions, N, the number of products, J and the number of planning periods, T are given, and 20 problem instances were generated. Due to extensive data requirements, for the sake of simplification, we assumed that the values of the parameters of d_{jnt}, r^c_n, b^c_n, c^c, c^d, v^c, v^d, g^c_j, g^d_j, c_m, h^F_j, h^p_m, h^c_j, h^d_m, z^d_{cj}, z^F_{dm}, w^d_c, w^F_d, n_{jm}, $prod^F_j$, cap^F_j, cap^c_j and cap^d_m of the first problem instance in each problem type were generated according to Table 6.

The parameters of the problem instances 2–20 in each problem type were generated in the same way as the first problem instance except that one of the parameter generating rules was varied. The different parameters and the generating rules for problem instances 2–20 are shown in Table 7. For these six problem types, problem type 1 has the smallest problem size and problem-type 2 has the second smallest problem size, and so on.

To measure the performance of the proposed HGA (hybrid genetic algorithm), the commercial software GAMS/CPLEX modeling language was adopted for the purpose of feasible solution comparison with the proposed HGA for all problem types. The proposed HGA was coded in Visual C++ 6.0 programming language

TABLE 6. Values of the parameters for the first test problem in each problem type.

$n_{jm} = m$	$v^c = 8000 + 800c$
$c_m = 5 + 0.5m$	$v^d = 15\,000 + 100(D - d)$
$prod_j^F = 100\,000 + 1000j$	$h_j^F = 0.02 + 0.01j$
$cap_j^F = 100\,000 + 1000j$	$h_m^p = 0.01 + 0.005m$
$cap_m^p = 50\,000 + 500m$	$h_m^d = 0.005 + 0.002m + 0.001d$
$cap_m^d = 50\,000 + 500(D - d)$	$h_j^c = 0.005 + 0.002j + 0.001c$
$cap_j^c = 50\,000 + 500c$	$g_j^c = 0.5 + 0.2(J - j) + 0.2(C - c)$
$d_{jnt} = 5000 + 20(j + n + t)$	$g_j^d = 1 + 0.2(J - j) + 0.1(D - d)$
$r_n^c = (0.3 - 0.01\|n - c\|)/C$	$\tau_c^j = 0.05$
$w_d^F = 5 + 0.5d$	$w_c^d = 5 + 0.5\|c - d\|$
$z_{cj}^d = 0.1 + 0.05j$	$z_{dm}^F = 0.02 + 0.01m$
$c^c = 30\,000 + 1000c$	$c^d = 50\,000 + 1000d$

TABLE 7. Replaced parameters and the generating rule for test cases 2–20.

		11	$w_c^d = 10 + 0.5\|c - d\|$
2	$c_m = 5.5 + 0.5m$	12	$w_c^d = 12 + 0.5\|c - d\|$
3	$c_m = 6.0 + 0.5m$	13	$w_c^d = 15 + 0.5\|c - d\|$
4	$c_m = 7.0 + 0.5m$	14	$z_{cj}^d = 0.2 + 0.05j$
5	$d_{jnt} = 5500 + 20j + 20n + 20t$	15	$z_{cj}^d = 0.25 + 0.05j$
6	$d_{jnt} = 6000 + 20j + 20n + 20t$	16	$z_{cj}^d = 0.3 + 0.05j$
7	$d_{jnt} = 6500 + 20j + 20n + 20t$	17	$v^d = 18\,000 + 100(D - d)$
8	$w_d^F = 10 + 0.5d$	18	$v^d = 20\,000 + 100(D - d)$
9	$w_d^F = 12 + 0.5d$	19	$c^c = 35\,000 + 1000c$
10	$w_d^F = 15 + 0.5d$	20	$c^c = 40\,000 + 1000c$

and along with the GAMS/CPLEX model were implemented on an Intel Core 2 Duo personal computer equipped with a speed of 2.4 GHz and 2GB of memory. For identifying the gaps between the results obtained from CPLEX and HGA for optimal solution, the Lingo global solver [32], which can identify the global minima, was used to solve all problems. For practical concerns, all algorithms were terminated if the execution time exceeded 10 h.

Problem instances in problem type 1 were run to determine the optimal combinations of population size, maximum number of generations, the crossover rate, mutation rate and maximum number of iterations. The parameters of HGA after the tests were set as follows. The population size was equaled to 14; the maximum number of iterations was set to be $5T$; the cloning parameter K was set at 3; the crossover rate was set at 100%; the mutation rate was set at 5%; the penalty values of η_es were set at 5 for all e.

To explain the application of the decisions, we show the dynamic facility and shipping decisions for the first problem instance of problem-type 1 that was solved by HGA. The computational result shows that $x_t^c = 1$ for all t and c, $s_t^d = 0$ for

TABLE 8. Values of O_{mt} for the first case.

m	$t=1$	$t=2$	$t=3$
1	33 054	33 186	33 316
2	66 108	66 372	66 632
3	99 162	99 558	99 948

TABLE 9. Values of Q^F_{jt} for the first case.

j	$t=1$	$t=2$	$t=3$
1	15 240	15 300	15 360
2	15 300	15 360	15 420
3	15 360	15 420	15 480

TABLE 10. Values of Y^3_{cjt} for the first case.

	$c=1$			$c=2$			$c=3$		
j	$t=1$	$t=2$	$t=3$	$t=1$	$t=2$	$t=3$	$t=1$	$t=2$	$t=3$
1	1422	1427	1432	1422	1427	1433	1422	1428	1433
2	1427	1432	1438	1427	1433	1438	1428	1433	1438
3	1432	1438	1444	1433	1438	1444	1433	1438	1444

TABLE 11. Values of Y^F_{3mt} for the first case.

m	$t=1$	$t=2$	$t=3$
1	12 846	12 894	12 944
2	25 692	25 788	25 888
3	38 538	38 682	38 832

all t and d except for $s^3_t = 1$ for all t. This implies that collection centers 1, 2 and 3, and treatment-center 3 were opened in period 1 and were kept open throughout the entire planning periods. The production and purchasing schedules are shown in Tables 8 and 9, respectively. The amount of returned products shipped from collection-centers to treatment-center 3 is shown in Table 10, and the amount of components shipped from treatment-center 3 to the manufacturer is shown in Table 11. For example, the amount of returned commodity-1 shipped from collection-center 1 to treatment-center 2 was 1422 in period 1.

The exact solution approach using Lingo global solver was only able to solve problem types 1 an 2 optimally. However, it failed to produce global solutions for problem types 3 to 6 after 10 h of computational time due to the complexity of the problem structure.

For problem types 1 and 2, the gaps between the heuristic solutions and global solutions, as well as those between the CPLEX solutions and the global solutions in terms of solution qualities and computational times for problem types 1 and 2 are shown in Tables 12 and 13, respectively. The criteria of performances considered were the quality of the total logistics cost and the amount of CPU time (s).

TABLE 12. Computational result for problem type 1.

No.	Optimal		GAMS		HGA		Gap	
	Sol	Time	Sol	Time	Sol	Time	S-CG	S-HG
1	4 363 171	6	4 363 171	3	4 363 171	21	0.00%	0.00%
2	4 661 839	10	4 661 839	3	4 661 839	28	0.00%	0.00%
3	4 960 507	10	4 960 507	3	4 960 507	28	0.00%	0.00%
4	5 557 843	7	5 557 843	3	5 557 843	27	0.00%	0.00%
5	4 742 128	8	4 742 128	3	4 742 128	27	0.00%	0.00%
6	5 121 093	13	5 121 263	3	5 121 093	27	0.00%	0.00%
7	5 500 000	13	5 500 000	3	5 500 000	27	0.00%	0.00%
8	4 413 460	8	4 413 460	3	4 413 460	27	0.00%	0.00%
9	4 433 576	10	4 433 576	3	4 433 576	27	0.00%	0.00%
10	4 463 750	10	4 463 750	2	4 463 750	27	0.00%	0.00%
11	4 401 880	10	4 401 880	3	4 401 880	27	0.00%	0.00%
12	4 417 363	7	4 417 363	4	4 417 363	27	0.00%	0.00%
13	4 440 588	5	4 440 588	3	4 440 588	27	0.00%	0.00%
14	4 384 447	7	4 384 452	3	4 384 447	27	0.00%	0.00%
15	4 394 768	10	4 394 768	3	4 394 768	27	0.00%	0.00%
16	4 405 084	9	4 405 084	3	4 405 084	27	0.00%	0.00%
17	4 366 171	12	4 366 821	2	4 366 171	27	0.01%	0.00%
18	4 368 171	10	4 368 821	2	4 368 171	27	0.01%	0.00%
19	4 408 171	11	4 408 171	4	4 408 171	27	0.00%	0.00%
20	4 453 171	8	4 453 171	4	4 453 171	27	0.00%	0.00%
Average		9.2		2.8		27.1	0.00%	0.00%

The solution percentage gap, defined as 100 (HGA (or CPLEX) solution – global solution)/(global solution) percentage points, was used to evaluate the solution quality of the HGA (or CPLEX) for samll size problems. The symbols of S-CG and S-HG are used to represent the solution percentage gaps between the exact solutions obtained by Lingo global solver and CPLEX's feasible solutions, and between the exact solutions and the HGA's feasible solutions, respectively. In addition, the time gap, (defined as the CPU time used by Lingo global solver – the CPU time used by HGA or CPLEX), was used to evaluate the efficiency of the proposed heuristic.

In terms of running time, from Table 12, we see that the average computational time to solve test cases in problem type 1 are 9.2, 2.8 and 27.1 for Lingo Global solver, CPLEX solver and HGA, respectively. We can also conclude from Table 12 that CPLEX outperformed HGA, and the Lingo Global solver outperformed HGA for problem type 1. From Table 13, we see that problem type 2 was solved in an average of 164.2, 7.1 and 135.2 seconds by the Lingo Global solver, CPLEX solver and HGA, respectively. From Table 13, we see that CPLEX outperformed the HGA, and HGA outperformed the Lingo Global solver for problem type 2. In general, either approach will need more computational time to solver the testing problems when the problem size increases. However, we can find from these two

Table 13. Computational result for problem type 2.

	Optimal		GAMS		HGA		Gap	
No.	Sol	Time	Sol	Time	Sol	Time	S-CG	S-HG
1	51 878 836	208	51 878 836	7	51 878 836	114	0.00%	0.00%
2	55 472 836	205	55 472 836	7	55 472 836	108	0.00%	0.00%
3	59 066 836	126	59 066 836	8	59 066 836	128	0.00%	0.00%
4	66 254 836	144	66 254 836	6	66 254 836	138	0.00%	0.00%
5	56 773 768	146	56 773 768	8	56 773 768	136	0.00%	0.00%
6	61 668 699	171	61 668 699	6	61 668 699	138	0.00%	0.00%
7	66 563 630	138	66 563 630	7	66 563 630	139	0.00%	0.00%
8	52 594 111	121	52 594 111	8	52 594 111	141	0.00%	0.00%
9	52 880 221	159	52 880 221	9	52 880 221	138	0.00%	0.00%
10	53 309 386	123	53 309 386	7	53 309 386	138	0.00%	0.00%
11	52 089 536	247	52 089 536	7	52 089 536	139	0.00%	0.00%
12	52 173 816	207	52 173 816	7	52 173 816	136	0.00%	0.00%
13	52 300 236	169	52 300 236	7	52 300 236	143	0.00%	0.00%
14	51 979 819	146	51 979 819	8	51 979 819	139	0.00%	0.00%
15	52 030 310	154	52 030 310	8	52 030 310	145	0.00%	0.00%
16	52 080 801	152	52 080 801	9	52 080 801	135	0.00%	0.00%
17	51 881 836	240	51 881 836	6	51 881 836	139	0.00%	0.00%
18	51 883 836	129	51 883 836	5	51 883 836	133	0.00%	0.00%
19	52 003 836	124	52 003 836	7	52 003 836	135	0.00%	0.00%
20	52 128 836	175	52 128 836	7	52 128 836	141	0.00%	0.00%
Average		164.2		7.1		135.2	0.00%	0.00%

tables that the time needed by HGA and CPLEX grow slower than the Lingo global solve.

In terms of solution qualities, we observe from Tables 12 and 13 that both HGA and CPLEX obtained the optimal solutions for all test cases in problem types 1 and 2. Therefore, in terms of solution qualities, both HGA and CPLEX performed well for problem types 1 and 2.

For problem types 3 to 6, since we failed to produce optimal solutions after more than 10 CPU hours, the feasible solution obtained by HGA was compared to those found by CPLEX solver. The percentage gap, defined as 100 (CPLEX solution -HGA solution)/(CPLEX solution) percentage points, is used to evaluate the solution quality of the HGA. In addition, the time gap (defined as the CPU time used by HGA – the CPU time used by CPLEX), is used to evaluate the efficiency of the proposed heuristic.

For problem types 3 and 4, we observe from Tables 14 an 15 that except for problem instance 1 of problem type 3, the feasible solutions found by HGA are all better than or are in par with the feasible solutions found by CPLEX solver. In addition, the average CPU time used in finding feasible solutions by HGA was much slower compared to the average times used by CPLEX solver. Therefore, from Tables 14 and 15, we also found that CPLEX outperformed HGA with respect to computational time in most cases.

TABLE 14. Computational result for problem type 3.

	GAMS		HGA		Gap	
No.	Sol	Time	Sol	Time	Sol	Time
1	104 671 617	435	104 684 093	739	−0.01%	304
2	111 928 992	213	111 835 409	1655	0.08%	1442
3	119 188 263	358	119 169 080	1180	0.02%	822
4	133 698 819	193	133 698 819	723	0.00%	530
5	114 462 556	350	114 384 650	875	0.07%	525
6	124 253 495	341	124 249 044	619	0.00%	278
7	134 117 822	896	134 038 906	578	0.06%	–317
8	106 149 061	919	106 113 681	635	0.03%	–284
9	106 691 426	294	106 641 412	1070	0.05%	776
10	107 578 412	669	107 558 044	584	0.02%	–85
11	105 094 781	218	105 094 781	1017	0.00%	799
12	105 267 264	378	105 264 966	610	0.00%	233
13	105 522 542	395	105 444 695	895	0.07%	500
14	104 884 734	579	104 873 234	730	0.01%	151
15	105 007 191	470	104 975 191	655	0.03%	184
16	105 096 189	359	105 077 149	723	0.02%	364
17	104 672 319	145	104 672 319	621	0.00%	476
18	104 674 319	26	104 674 319	654	0.00%	627
19	104 927 657	400	104 919 319	626	0.01%	226
20	105 172 285	290	105 169 319	629	0.00%	339
Average		396		791	0.02%	394

For problem types 5 and 6, we found that CPLEX failed to produce feasible solutions after more than 10 h for problem types 4 and 5 while the proposed HGA was still successful in generating feasible solution for all test problems within reasonable CPU times. In order to test the performance of the proposed HGA on a problem size which CPLEX solver can not solve, we compared the total cost obtained by HGA, with associated lower bound to evaluate the solution quality of the proposed HGA. The lower bound was obtained by using Lingo solver to solve a relaxed $\mathbf{P}_1$ model in which the integer constraints on variable Q^F_{jt}, O_{mt}, Y^d_{cjt} and Y^F_{dmt} are relaxed. The solution and computational time are shown in Tables 15 and 16, respectively. In these tables,the HL-gap denotes the percentage gap between the feasible solution obtained by HGA and the lower bound obtained in each instance of problem $\mathbf{P}_1$. We found from Tables 15 and 16 that the proposed HGA produced near-optimal solutions with deviations of less than 0.1% from the lower bound solutions. Accordingly, from the above computational results, the proposed HGA can be used as a local search heuristic to solve the dynamic facility logistics problems.

Summary of the computational results:

1. In problem types 1 and 2, from Tables 12 and 13, we found that both HGA and CPLEX solver can produce global costs for all problem instances. In terms of computational time, CPLEX solver is superior to HGA.

TABLE 15. Computational result for problem type 4.

	GAMS		HGA		Gap	
No.	Sol	Time	Sol	Time	Sol	Time
1	222 171 590	1379	221 926 041	1723	0.11%	344
2	236 994 844	1088	236 948 180	1876	0.02%	787
3	251 883 715	1003	251 883 715	1440	0.00%	437
4	281 666 335	1586	281 616 563	1653	0.02%	67
5	242 992 651	1977	242 958 584	1283	0.01%	–695
6	263 884 049	1518	263 801 948	1903	0.03%	385
7	284 775 441	1593	284 775 441	1327	0.00%	–266
8	225 238 432	1501	225 129 201	1509	0.05%	8
9	226 493 366	1462	226 433 723	1779	0.03%	317
10	228 375 769	840	228 375 769	1627	0.00%	787
11	222 751 105	1112	222 651 239	2025	0.04%	913
12	223 011 109	2154	222 987 770	1480	0.01%	–674
13	223 401 115	2017	223 401 115	1874	0.00%	–143
14	222 395 968	1591	222 287 697	1872	0.05%	281
15	222 543 404	1158	222 463 411	2297	0.04%	1139
16	222 690 841	1692	222 690 841	1377	0.00%	–315
17	222 104 095	986	222 101 507	1794	0.00%	808
18	222 106 095	521	222 106 095	1472	0.00%	951
19	222 401 095	539	222 401 095	1225	0.00%	686
20	222 701 095	971	222 606 639	1970	0.04%	998
Average		1334		1675	0.02%	341

2. In problem types 3 and 4, comparing the HGA's feasible solutions with CPLEX's feasible solutions, we found that HGA is better than CPLEX solver in most problem instances in terms of solution quality. In terms of computational time, CPLEX solver is still superior to HGA.
3. The CPLEX solver can not produce feasible solutions over 10 h. However, the proposed HGA can still produce feasible solutions. We compare HGA feasible solutions with associated lower bounds and found that HGA's feasible solutions are very close to low-bound costs.
4. As for the computational efficiency, Tables 12–17 illustrate that the computational time used by the HGA method range from 27.1 s to about 3.41 h as the problem size increases, and the growth rate is moderate. This implies that the proposed HGA algorithm is scalable against problem complexity.

5. Conclusion

Many firms have begun to incorporate reverse logistics into their supply chain systems since they are not only able to fill customers' demands but also improve corporate images. In terms of product flows, integrated supply chains consist of both forward flows and reverse flows. In forward flows, production and material

TABLE 16. Computational result for problem type 5.

	LB	HGA		Gap
No.	Sol	Sol	Time	Sol
1	597 398 901	597 455 968	2137	0.01%
2	634 461 656	634 608 100	2587	0.02%
3	671 524 413	671 546 786	1562	0.00%
4	745 649 924	745 672 298	1808	0.00%
5	653 776 350	653 798 596	1841	0.00%
6	710 153 962	710 176 100	2157	0.00%
7	766 531 905	766 553 953	1904	0.00%
8	606 219 833	606 242 206	1650	0.00%
9	609 748 206	609 770 579	1795	0.00%
10	615 040 766	615 063 138	1861	0.00%
11	598 440 786	598 463 159	1587	0.00%
12	598 857 541	598 879 914	1638	0.00%
13	599 482 672	599 505 045	1747	0.00%
14	597 830 375	597 852 748	1950	0.00%
15	598 046 112	598 068 485	1826	0.00%
16	598 261 849	598 284 222	1803	0.00%
17	597 404 158	597 424 274	1826	0.00%
18	597 407 639	597 426 274	1686	0.00%
19	597 718 901	597 741 274	1744	0.00%
20	598 038 900	598 061 274	1731	0.00%
Average			1842	0.00%

purchasing schedules have great effects on the performance of production management since unsuitable schedules usually lead to high inventory cost. In reverse flows with dynamic facilities, the timing for opening/closing of facilities and the amount of shipping between facilities have an important influence on firms' efficiency since poorer shipping strategies result in high warehousing and transportation costs.

In this study, a supply chain problem with a forward and reverse logistics network was formulated and an efficient heuristic was developed. The problem studied here is formulated as an integer linear model with multi-periods, a dynamic reverse logistics system, multi-commodities and capacitated facilities. The model introduces the possibility of opening and closing a facility more than once during the planning horizon. It also considers explicitly not only leasing and operating costs but also start-up costs incurred when reopening facilities. Since such a problem belongs to a class of NP hard problems, a hybrid genetic heuristic approach was developed in order to solve a realistically sized problem. We report computational results which show the gaps between the feasible solutions that we proposed and the lower bounds of the optimal solution and exact solutions. The values of the gaps and the computational times shown in Tables 12 to 17 indicate that the proposed HGA is capable of producing optimal solutions for small size problems, solving large size problems and producing good feasible solutions which are very

TABLE 17. Computational result for problem type 6.

No.	LB Sol	HGA Sol	HGA Time	Gap Sol
1	2 013 961 021	2 014 047 261	9423	0.00%
2	2 129 865 271	2 129 976 761	10 210	0.01%
3	2 245 769 521	2 245 804 419	13 664	0.00%
4	2 477 578 021	2 477 742 042	15 506	0.01%
5	2 202 327 162	2 202 449 675	14 978	0.01%
6	2 390 723 904	2 391 343 585	11 534	0.03%
7	2 579 056 135	2 579 766 166	12 169	0.03%
8	2 041 563 000	2 041 599 594	8828	0.00%
9	2 052 604 470	2 052 908 151	12 343	0.01%
10	2 069 166 675	2 069 985 534	11 067	0.04%
11	2 016 059 175	2 016 618 310	11 300	0.03%
12	2 016 899 115	2 016 982 918	18 107	0.00%
13	2 018 159 025	2 019 295 750	11 457	0.06%
14	2 014 768 000	2 016 357 392	10 081	0.08%
15	2 015 172 338	2 015 975 878	11 098	0.04%
16	2 015 604 988	2 015 613 269	12 940	0.00%
17	2 013 964 677	2 014 033 053	11 960	0.00%
18	2 013 968 244	2 014 018 972	8532	0.00%
19	2 014 459 325	2 015 860 214	19 301	0.07%
20	2 014 959 325	2 014 995 919	11 148	0.00%
Average			12 282	0.02%

close to low-bound costs for large size problems that CPLEX solver fails to produce feasible solutions.

Generally, a facility may be dedicated or flexible. In the first case, the facility can only supply its local market while the later can supply not only its local market but other markets as well. In this study, we consider a case in which a collection-center is the dedicated facility of an assembly center. However, the model also can be transferred to cases in which a collection-center is a flexible facility. This can be done by eliminating the variable and the constraints of equations (2.19) to (2.22) and equations (2.25) to (2.28), and increasing the following constraints: $\sum_{d=1}^{D} Y_{cjt}^{d} \leq x_t^c B$, $\sum_{d=1}^{D} Y_{cjt}^{d} \geq x_t^c$ for all c, j, t and $\sum_{c=1}^{C} Y_{cjt}^{d} \leq s_t^d B$, $\sum_{c=1}^{C} Y_{cjt}^{d} \geq s_t^d$ for all d, j, t.

REFERENCES

[1] N. Brahimi, S. Dauzere-Peres, N.M. Najid and A. Nordli, Single item lot sizing problems. *Eur. J. Oper. Res.* **168** (2006) 1–16.

[2] C. Canel, B.M. Khumawala, J. Law and A. Loh, An algorithm for the capacitated, multi-commodity multi-period location problem. *Comput. Oper. Res.* **28** (2001) 411–427.

[3] K.W. Chau, A two-stage dynamic model on allocation of construction facilities with genetic algorithm. *Automat. Constr.* **13** (2004) 481–490.

[4] J. Dias, M.E. Captivo and J. Climaco, Efficient primal-dual heuristic for a dynamic location problem. *Comput. Oper. Res.* **34** (2007) 1800–1823.

[5] S.D. Ekşioğlu, *Optimizing integrated production, inventory and distibution problems in supply chains.* Ph. D. thesis, University of Florida, USA (2002) (fcla.edu/fcla/etd/UFE0000529).

[6] M. Fleischmann, P. Beullens and J.M. Bloemhof-Ruwaard, The impact of product recovery on logistics network design. *Prod. Oper. Manage.* **10** (2001) 156–173.

[7] Y. Hinojosa, J. Kalcsics, S. Nickel, J. Puerto and S. Velten, Dynamic supply chain design with inventory. *Comput. Oper. Res.* **35** (2008) 373–391.

[8] R. Jans and Z. Degraeve, Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. *Eur. J. Oper. Res.* **177** (2007) 1855–1875.

[9] V. Jayaraman, R.A. Patterson and E. Rolland, The design of reverse distribution network: Model and solution procedures. *Eur. J. Oper. Res.* **150** (2003) 128–149.

[10] J. Krarup and P.M. Pruzan, The simple plant location problem: survey and synthesis. *Eur. J. Oper. Res.* **12** (1983) 36–81.

[11] H.J. Ko and G.W. Evans, A genetic algorithm-based heuristic for dynamic integrated forward/reverse logistics network for 3PLs. *Comput. Oper. Res.* **34** (2007) 346–366.

[12] H.J. Ko, C.S. Ko and T. Kim, A hybrid optimization/simulation approach for a distribution network design of 3PLS. *Comput. Ind. Eng.* **50** (2006) 440–449.

[13] J. Krarup and P.M. Pruzan, The simple plant location problem: Survey and synthesis. *Eur. J. Oper. Res.* **12** (1983) 36–57.

[14] R.D. Kusumastuti, R. Piplani and G.H. Lim, Redesigning closed-loop service network at a computer manufacturer: A case study. *Int. J. Prod. Econ.* **111** (2008) 244–260.

[15] I.M. Langella, Heuristics for demand-driven disassembly planning. *Comput. Oper. Res.* **34** (2007) 552–577.

[16] H. Lee and M. Dong, A heuristic approach to logistics network design for end-of-lease computer products recovery. *Transport. Res. E.* **44** (2008) 455–474.

[17] K. Lieckens and N. Vandaele, Reverse logistics network design with stochastic lead times. *Comput. Oper. Res.* **34** (2007) 395–416.

[18] Z. Lu and N. Bostel, A facility location model for logistics systems including reverse flows: The case of remanufacturing activities. *Comput. Oper. Res.* **34** (2007) 299–323.

[19] E. Melachrinoudis, H. Min and X. Wu, A multiobjective model for the dynamic location of landfills. *Location Science* **3** (1995) 143–166.

[20] M.T. Melo, S. Nickel and F.S. Gama, Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning. *Comput. Oper. Res.* **33** (2005) 181–208.

[21] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs.* 3rd edition, Springer-Verlag, London, UK (1996).

[22] H. Min and H.J. Ko, The dynamic design of a reverse logistics network from the perspective of third-party logistics service providers. *Int. J. Prod. Econ.* **113** (2008) 176–192.

[23] H. Min, C.S. Ko and H.J. Ko, The spatial and temporal consolidation of returned products in a closed-loop supply chain network. *Comput. Ind. Eng.* **51** (2006) 309–320.

[24] G.C. Onwubolu and B.V. Babu, *New optimization techniques in engineering.* Springer-Verlag, Berlin, Heidelberg (2004) Chap. 2.

[25] M.S. Pishvaee, R.Z. Farahani and W. Dullaert, A memetical gorithm for bi-objective integrated forward/reverse logistics network design. *Comput. Oper. Res.* **37** (2005) 181–208.

[26] C. Prahinski and C. Kocabasoglu, Empirical research opportunities in reverse supply chains. *Omega-Int. J. Manage. Sci.* **34** (2006) 519–532.

[27] M.I.G. Salema, A.P. Barbosa-Povoa and A.Q. Novais, An optimization model for the design of a capacitated multi-product reverse logistics network with uncertainty. *Eur. J. Oper. Res.* **179** (2007) 1063–1077.

[28] E.A. Silver, D.F. Pyke and R. Peterson, *Inventory management and production planning and scheduling*. 3rd edition, John Wiley & Sons, USA (1998).

[29] M. Sodhi and B. Reimer, Models for recycling end-of-life products. *OR-Spectrum* **23** (2001) 97–115.

[30] T. Spengler, H. Püchert, T. Penkuhn and O. Rentz, Environmental integrated production and recycling management. *Eur. J. Oper. Res.* **97** (1997) 308–326.

[31] Z. Yongsheng and W. Shouyang, Generic Model of Reverse Logistics Network Design. *J. Transport. Syst. Eng. Inf. Tech.* **8** (2008) 71–78.

[32] `http://www.lindo.com/index.php`