TRANSPORTATION SCIENCE
Vol. 40, No. 3, August 2006, pp. 287–299
ISSN 0041-1655 | EISSN 1526-5447 | 06 | 4003 | 0287

informs®

DOI 10.1287/trsc.1050.0130

Inbound Logistic Planning: Minimizing Transportation and Inventory Cost

Oded Berman
Joseph L. Rotman School of Management, University of Toronto, 105 St. George Street, Toronto, Ontario M5S 3E6, Canada, berman@rotman.utoronto.ca

Qian Wang
Management School, Graduate University of China Academy of Sciences, 80 Zhongguancunrd, Beijing 10, 00080 P. R. China, wangqian@gucas.ac.cn

In today's competitive environment, supply chain management is a major concern for a company. Two of the key issues in supply chain management are transportation and inventory management. To achieve significant savings, companies should integrate these two issues instead of treating them separately. This paper considers the problem of selecting the appropriate distribution strategy for delivering a family of products from a set of suppliers to a set of plants so that the total transportation, pipeline inventory, and plant inventory costs are minimized. With reasonable assumptions, a simple model is presented to provide a good solution that can serve as a guideline for the design and implementation of the distribution network. Due to the plant inventory cost, the problem is formulated as a nonlinear integer programming problem. The problem is difficult to solve because the objective function is highly nonlinear and neither convex nor concave. A greedy heuristic is proposed to find an initial solution and an upper bound. A heuristic and a branch-and-bound algorithm are developed based on the Lagrangian relaxation of the nonlinear program. Computational experiments are performed, and based on the results we can conclude that the performance of the algorithms are promising.

Key words: supply chain management; distribution strategies; inventory
History: Received: December 2004; revision received: June 2005; accepted: July 2005.

1. Introduction

In a typical supply chain, there are sets of suppliers and plants. Products (e.g., raw materials or parts) are shipped from suppliers to plants to be further processed. For example, consider an automobile company. The car is assembled at one of the company's assembling plants. The assembling plant does not produce all the parts that are required for the assembly. It typically procures the required parts from a set of suppliers, such as an engine supplier, a tire supplier, etc. A supplier produces one or several types of parts. It may satisfy the demands of one or several assembling plants. The suppliers and the assembling plants form a two-level supply network.

For many companies, the products are shipped from a supplier to a plant by trucks. There are several distribution strategies for truck delivery:

• Direct: Trucks travel directly from a supplier to a plant, without any stop.

• Milk-run (peddling): Trucks pick up products at one or several suppliers and deliver them to one or several plants.

• Cross-dock: Products are delivered from suppliers to a cross-dock, and then from the cross-dock to plants.

The three distribution strategies are illustrated in Figure 1.

Different distribution strategies have different transportation cost and time. For example, direct delivery has the shortest distance, and therefore the lowest transportation cost and the shortest delivery time. A delivery through a cross-dock has the longest distance, and therefore the highest transportation cost and the longest delivery time. When each truck is fully or almost fully loaded (which is appropriate for the case when the amount to be shipped is fairly large), because direct delivery can only consolidate the products from the same supplier to the same plant, low delivery frequency and high plant inventory are incurred. Meanwhile, cross-dock can combine products from different suppliers, which leads to high delivery frequency and low plant inventory. The relationship between distribution strategy and inventory cost is listed in Table 1.

In this paper, we consider the problem of selecting the appropriate distribution strategy for delivering a family of products from a set of suppliers to a set of plants so that the total transportation, pipeline inventory, and plant inventory costs are minimized.

Models that attempt to combine inventory and transportation cost are not new in the literature. Inter-

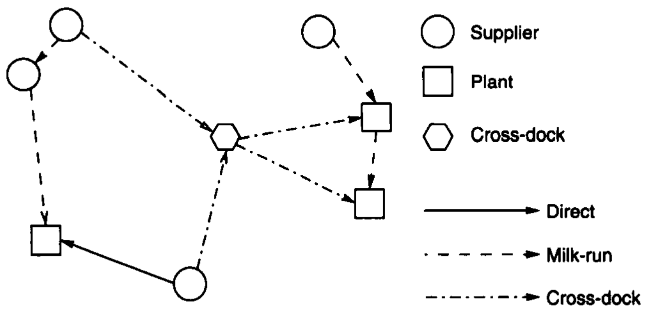

Figure 1 Distribution Strategy

ested readers are referred to Bertazzi and Speranza (1999) and Baita et al. (1998) for a survey on the models and algorithms for the minimization of inventory and transportation costs. The one that is the most closely related to our problem is presented in Blumenfeld et al. (1987). The problem studied in Blumenfeld et al. (1987) is how to ship products from suppliers to plants for a General Motors division, Delco Electronics Division, so that the total of transportation and inventory costs are minimized. However, the paper does not present any analytical model.

Most of the papers that deal with analytical models on inventory and transportation management can be classified according to the source and destination of the distributed products. They include the following.

(1) *Single source to single destination.* Speranza and Ukovich (1994) dealt with the problem of minimizing the sum of transportation and inventory costs for shipping several products on a common link where the shipping frequency is selected from a finite set of potential values. Zhao et al. (2004) addressed the problem of determining the optimal ordering quantity and frequency for a supplier-retailer logistic system in which the transportation cost as well as the multiple uses of vehicles are considered.

(2) *Single source to multiple destinations.* Burns et al. (1985) studied the problem of minimizing total inventory and transportation costs from a supplier to multiple customers. They derived formulas for the two costs and determined the optimal trade-off between these costs. Campbell (1993) presented and optimized an approximate analytical model of distribution from a single origin to many destinations via transshipment terminals. Cetinkaya and Lee (2000) presented an analytical model for coordinating inventory and transportation decisions in a vendor-managed inventory system. They developed a renewal-theoretical model for the case of Poisson demand to compute the optimal replenishment quantity and dispatch frequency simultaneously. Chan et al. (2002) presented a model to design simple inventory policies and transportation strategies to satisfy time-varying demands over a finite horizon, while minimizing systemwide cost by taking advantage of quantity discounts in the transportation cost structures.

(3) *Multiple sources to single destination.* Popken (1994) presented a model to consolidate inbound freight at transshipment points from multiple sources to a single destination to minimize the overall transportation and inventory costs.

The papers that involve distributing products from multiple sources to multiple destinations focus mostly on the issues of distribution network design with or without transshipment centers (hubs). A transshipment center is used to increase the efficiency of delivery systems when economies of scale are taken into account, i.e., the marginal cost decreases with flow volume. For example, Campbell (1996) and O'Kelly and Bryan (1998) studied the problem of finding locations for the hubs and assigning nodes to them. These models focused on the hub-and-spoke system only. There are also some papers that considered the mixed delivery systems. Aykin (1995) studied the problem of simultaneously determining the hub locations and delivery mode for each demand. Liu, Li, and Chan (2003) studied a mixed truck delivery system that allows both hub-and-spoke and direct shipment delivery modes. The computational experiment results showed that the mixed system can save around 10% of total traveling distance on average as compared with either of the two pure systems. Lapierre, Ruiz, and Soriano (2004) presented a new model and an efficient metaheuristic that determines the number and the location of hubs, as well as the best distribution strategies on each segment accounting for both weight and volume metrics. However, these models do not take into account the inventory cost.

Blumenfeld et al. (1985) studied the problem of determining optimal shipping strategies on a freight network with a single consolidation terminal by analyzing trade-offs between transportation, inventory, and production set-up costs. A decomposition method was presented to solve problems with few origins and shipment sizes. Bookbinder and Fox (1998) considered the problem of finding the optimal routings for intermodal containerized transport from five Canadian origins to three major Mexican destinations. Nondominated time/cost trade-offs are identified for each origin-destination pair. The plant inventories were not considered in their model.

Table 1 Distribution Strategies and Inventory Costs for a Full Truck

Distribution strategy	Delivery time	Delivery frequency	Inventory cost	
			Pipeline	Plant
Direct	short	low	low	high
Milk-run	medium	medium	medium	medium
Cross-dock	long	high	high	low

Shen, Coullard, and Daskin (2003) proposed a joint inventory-location model that involves a single supplier and multiple retailers. Its goal is to determine which retailers should serve as distribution centers and how to allocate the other retailers to the distribution center. Snyder, Daskin, and Teo (2006) extended the model to the situation with random parameters described by discrete scenarios.

The problem of selecting the distribution strategy studied in this paper is very complicated if all the details must be captured. As mentioned in Daganzo (1999), often a two-step solution approach is suitable for logistic problems. The first (analytical) step involves few details and uses approximation to get broad solution concepts; the second (fine-tuning) step generates specific solutions based on the solution obtained in the first step by taking into account all the relevant detailed information which are ignored in the first step. We focus on the first analytical step. To have a solvable problem that captures the essence of the cost, we make assumptions about the product quantities and frequencies.

ASSUMPTION 1. *Product quantities are infinitely splittable, i.e., a product can be shipped in any quantity within a vehicle shipment.*

ASSUMPTION 2. *Delivery frequency can be any positive number and is not limited to a set of potential numbers.*

We also ignore some operational details by assuming that

ASSUMPTION 3. *Products are always available for shipping at suppliers, no matter which distribution strategy is chosen.*

ASSUMPTION 4. *Inbound-outbound coordination at the cross-dock is ignored.*

ASSUMPTION 5. *All units of the same flow (a flow is a combination of supplier, plant, and product) are assigned to the same transportation option, i.e., direct or through the same cross-dock.*

ASSUMPTION 6. *Each truck is fully loaded. Only the volume of products is concerned when calculating truck capacity usage. The transportation costs are only determined by the source and destination, regardless of the weight.*

The solution of the simple model based on Assumptions 1–4 may not be directly implementable. For example, products may not be available all the time due to the capacity limit of the suppliers, or the interval between two shipments to the plants through cross-dock may not be constant due to the coordination at the cross-dock. However, the solution of the simple model can serve as a guideline for the implementation of distribution and an approximation of true costs. Management can get an implementable solution by fine-tuning it.

We note that Assumption 5 is practical due to the simplicity of its implementation. Although full-truckload delivery policy may give an infinitely large error with respect to the optimum, as pointed out in Bertazzi and Speranza (2005), Assumption 6 is reasonable if the quantity of product shipped is not too low and the ratio between the inventory-carrying cost and the volume of the products is not too high.

In this paper, we focus on the special case that satisfies the following conditions:

(1) The demand rate for each product at each plant from each supplier is constant.

(2) Only direct and cross-dock distribution strategies are considered.

(3) Only one truck type is available.

We do not consider peddling explicitly in this paper because peddling can be done in a preprocessing procedure, as mentioned in Blumenfeld et al. (1987). We can group together suppliers (plants) in close proximity to form peddling regions by visual observation. Each group of suppliers (plants) is treated as a single supplier (plant) and the flows that originate from (or are destined to) the same group of suppliers (plants) are combined as well. The transportation costs are approximated by the cost to the closest supplier (plant) in the group plus the cost of traveling between suppliers (plants) in the region.

The rest of this paper is organized as follows. Section 2 presents a nonlinear integer mathematical formulation for our model. Section 3 proposes a greedy heuristic to find an initial solution. A Lagrangian relaxation heuristic and a branch-and-bound algorithm are developed in §4. Computational experiments and a sensitivity analysis are reported in §5. Section 6 concludes the paper with a summary and directions for future work.

2. Formulation

The notation used in this paper are listed in Table 2, where we use a general "period" for measuring quantities (period can either be a week, a month, or any other time unit that management prefers).

Note that t_{ij}^d, t_{ik}^i, and t_{kj}^o are the ratios of transportation time to the length of a period. For example, if a period is a week and t_{11}^d is three days, then $t_{11}^d = 3/7$.

Note also that the number of products can be reduced dramatically if we group similar products together. For example, consider a paint supplier for the automobile assembly plants. Although the number of colors for a specific type of paint may be in the tens or in the hundreds, we can group this type of paint as one product because they have the same price and occupy the same truck capacity. By doing

Table 2 Notation

I	set of suppliers
J	set of plants
P	set of products
K	set of cross-docks
C	truck capacity
c^d_{ij}	direct transportation cost of shipping one truckload of products from supplier i to plant j
c^i_{ik}	inbound transportation cost of shipping one truckload of products from supplier i to cross-dock k
c^o_{kj}	outbound transportation cost of shipping one truckload of products from cross-dock k to plant j
t^d_{ij}	direct transportation time (periods) from supplier i to plant j
t^i_{ik}	inbound transportation time (periods) from supplier i to cross-dock k
t^o_{kj}	outbound transportation time (periods) from cross-dock k to plant j
T_k	time spent in transferring from inbound to outbound at cross-dock k
b_p	truck capacity occupied by one unit of product p
h_p	inventory-carrying cost of one unit of product p per period
d_{ijp}	demand of product p required from supplier i by plant j per period
$F = \{(i,j,p): i \in I,\ j \in J,\ p \in P,\ d_{ijp} > 0\}$, set of flows	
$F_j = \{(i,j,p): i \in I,\ p \in P,\ d_{ijp} > 0\}$, set of flows to plant j	
$F_{ij} = \{(i,j,p): p \in P,\ d_{ijp} > 0\}$, set of flows from supplier i to plant j	
$P_{ij} = \{p: (i,j,p) \in F_{ij}\}$	
$IP_j = \{(i,p): (i,j,p) \in F_j\}$	

so, we can reduce the number of flows even more dramatically.

To obtain a compact mathematical formulation, we introduce a dummy cross-dock 0, where $0 \notin K$. A flow assigned to cross-dock 0 means that it is actually shipped directly. Let $K^0 = K \cup \{0\}$. With the introduction of decision variables

$$x_{ijpk} = \begin{cases} 1 & \text{if flow } (i,j,p) \text{ is shipped through cross-dock } k, \\ 0 & \text{otherwise,} \end{cases}$$

(note that $x_{ijp0} = 1$ means that flow (i,j,p) is shipped directly), the mathematical formulation of the problem can be stated as follows:

$$(P) \quad \min \sum_{k \in K^0} g_k(X) \tag{1}$$

subject to

$$\sum_{k \in K^0} x_{ijpk} = 1 \quad \forall (i,j,p) \in F, \tag{2}$$

$$x_{ijpk} \in \{0,1\} \quad \forall (i,j,p) \in F,\ k \in K^0, \tag{3}$$

where X is the vector of all the decision variables x_{ijpk}, and $g_k(X)$ is the total of transportation, pipeline inventory, and plant inventory costs to ship flows through cross-dock k. Constraints (2) ensure that every flow is delivered and constraints (3) guarantee that the same flow follows the same route.

Next, we present the functions $g_k(X), k \in K^0$. In the remainder of this paper, we assume that $0/0 \equiv 0$.

2.1. $g_0(X)$

Function $g_0(X)$ is the total cost of direct delivery. For each supplier-plant pair (i,j), the frequency of shipment is $f^d_{ij} = \sum_{p\in P_{ij}} b_p d_{ijp} x_{ijp0}/C$, the transportation cost is $f^d_{ij} c^d_{ij}$, the pipeline inventory cost is $\sum_{p \in P_{ij}} t^d_{ij} h_p \cdot d_{ijp} x_{ijp0}$, and the plant inventory cost is $\sum_{p\in P_{ij}} h_p d_{ijp} x_{ijp0}/(2 f^d_{ij})$. Hence, we have

$$\begin{aligned} g_0(X) &= \sum_{i\in I}\sum_{j \in J}\left[f^d_{ij} c^d_{ij} + \sum_{p\in P_{ij}}\left(t^d_{ij} h_p d_{ijp} x_{ijp0} + \frac{h_p d_{ijp} x_{ijp0}}{2 f^d_{ij}}\right)\right] \\ &= \sum_{i\in I}\sum_{j\in J}\left[\sum_{p \in P_{ij}}\left(\frac{b_p d_{ijp} c^d_{ij}}{C} x_{ijp0} + t^d_{ij} h_p d_{ijp} x_{ijp0}\right) + \frac{\sum_{p\in P_{ij}} h_p d_{ijp} x_{ijp0}}{2\sum_{p\in P_{ij}} b_p d_{ijp} x_{ijp0}/C}\right] \\ &= \sum_{i\in I}\sum_{j\in J}\left[\sum_{p\in P_{ij}}\left(\frac{b_p c^d_{ij}}{C} + t^d_{ij} h_p\right) d_{ijp} x_{ijp0} + \frac{\sum_{p\in P_{ij}} h_p d_{ijp} x_{ijp0}}{\sum_{p\in P_{ij}} 2 b_p d_{ijp} x_{ijp0}/C}\right]. \end{aligned} \tag{4}$$

Let $\bar{b}_p = 2b_p/C$ and $\bar{c}_{ijp0} = b_p c^d_{ij}/C + t^d_{ij} h_p$. Then,

$$g_0(X) = \sum_{i \in I}\sum_{j\in J}\left[\sum_{p \in P_{ij}} \bar{c}_{ijp0} d_{ijp} x_{ijp0} + \frac{\sum_{p\in P_{ij}} h_p d_{ijp} x_{ijp0}}{\sum_{p \in P_{ij}} \bar{b}_p d_{ijp} x_{ijp0}}\right]. \tag{5}$$

2.2. $g_k(X), k \in K$

For $k \in K$, function $g_k(X)$ is the cost of shipping flows that travel through cross-dock k. The transportation cost consists of two parts: the inbound transportation cost and the outbound transportation cost. For each $i \in I$, the frequency of inbound shipment is $f^i_{ik} = \sum_{j\in J}\sum_{p\in P} b_p d_{ijp} x_{ijpk}/C$ and the total inbound transportation cost is $\sum_{i\in I} f^i_{ik} c^i_{ik}$. For each $j \in J$, the frequency of outbound shipment is $f^o_{kj} = \sum_{i \in I}\sum_{p\in P} b_p d_{ijp} x_{ijpk}/C$ and the total outbound transportation cost is $\sum_{j\in J} f^o_{kj} c^o_{kj}$.

Because the transportation time for shipping flow (i,j,p) through cross-dock k is $(t^i_{ik} + T_k + t^o_{kj})$, the pipeline inventory cost is $(t^i_{ik} + T_k + t^o_{kj}) h_p d_{ijp} x_{ijpk}$ and the plant inventory cost is $h_p d_{ijp} x_{ijpk}/(2 f^o_{kj})$. Hence, we have

$$\begin{aligned} g_k(X) &= \sum_{i\in I} f^i_{ik} c^i_{ik} + \sum_{j\in J} f^o_{kj} c^o_{kj} + \sum_{j \in J}\sum_{(i,p)\in IP_j}\left[(t^i_{ik} + T_k + t^o_{kj}) \cdot h_p d_{ijp} x_{ijpk} + h_p d_{ijp} x_{ijpk}/(2 f^o_{kj})\right] \\ &= \sum_{j\in J}\left[\sum_{(i,p)\in IP_j}\left(\frac{b_p d_{ijp} c^i_{ik}}{C} x_{ijpk} + \frac{b_p d_{ijp} c^o_{kj}}{C} x_{ijpk} + (t^i_{ik} + T_k + t^o_{kj}) h_p d_{ijp} x_{ijpk}\right) + \frac{\sum_{(i,p)\in IP_j} h_p d_{ijp} x_{ijpk}}{2\sum_{(i,p)\in IP_j} b_p d_{ijp} x_{ijpk}/C}\right] \end{aligned}$$

$$= \sum_{j\in J}\left[\sum_{(i,p)\in IP_j}\left(\frac{b_p c^i_{ik}}{C}+\frac{b_p c^o_{kj}}{C}+(t^i_{ik}+T_k+t^o_{kj})h_p\right)\cdot d_{ijp}x_{ijpk}+\frac{\sum_{(i,p)\in IP_j} h_p d_{ijp}x_{ijpk}}{2\sum_{(i,p)\in IP_j} b_p d_{ijp}x_{ijpk}/C}\right]. \quad (6)$$

Let $\bar{c}_{ijpk} = (c^i_{ik}+c^o_{kj})b_p/C+(t^i_{ik}+T_k+t^o_{kj})h_p$ then

$$g_k(X) = \sum_{j\in J}\left[\sum_{(i,p)\in IP_j}\bar{c}_{ijpk}d_{ijp}x_{ijpk}+\frac{\sum_{(i,p)\in IP_j} h_p d_{ijp}x_{ijpk}}{\sum_{(i,p)\in IP_j}\bar{b}_p d_{ijp}x_{ijpk}}\right]. \quad (7)$$

Obviously, $\forall k \in K^0$, $g_k(X)$ is neither convex nor concave, and therefore our problem is a nonlinear integer program. We tried to use commercial nonlinear solvers CONOPT, DICOPT, and MINOS to solve a small instance of problem (P) with 3 suppliers, 2 plants, 2 products, 10 flows, and 2 cross-docks. Although all the solvers terminated very quickly, they only presented us with a local optimal solution. Furthermore, they did not provide any information about the solution quality, i.e., we have no idea of the gap between the solution obtained and the global optimal solution. Therefore, we need to develop a solution procedure for this problem. In the next two sections, we will present two heuristics and a branch-and-bound (BB) algorithm to solve problem (P). The first heuristic is to fix variables following a greedy criterion and the second heuristic is based on the Lagrangian relaxation (LR) of problem (P). The BB algorithm uses the LR to get a lower bound for each subproblem.

3. Greedy Heuristic

In this section, we propose a greedy heuristic to find an initial feasible solution and an upper bound for problem (P).

Let $F^k_j \subset F_j$ be the set of flows that may go through cross-dock k and $F^d_{ij} \subset F_{ij}$ be the set of flows that are fixed to be shipped directly. Denote $P^d_{ij} = \{p\colon (i,j,p) \in F^d_{ij}\}$ and $IP^k_j = \{(i,p)\colon (i,j,p) \in F^k_j\}$.

From Equations (5) and (7), we can see that for each flow $(i,j,p) \in F$, the transportation and pipeline inventory costs are independent of the distribution strategies for other flows, while the plant inventory cost is dependent on the distribution strategies for other flows. For each unfixed flow $(i,j,p) \in F$, the transportation and pipeline inventory costs through cross-dock $k \in K$ are $\bar{c}_{ijpk}d_{ijp}$, no matter if there is any other flow going through cross-dock k. However, the plant inventory cost is a function of all flows going through cross-dock k. The more flows are shipped through cross-dock k, the less is the plant inventory cost for flow (i,j,p). Therefore, the minimum possible plant inventory cost of flow (i,j,p) through cross-dock k is $h_p d_{ijp}/(\sum_{(i',p')\in IP^k_j}\bar{b}_{p'}d_{i'jp'})$. Similarly, $\bar{c}_{ijp0}d_{ijp}$ is the transportation and pipeline inventory costs to ship flow (i,j,p) directly, and $h_p d_{ijp}/(\bar{b}_p d_{ijp}+\sum_{p'\in P^d_{ij}}\bar{b}_{p'}d_{ijp'})$ is the maximum possible plant inventory cost to ship flow (i,j,p) directly. Let

$$\alpha_{ijpk} = \bar{c}_{ijpk}d_{ijp} - \bar{c}_{ijp0}d_{ijp}, \quad (8)$$

$$\delta_{ijpk} = \left[\bar{c}_{ijpk}d_{ijp}+\frac{h_p d_{ijp}}{\sum_{(i',p')\in IP^k_j}\bar{b}_{p'}d_{i'jp'}}\right] - \left[\bar{c}_{ijp0}d_{ijp}+\frac{h_p d_{ijp}}{\bar{b}_p d_{ijp}+\sum_{p'\in P^d_{ij}}\bar{b}_{p'}d_{ijp'}}\right]$$
$$= \alpha_{ijpk}+\frac{h_p d_{ijp}}{\sum_{(i',p')\in IP^k_j}\bar{b}_{p'}d_{i'jp'}}-\frac{h_p d_{ijp}}{\bar{b}_p d_{ijp}+\sum_{p'\in P^d_{ij}}\bar{b}_{p'}d_{ijp'}}. \quad (9)$$

Then, α_{ijpk} amounts to the extra transportation and pipeline inventory costs for shipping flow (i,j,p) through cross-dock k instead of shipping it directly, and $-\delta_{ijpk}$ is the maximum possible saving for shipping flow (i,j,p) through cross-dock k instead of shipping it directly if $\delta_{ijpk} < 0$.

Next, we present a greedy heuristic to obtain a feasible distribution strategy for the flows based on the values of δ_{ijpk}.

Algorithm 1: Greedy Heuristic for Problem (P).

Step 1. Set $IP^k_j = IP_j$ and $P^d_{ij} = \varnothing$, $N = \{(i,j,p,k)\colon (i,j,p) \in F,\ k \in K\}$. Calculate α_{ijpk}, $(i,j,p,k) \in N$ by (8) and sort them in nonincreasing order.

Step 2. Set stopflag = true. Number the remaining items in N as $1,2,\ldots,|N|$ according to the sorted order, i.e., $\alpha_{n_1} \ge \alpha_{n_2}$ if $n_1 < n_2$. For $n = 1$ to $|N|$, let (i,j,p,k) be the nth item. Calculate δ_{ijpk} by (9). If $\delta_{ijpk} \ge 0$, set $x_{ijpk} = 0$, $IP^k_j = IP^k_j\backslash\{(i,p)\}$, $N = N\backslash\{(i,j,p,k)\}$, stopflag = false. If for all $k \in K$, $x_{ijpk} = 0$, then set $x_{ijp0} = 1$ and $P^d_{ij} = P^d_{ij}\cup\{p\}$.

Step 3. If $N = \varnothing$, stop; else if stopflag = false, go to Step 2; otherwise, go to Step 4.

Step 4. Denote (i',j',p',k') as the one that has the minimum δ_{ijpk} $\forall (i,j,p,k) \in N$ (ties are broken arbitrarily). Let $K' = \{k \in K\colon (i',j',p',k) \in N\}$. Set $x_{i'j'p'k'} = 1$. $\forall k \in K'\backslash\{k'\}$, set $x_{i'j'p'k} = 0$ and update $IP^k_{j'} = IP^k_{j'}\backslash\{(i',p')\}$. Set $N = N\backslash\{(i',j',p',k)\colon k \in K'\}$. If $N \neq \varnothing$, go to Step 2; otherwise, stop.

The larger the value of α_{ijpk}, the more possible that flow (i,j,p) will not go through cross-dock k. Hence, in Step 1 we sort α_{ijpk} in decreasing order so that we can reduce the set of F^k_j (or IP^k_j) and fix more x_{ijpk} in Step 2.

In Step 2, if $\delta_{ijpk} \ge 0$, then we set $x_{ijpk} = 0$ because for flow (i,j,p), going through cross-dock k is a worse distribution strategy than being shipped directly.

Step 4 first assigns the flow to the cross-dock with the maximum possible savings (from Steps 2 and 3, we can see that $\delta_{i'j'p'k'} < 0$), then it goes back to Step 2 to fix more variables based on this decision.

The greedy heuristic is illustrated by Example 1.

EXAMPLE 1. Consider an instance of problem (*P*) with two suppliers, two plants, one product, four flows, and one cross-dock. The supplier and plant for each flow are

Flow	Supplier	Plant
1	1	1
2	1	2
3	2	1
4	2	2

After data processing as illustrated in §§2.1 and 2.2, we have

$$\bar{c}_{ijpk}d_{ijp} = \begin{pmatrix} 10 & 11 \\ 13 & 16 \\ 12 & 14 \\ 11 & 17 \end{pmatrix} \quad \text{for } k = 0, 1,$$

$$h_p d_{ijp} = (4, 3, 2, 3),$$

$$\bar{b}_p d_{ijp} = (1, 0.75, 0.5, 0.75).$$

Step 1. $IP_1^1 = \{(1,1),(2,1)\}$, $IP_2^1 = \{(1,2),(2,2)\}$, $P_{11}^d = P_{12}^d = P_{21}^d = P_{22}^d = \varnothing$, $N = \{(1,1,1,1),(1,2,1,1),(2,1,1,1),(2,2,1,1)\}$. $\alpha_{1111} = 1$, $\alpha_{1211} = 3$, $\alpha_{2111} = 2$, $\alpha_{2211} = 6$. The sorted order of N is $(2,2,1,1)$, $(1,2,1,1)$, $(2,1,1,1)$, $(1,1,1,1)$.

ITERATION 1.

Step 2. stopflag = true. Number items $(2,2,1,1)$, $(1,2,1,1)$, $(2,1,1,1)$, $(1,1,1,1)$ as 1, 2, 3, and 4.

For $n = 1$, $\delta_{2211} = 6 + 3/(0.75 + 0.75) - 3/0.75 = 4 > 0$, $x_{2211} = 0$, stopflag = false. $IP_2^1 = \{(1,2)\}$, $N = N \backslash \{(2,2,1,1)\}$. $x_{2210} = 1$, $P_{22}^d = \{1\}$.

For $n = 2$, $\delta_{1211} = 3 + 3/0.75 - 3/0.75 = 3 > 0$, $x_{1211} = 0$. $IP_2^1 = \varnothing$, $N = N \backslash \{(1,2,1,1)\}$. $x_{1210} = 1$, $P_{12}^d = \{1\}$.

For $n = 3$, $\delta_{2111} = 2 + 2/(1 + 0.5) - 2/0.5 = -0.667 < 0$.

For $n = 4$, $\delta_{1111} = 1 + 4/(1 + 0.5) - 4/1 = -0.333 < 0$.

Step 3. $N \neq \varnothing$ and stopflag = false, go to Step 2.

ITERATION 2.

Step 2. stopflag = true. Number items $(2,1,1,1)$, $(1,1,1,1)$ as 1 and 2.

For $n = 1$, $\delta_{2111} = 2 + 2/(1 + 0.5) - 2/0.5 = -0.667 < 0$.

For $n = 2$, $\delta_{1111} = 1 + 4/(1 + 0.5) - 4/1 = -0.333 < 0$.

Step 3. stopflag = true, go to Step 4.

Step 4. $(2,1,1,1)$ has the minimum value of δ $x_{2111} = 1$, $x_{2110} = 0$, $N = \{(1,1,1,1)\}$.

ITERATION 3.

Step 2. stopflag = true. Number items $(1,1,1,1)$ as 1.

For $n = 1$, $\delta_{1111} = 1 + 4/(1 + 0.5) - 4/1 = -0.333 < 0$.

Step 3. stopflag = true, go to Step 4.

Step 4. $(1,1,1,1)$ has the minimum value of δ. $x_{1111} = 1$, $x_{1110} = 0$, $N = \varnothing$, stop.

The solution generated by the greedy heuristic is to shipping flows 1 and 3 through cross-dock and shipping flows 2 and 4 directly. In fact, it is an optimal solution for this example.

4. Lagrangian Relaxation Heuristic and Branch-and-Bound Algorithm

In this section, we present an LR of problem (*P*) and develop heuristic and a BB algorithm based on the LR to solve problem (*P*).

4.1. Lagrangian Relaxation

The LR of problem (*P*) with respect to constraints (2) is

$$\text{LR}(\lambda) \quad \min\left[\sum_{k\in K^0} g_k(X) + \sum_{(i,j,p)\in F} \lambda_{ijp}\left(1 - \sum_{k\in K^0} x_{ijpk}\right)\right]$$

subject to

$$x_{ijpk} \in \{0,1\} \quad \forall (i,j,p) \in F,\ k \in K^0.$$

Problem LR(λ) can be decomposed into $|K| + 1$ subproblems corresponding to each $k \in K^0$:

$$\text{LR}(\lambda, k) \quad \min\left[g_k(X) - \sum_{(i,j,p)\in F} \lambda_{ijp} x_{ijpk}\right]$$

subject to

$$x_{ijpk} \in \{0,1\} \quad \forall (i,j,p) \in F.$$

4.1.1. LR(λ, 0). From Equation (5) in §2.1, problem LR(λ, 0) can be rewritten as

$$\min \sum_{i\in I}\sum_{j\in J}\left(\sum_{p\in P_{ij}} (\bar{c}_{ijp0}d_{ijp} - \lambda_{ijp})x_{ijp0} + \frac{\sum_{p\in P_{ij}} h_p d_{ijp} x_{ijp0}}{\sum_{p\in P_{ij}} \bar{b}_p d_{ijp} x_{ijp0}}\right)$$

subject to

$$x_{ijp0} \in \{0,1\} \quad \forall (i,j,p) \in F.$$

This problem can be further decomposed into $|I| * |J|$ subproblems corresponding to each $i \in I$, $j \in J$:

$$SP^0(\lambda, i, j) \quad \min \sum_{p\in P_{ij}} (\bar{c}_{ijp0}d_{ijp} - \lambda_{ijp})x_{ijp0} + \frac{\sum_{p\in P_{ij}} h_p d_{ijp} x_{ijp0}}{\sum_{p\in P_{ij}} \bar{b}_p d_{ijp} x_{ijp0}}$$

subject to

$$x_{ijp0} \in \{0,1\} \quad \forall p \in P_{ij}.$$

4.1.2. LR(λ, k), $k \in K$. From Equation (7) in §2.2, problem LR(λ, k) can be rewritten as

$$\sum_{j\in J}\left(\sum_{(i,p)\in IP_j} (\bar{c}_{ijpk}d_{ijp} - \lambda_{ijp})x_{ijpk} + \frac{\sum_{(i,p)\in IP_j} h_p d_{ijp} x_{ijpk}}{\sum_{(i,p)\in IP_j} \bar{b}_p d_{ijp} x_{ijpk}}\right)$$

subject to

$$x_{ijpk} \in \{0, 1\} \quad \forall (i, j, p) \in F.$$

This problem can be further decomposed into $|J|$ subproblems corresponding to each $j \in J$:

$$SP^k(\lambda, j) \quad \min \sum_{(i,p)\in IP_j} (\bar{c}_{ijpk} d_{ijp} - \lambda_{ijp}) x_{ijpk} + \frac{\sum_{(i,p)\in IP_j} h_p d_{ijp} x_{ijpk}}{\sum_{(i,p)\in IP_j} \bar{b}_p d_{ijp} x_{ijpk}}$$

subject to

$$x_{ijpk} \in \{0, 1\} \quad \forall (i, p) \in IP_j.$$

4.1.3. Solving SP^k, $k \in K^0$. It is easy to see that all the subproblems SP^k, $k \in K^0$, share the following structure:

$$(GSP) \quad \min Z = \sum_{n\in N} u_n z_n + \frac{\sum_{n\in N} v_n z_n}{\sum_{n\in N} w_n z_n} \tag{10}$$

subject to

$$z_n \in \{0, 1\} \quad \forall n \in N. \tag{11}$$

Note that objective function (10) is neither convex nor concave and $v_n, w_n > 0$ $\forall n \in N$. Denote $Z(N')$ as the objective function value of (GSP) when $z_n = 1$ $\forall n \in N'$ and $z_n = 0$ $\forall n \notin N'$. We assume that the objective function value equals 0 when $z_n = 0$ $\forall n \in N$, i.e., $Z(\varnothing) = 0$.

4.1.3.1. v_n/w_n Is Constant. First, we consider a special case of problem (GSP) when v_n/w_n is constant. v_n/w_n will be constant when one of the following conditions is met:

- There is only one product under consideration.
- For each plant, there is only one type of product shipped to it.
- h_p/b_p is constant for all $p \in P$ when there are multiple products.

Suppose that $v_n/w_n = q$ $\forall n \in N$. Then, $\sum_{n\in N'} v_n / \sum_{n\in N'} w_n = q$ $\forall N' \subset N$. Let $N^- = \{n \in N: u_n < 0\}$. For any $N' \subset N$, we have

$$Z(N') = \sum_{n\in N'} u_n + \frac{\sum_{n\in N'} v_n}{\sum_{n\in N'} w_n} \geq \sum_{n\in N^-} u_n + q = \sum_{n\in N^-} u_n + \frac{\sum_{n\in N^-} v_n}{\sum_{n\in N^-} w_n} = Z(N^-).$$

Hence, if $\sum_{n\in N^-} u_n \geq -q$, $z_n = 0$ $\forall n \in N$ is optimal to (GSP); otherwise, $z_n = 1$ $\forall n \in N^-$ and $z_n = 0$ $\forall n \in N \backslash N^-$ is optimal to (GSP).

4.1.3.2. v_n/w_n Is Not Constant. Next, we develop a BB algorithm to solve (GSP) when v_n/w_n is not constant. Before proposing the algorithm, we present some properties of (GSP).

Without loss of generality, we assume that v_n/w_n, $n \in N$, are sorted in nondecreasing order. Problem (GSP) has the following properties.

Theorem 1. *If there exists an l such that for any $n \geq l$, $u_n \geq 0$, then there exists an optimal solution such that $z_n = 0$ $\forall n \geq l$.*

Proof. Suppose that z^* is an optimal solution and let $m = \max\{n \in N: z_n^* = 1\}$. Suppose that $m \geq l$. Construct a new solution z^1 with $z_m^1 = 0$ and $z_n^1 = z_n^*$ $\forall n \neq m$. Let $a = \sum_{n<m} v_n z_n^*$ and $b = \sum_{n<m} w_n z_n^*$. Because $v_m/w_m \geq v_n/w_n$ $\forall n < m$, then $a/b \leq (a + v_m)/(b + w_m)$. Hence,

$$Z^* - Z^1 = u_m + \frac{a + v_m}{b + w_m} - \frac{a}{b} \geq u_m \geq 0.$$

Because z^* is optimal, z^1 is also optimal. We can construct an optimal solution satisfying $z_n = 0$ $\forall n \geq l$ by repeating this procedure. □

Based on Theorem 1, we can remove the set of decision variables z_n, $n \geq l$, and focus on the remaining decision variables.

Theorem 2. *If there exist l and m such that $u_l \leq u_m$, $v_l \leq v_m$, $w_l \geq w_m$, and at least one of the inequalities is strict, then $z_l \geq z_m$ is held in the optimal solution.*

Proof. Suppose that z^* is an optimal solution and $z_l^* < z_m^*$, i.e., $z_l^* = 0$, $z_m^* = 1$. Construct a new solution z^1 with $z_l^1 = 1$, $z_m^1 = 0$, and $z_n^1 = z_n^*$ $\forall n \neq l, m$. Let $a = \sum_{n\neq l,m} v_n z_n^*$ and $b = \sum_{n\neq l,m} w_n z_n^*$. Hence,

$$Z^* - Z^1 = u_m + \frac{a + v_m}{b + w_m} - \left(u_l + \frac{a + v_l}{b + w_l}\right) = (u_m - u_l) + \left(\frac{a + v_m}{b + w_m} - \frac{a + v_l}{b + w_l}\right) > 0.$$

This contradicts the fact that z^* is optimal. □

Based on Theorem 2, we can conclude that if $z_l = 0$ in the optimal solution, then z_m must also be 0.

Let $N^1 \subset N$ be the set of variables that are fixed to be 1, $N^0 \subset N$ be the set of variables that are fixed to be 0, and $N^u = N \backslash (N^0 \cup N^1)$ be the set of variables that are undetermined. Let $V^1 = \sum_{n\in N^1} v_n$, $W^1 = \sum_{n\in N^1} w_n$, $V^2 = \sum_{n\in N\backslash N^0} v_n$, and $W^2 = \sum_{n\in N\backslash N^0} w_n$. Next, we present two theorems based on fixing some variables to be 1 or 0 for the subproblem with respect to N^0, N^1, N^u. In the proof of the next two theorems, "optimal solution" means the best solution with variables in N^1 and N^0 fixed to be 1 and 0, respectively.

Theorem 3. *$\forall m \in N^u$, if $u_m + v_m/(w_m + W^1) \leq V^1 w_m / [W^2(W^2 - w_m)]$, then there exists an optimal solution such that $z_m = 1$.*

Proof. Suppose that there exist an optimal solution z^* and a variable z_m such that $z_m^* = 0$ and $u_m + v_m/(w_m + W^1) \leq V^1 w_m / [W^2(W^2 - w_m)]$. Construct a new solution z^1 with $z_m^1 = 1$ and $z_n^1 = z_n^*$ $\forall n \neq m$. Let $a = \sum_{n\in N} v_n z_n^*$ and $b = \sum_{n\in N} w_n z_n^*$. Then, we have $V^1 \leq a \leq V^2 - v_m$ and $W^1 \leq b \leq W^2 - w^m$. Therefore,

$u_m + v_m/(b+w_m) \le u_m + v_m/(W^1+w_m)$ and $aw_m/[b(b+w_m)] \ge V^1 w_m/[(W^2-w_m)W^2]$. Hence,

$$\begin{aligned} Z^1 - Z^* &= u_m + \frac{a+v_m}{b+w_m} - \frac{a}{b} = u_m + \frac{bv_m - aw_m}{b(b+w_m)} \\ &= \left(u_m + \frac{v_m}{b+w_m}\right) - \frac{aw_m}{b(b+w_m)} \\ &\le u_m + \frac{v_m}{W^1+w_m} - \frac{V^1 w_m}{(W^2-w_m)W^2} \le 0. \end{aligned}$$

Because z^* is optimal, z^1 is also optimal. □

Theorem 4. *$\forall m \in N^u$, if $u_m \ge w_m(V^2 - v_m)/(W^1(w_m + W^1)) - v_m/W^2$, then there exists an optimal solution such that $z_m = 0$.*

Proof. Suppose that there exist an optimal solution z^* and a variable z_m such that $z_m^* = 1$ and $u_m \ge w_m(V^2 - v_m)/(W^1(w_m + W^1)) - v_m/W^2$. Construct a new solution z^1 with $z_m^1 = 0$ and $z_n^1 = z_n^*$ $\forall n \ne m$. Let $a = \sum_{n\in N,\, n\ne m} v_n z_n^*$ and $b = \sum_{n\in N,\, n\ne m} w_n z_n^*$. Then,

$$Z^* - Z^1 = u_m + \frac{a+v_m}{b+w_m} - \frac{a}{b} = u_m + \frac{bv_m - aw_m}{b(b+w_m)}.$$

Because $a \le V^2 - v_m$, $b \ge W^1$, $b + w_m \le W^2$,

$$\begin{aligned} u_m &\ge \frac{w_m(V^2 - v_m)}{W^1(w_m + W^1)} - \frac{v_m}{W^2} \\ &\ge \frac{w_m a}{b(b+w_m)} - \frac{v_m}{w_m + b} = -\frac{bv_m - aw_m}{b(b+w_m)}. \end{aligned}$$

Hence, $Z^* - Z^1 \ge 0$. Because z^* is optimal, z^1 is also optimal. Hence, $m \in N^0$. □

Based on Theorems 1–4, we can fix some variables for a subproblem of (*GSP*) with respect to N^0, N^1, N^u. In Algorithm 2, we assume that we have already implemented Theorem 1.

Algorithm 2: Fix Variables for a (*GSP*) Subproblem.

Step 1. Calculate $V^1 = \sum_{n\in N^1} v_n$, $W^1 = \sum_{n\in N^1} w_n$, $V^2 = \sum_{n\in N\setminus N^0} v_n$, $W^2 = \sum_{n\in N\setminus N^0} w_n$.

Step 2. Sort $u_n + v_n/w_n$, $n \in N^u$, in nondecreasing order. Let $n_1, n_2, \ldots, n_{|N^u|}$ be the index of the sorted order. Set $k = 1$.

Step 3. If $k > |N^u|$, go to Step 4. Else, if $u_{n_k} + v_{n_k}/(W^1 + w_{n_k}) \le V^1 w_{n_k}/[W^2(W^2 - w_{n_k})]$, set $N^1 = N^1 \cup \{n_k\}$, $N^u = N^u \setminus \{n_k\}$, $V^1 = V^1 + v_{n_k}$, $W^1 = W^1 + w_{n_k}$, set $k = k+1$, repeat Step 3.

Step 4. For each $n \in N^u$, if $u_n \ge w_n(V^2 - v_n)/[W^1(W^1 + w_n)] - v_n/W^2$, set $N^0 = N^0 \cup \{n\}$, $V^2 = V^2 - v_n$, $W^2 = W^2 - w_n$.

Step 5. For each $n \in N^u$, if there exists an $l \in N^0$ such that $u_l \le u_n$, $v_l \le v_n$, $w_l \ge w_n$ and at least one of the inequalities are strict, set $N^0 = N^0 \cup \{n\}$, $V^2 = V^2 - v_n$, $W^2 = W^2 - w_n$.

Let $v_{\min} = \min\{v_n : n \in N^u\}$ and $w_{\max} = \max\{w_n : n \in N^u\}$. Sort u_n in N^u in nondecreasing order and denote them as $u_{n_1}, u_{n_2}, \ldots, u_{n_{|N^u|}}$. A lower bound of subproblem (*GSP*) with respect to N^0, N^1, N^u can be calculated as follows:

Theorem 5. *A lower bound of subproblem* (*GSP*) *with respect to N^0, N^1, N^u is*

$$\mathrm{LB} = \sum_{n\in N^1} u_n + \min\left\{\sum_{j=1}^{k} u_{n_j} + \frac{V^1 + kv_{\min}}{\min\{W^2, W^1 + kw_{\max}\}}: k = 0, 1, \ldots, |N^u|\right\}. \quad (12)$$

Proof. For any subset N' of N^u,

$$\begin{aligned} Z(N') &= \sum_{n\in N^1} u_n + \sum_{n\in N'} u_n + \frac{V_1 + \sum_{n\in N'} v_n}{W_1 + \sum_{n\in N'} w_n} \\ &\ge \sum_{n\in N^1} u_n + \sum_{j=1}^{k} u_{n_j} + \frac{V^1 + kv_{\min}}{\min\{W^2, W^1 + kw_{\max}\}}, \end{aligned}$$

where $k = |N'|$. Hence, LB is a lower bound of (*GSP*). □

Based on the above theorems, we develop a BB algorithm to solve problem (*GSP*).

Algorithm 3: Branch-and-Bound Algorithm for (*GSP*).

Step 1. Let $N^0 = N^1 = \varnothing$, $N^u = N$. Sort v_n/w_n, $n \in N^u$, in nondecreasing order. Let $n_1, n_2, \ldots, n_{|N^u|}$ be the index of the sorted order. Set $k = |N^u|$.

Step 2. If $u_{n_k} < 0$, go to Step 3. Else, update $N^0 = N^0 \cup \{n_k\}$, $N^u = N^u \setminus \{n_k\}$. Set $k = k - 1$. Go to Step 2.

Step 3. Let $Z^* = \infty$. Initialize the BB tree with a single node (root node) corresponding to problem (*GSP*). Select the root node from the BB tree.

Step 4. Use Algorithm 2 to fix variables for the selected subproblem. If $N^u = \varnothing$, a feasible solution z' is found. Let Z' be its objective function value. If $Z' < Z^*$, then $z^* = z'$, $Z^* = Z'$, fathom the node; otherwise, calculate LB based on (12). If LB $> Z^*$, fathom the node and go to Step 5; otherwise, select a variable to branch. Add two nodes corresponding to the two subproblems to the BB tree.

Step 5. If all the nodes of the BB tree are fathomed, stop. Otherwise, select an unfathomed node from the BB tree and go to Step 4.

Algorithm 3 is illustrated by the following example:

Example 2. Consider a (*GSP*) with

$$u_i = (1, -1, -2, -3, 1), \quad (13)$$

$$v_i = (1, 1, 3, 4, 5), \quad (14)$$

$$w_i = (1, 1, 1, 1, 1). \quad (15)$$

Step 1 of Algorithm 3: $N^0 = N^1 = \varnothing$, $N^u = N$. v_n/w_n are already sorted.

Step 2 of Algorithm 3: $z_5 = 0$, $N^0 = \{5\}$, $N^u = \{1, 2, 3, 4\}$.

Step 1 of Algorithm 2: $V^1 = W^1 = 0$, $V^2 = 9$, $W^2 = 4$.

Steps 2 and 3 of Algorithm 2: $N^1 = \{2, 3, 4\}$, $V^1 = 8$, $W^1 = 3$, $V^2 = 9$, $W_2 = 4$.

Step 4 of Algorithm 2: $z_1 = 0$, $N^0 = \{1, 5\}$. Hence, the optimal solution is $z_1 = z_5 = 0$ and $z_2 = z_3 = z_4 = 1$ with an objective value of $-10/3$.

4.2. Lagrangian Relaxation Heuristic

The Lagrangian dual of problem (P) with respect to constraints (2) is

$$(LD) \quad \max_{\lambda} \mathrm{LR}(\lambda).$$

It provides a lower bound to the primal problem. The dual problem (LD) can be solved by the subgradient algorithm (Beasley 1993).

In each iteration of the subgradient algorithm, for a given λ, we obtain a LR solution. Based on this solution, we can construct a feasible solution to problem (P) as follows:

- First, consider the flows that are assigned to only one cross-dock in the solution. (The corresponding constraints in (2) are satisfied. Note that direct shipment is regarded as assigned to dummy cross-dock 0.) In the new solution, the flows are still assigned to the same cross-dock as in the LR solution.
- Next, consider the flows that are assigned to more than one cross-dock in the solution. (The corresponding constraints in (2) are not satisfied with the left-hand side greater than the right-hand side.) For each such flow and each cross-dock that is assigned in the LR solution, we calculate the cost by assigning the flow to the selected cross-dock and removing it from the nonselected cross-docks. When we remove the flow from the nonselected cross-docks, the cost for the flows that are assigned to the nonselected cross-docks may increase. In the new solution, the flow is assigned to the cross-dock that has the minimum objective value.
- Finally, we consider the flows that are not assigned to any cross-dock in the solution. (The corresponding constraints in (2) are not satisfied with the left-hand side less than the right-hand side.) For each such flow, we calculate the cost by assigning the flow to each cross-dock. When we add a flow to a cross-dock, the cost for the flows that are assigned to the selected cross-dock may decrease. In the new solution, the flow is assigned to the cross-dock with the minimum objective function value.

In the LR heuristic, we apply the above steps only under the following conditions:

(1) when the lower bound is improved;

(2) when the LR solution satisfies constraints (2) for most flows, i.e., $\sum_{(i,j,p)\in F}[\sum_{k\in K^0} x_{ijpk}^{lr} - 1] \le r$, where x^{lr} is the LR solution and r is a given positive number.

4.3. Branch-and-Bound Algorithm

A BB algorithm can be developed to solve problem (P). For each subproblem, the lower bound is obtained by their corresponding LR with respect to constraints (2). When branching, we select a flow and generate $|K| + 1$ subproblems corresponding to cross-dock k, $k \in K^0$. In each subproblem, the selected flow is assigned to the corresponding cross-dock.

5. Computational Results

In this section, the performances of the LR heuristic and the BB algorithm are tested by solving randomly generated problems of different sizes. The algorithm is programmed in Microsoft Visual C++ 5.0. All of the experimental tests were carried out on a Dell OptoPlex GX240 with 512 MB RAM and 1.8 GHz CPU. Computation times are in seconds.

Let $U[a, b]$ be a uniform distribution between a and b. The problems were randomly generated by the following scheme. The Cartesian coordinates of the suppliers, the plants, and the cross-docks are drawn from $U[0, 1{,}000]$. The distances are calculated using Euclidean distances. The transportation costs between suppliers, plants, and cross-docks c_{ij}^d, c_{ik}^i, c_{jk}^o are equal to the corresponding distance, and the transportation times t_{ij}^d, t_{ik}^i, t_{jk}^o are the values of their corresponding distances divided by 4,000. The quantities d_{ijp} are drawn from $U[10, 100]$. The truck capacity C is 1,000 units. The inventory-carrying cost h_p and the truck capacity occupied by the products b_p are generated from $U[1, 10]$. The time spent at cross-dock T_k is drawn from $U[0.1, 0.2]$.

Three data sets were generated based on the scheme described above. The first one has 50 flows with $|I| = |J| = 5$ and $|P| = |K| = 3$, the second one has 100 flows with $|I| = 10$, $|J| = |P| = 5$, and $|K| = 3$, and the third one has 200 flows with $|I| = |J| = 10$ and $|P| = |K| = 5$. For each data set, we change the quantities d_{ijp} and inventory-carrying costs h_p to see the impact of these changes on the solution.

The parameters for the LR are as follows. The initial feasible solution is generated by the greedy heuristic. The initial Lagrangian multiplier corresponding to each flow is equal to the minimum of transportation plus pipeline cost for that flow. The initial value of β (the factor used in determining the step size) is 2. Its value will be halved if the lower bound does not improve after 10 iterations. In the LR heuristic, the maximum number of iterations is 50. In the BB algorithm, the maximum number of LR iterations is 50 and 20 for the root node and other nodes, respectively.

The computational results are presented in Table 3. To express the changes in d_{ijp} and h_p, in the table, Column "w_d" is the demand weight, which means that

Table 3 Computational Results for Small-Size Problems

No. of flows	w_d	w_i	LR time	BB time	LDG (%)	LRG (%)	NFSD	Direct cost (%)
50	1	0.1	0.046	0.046	0	0	14	35.61
		0.3	0.093	0.093	0	0	0	0.00
		0.5	0.141	0.157	0.052	0.044	0	0.00
		0.7	0.047	0.047	0	0	0	0.00
		1	0.14	0.515	0.022	0.015	0	0.00
100	1	0.1	0.156	0.203	0.023	0.014	62	62.26
		0.3	0.156	0.156	0	0	40	40.78
		0.5	0.188	0.188	0	0	21	18.38
		0.7	0.188	0.188	0	0	12	10.89
		1	0.079	0.079	0	0	7	5.45
200	1	0.1	0.282	0.297	0	0	141	64.77
		0.3	0.626	16.068	0.002	0	99	46.01
		0.5	0.938	0.938	0	0	84	38.72
		0.7	0.391	0.391	0	0	77	36.71
		1	0.828	1	0.023	0.020	66	30.21
50	2	0.1	0.093	0.703	0.204	0.134	24	54.62
		0.3	0.11	1.375	0.883	0.682	8	17.71
		0.5	0.109	0.234	0.131	0.051	2	6.85
		0.7	0.109	0.14	0.039	0.037	0	0.00
		1	0.125	0.547	0.039	0.018	0	0.00
100	2	0.1	0.14	2.547	0.020	0.001	81	79.10
		0.3	0.188	1.094	0.037	0	67	69.88
		0.5	0.313	4.017	0.582	0.196	49	52.45
		0.7	0.329	2.063	0.057	0.023	40	39.69
		1	0.328	1.735	0.376	0.284	40	40.63
200	2	0.1	0.297	0.5	0.052	0.035	180	87.77
		0.3	0.343	32.288	2.879	0.465	146	72.19
		0.5	0.563	21.145	1.542	0.478	141	69.32
		0.7	0.469	16.144	2.398	0.870	128	63.69
		1	0.563	8.69	3.462	1.572	114	56.23
50	3	0.1	0.094	1.048	0.009	0	40	81.36
		0.3	0.079	0.079	0	0	23	52.23
		0.5	0.109	0.156	1.254	0.266	19	46.89
		0.7	0.125	0.781	1.753	0.575	7	15.56
		1	0.094	0.094	0	0	5	12.09
100	3	0.1	0.094	1.86	0.115	0.047	82	79.43
		0.3	0.125	0.125	0	0	81	79.32
		0.5	0.218	1.922	0.198	0	70	72.22
		0.7	0.281	1.359	0.565	0.131	62	66.93
		1	0.265	1.406	1.180	0.110	61	65.87
200	3	0.1	0.219	15.269	1.020	0.005	194	95.74
		0.3	0.281	25.302	3.556	0.210	180	88.64
		0.5	0.344	150.39	5.223	0.530	163	81.34
		0.7	0.328	55.728	4.807	0.663	159	78.57
		1	0.563	24.457	1.236	0.278	155	77.09

the demand of the corresponding problem is $w_d d_{ijp}$ and Column "w_i" is the inventory weight, which means that the inventory-carrying cost of the corresponding problem is $w_i h_p$. Column "LR time" ("BB time") is the CPU time for the LR heuristics (the BB algorithm). Column "LDG" is the duality gap of the LR heuristic, which is equal to $(UB_{LR} - LB_{LR})/LB_{LR}$. Column "LRG" is the gap between the upper bound found by the LR heuristic and the optimal solution found by the BB algorithm. Column "NFSD" is the number of flows that are shipped directly. Column "Direct cost" is the percentage of the direct cost in the total cost.

From Table 3, we can conclude the following.

• The LR heuristic is very fast. The LR time increases with the number of flows. The demand weight and the inventory weight have no significant impact on the CPU time. This is also true for the BB time.

• The LR lower bound is tight. It is within 5% of the LR upper bound for almost all the problems except one. The average LDG is 0.75%.

Table 4 Computational Results for Medium-Size Problems Solved by BB

$\|I\|$	$\|J\|$	$\|P\|$	$\|K\|$	$\|F\|$	BB time
10	10	10	5	500	12,369.20
10	10	100	5	500	12,279.20
10	50	10	5	500	38.77
10	50	100	5	500	386.80
50	10	10	5	500	24.50
50	10	100	5	500	5,386.73
50	50	10	5	500	93.43
50	50	100	5	500	371.77
10	10	10	5	1,000	6,629.47
10	10	100	5	1,000	17,878.56
10	50	10	5	1,000	2,070.90
10	50	100	5	1,000	1,745.86

• The LR heuristic generates high-quality solutions. The gap between the LR heuristic solution and the optimal solution is less than 1% for almost all the problems except one. The average LRG is 0.172%.

Note that the values LRG and LDG can be reduced if we set the stop criteria for the subgradient algorithm at the root node of the BB tree with a larger iteration limit. In the experimental experience, only 50 iterations are performed.

To report on the size of problems that the LR heuristic and the BB algorithm can solve in a reasonable time (for example, an hour of CPU time), we generated medium-size problems to be solved by the BB algorithm and large-size problems to be solved by the LR heuristic (here we increase the number of iterations in the subgradient algorithm to 100). The results are listed in Tables 4 and 5 for the BB algorithm and the LR heuristics, respectively.

From Table 4, we can see that the BB algorithm spent more than one hour to obtain an optimal solution for all the problems with 10 plants except for one, while it spent less than one hour for all the problems with 50 plants. The reason behind this might be that for a given number of flows, when the number of plants is small, the number of flows assigned to each plant is large, and therefore the corresponding (*GSP*) problem is large.

Table 5 Computational Results for Large-Size Problems Solved by LR

$\|I\|$	$\|J\|$	$\|P\|$	$\|K\|$	$\|F\|$	LR time	LRG (%)
10	100	100	5	3,000	1,523.41	1.17
10	100	1,000	5	3,000	1,437.07	0.70
10	100	100	5	5,000	3,721.83	1.55
10	100	1,000	5	5,000	3,258.39	0.88
100	10	100	5	3,000	2,156.30	0.06
100	10	1,000	5	3,000	723.57	0.00
100	10	100	5	5,000	2,785.82	0.06
100	10	1,000	5	5,000	3,328.56	0.04
100	100	100	5	3,000	1,287.02	0.39
100	100	1,000	5	3,000	1,285.74	0.41
100	100	100	5	5,000	3,428.06	0.35
100	100	1,000	5	5,000	3,620.60	0.82

Table 6 Computational Results for a Single Product

$\|I\|$	$\|J\|$	$\|K\|$	$\|F\|$	LR time	BB time	LDG (%)	LRG (%)
50	50	5	500	0.063	0.125	0.0032	0
70	70	5	1,200	0.281	0.297	0.0002	0
100	100	5	2,000	1.313	5.407	0.1956	0.0010

The longest time that the LR heuristic took for the problems in Table 5 is a little more than one hour. It also shows that the quality of the solutions generated by the LR heuristic is still high for large-size problems. The largest gap is only 1.55%.

We also tested the performance of the LR heuristic and the BB algorithm for the problems of shipping a single type of product. The results are listed in Table 6. For this type of problems, our algorithms can find a solution very quickly for large problems, and the solution quality is very good.

Next, we investigate the change in the cost when we change the demand weight and inventory weight. We used the data set with 200 flows as our example. Figures 2 and 3 show the sensitivity of the total cost and the number of flows that are shipped directly (NFSD) to the changes in the demand and inventory weights. As expected, when the demand and inventory weights increase, the total cost increases. NFSD increases as the demand weight increases. The reason for this is that when demand weight is high, the frequency of direct shipping increases, and therefore fewer flows need to be shipped through cross-dock. However, NFSD decreases as the inventory weight increases. This is due to the fact that when the inventory weight is high, it is more desirable to ship the flow at a higher delivery frequency, which can be realized by shipping it through a cross-dock together with other flows.

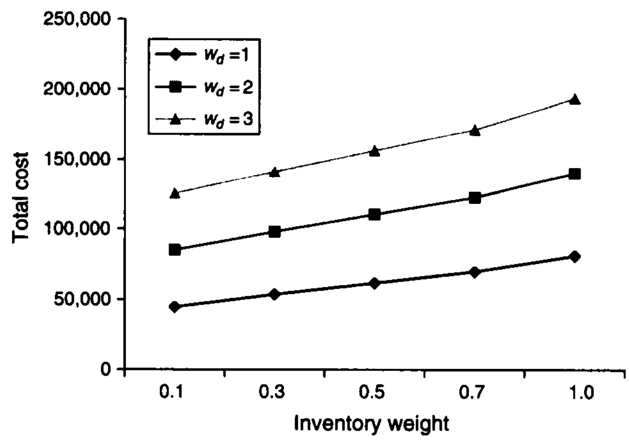

Figure 2 Sensitivity of Total Cost to the Change of Demand and Inventory Weights

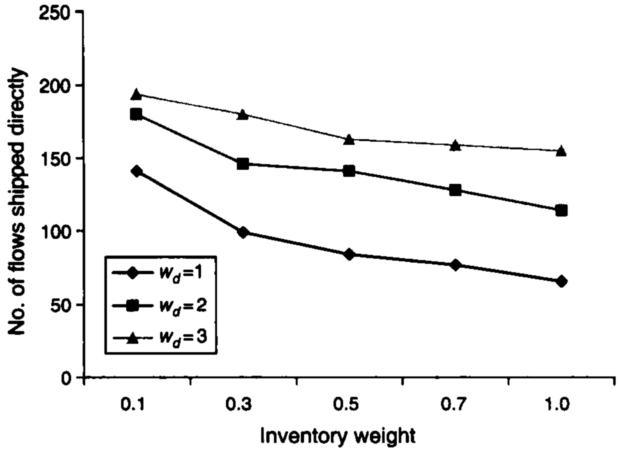

Figure 3 Sensitivity of NFSD to the Change of Demand and Inventory Weights

Figure 4 illustrates the cost distribution for different demand weights for the data set when the inventory weight is fixed to be 1. We refer to "transportation cost" as the sum of the transportation and pipeline inventory cost (the reason we combine these two is that both transportation and pipeline inventory costs are determined by the demand quantity instead of delivery frequency) and "inventory cost" as the plant inventory cost. It is not surprising to see that both the direct transportation and inventory costs increase with the demand weight because more flows are shipped directly.

Figure 5 shows the cost distribution for different inventory weights for the data set when the demand weight is fixed to be 1. When the inventory weight increases, the direct transportation cost decreases, which is consistent with the fact that NFSD decreases. Although the direct transportation cost decreases, the direct inventory cost increases with the inventory weight. Both the cross-dock transportation and inventory costs increase with the inventory weight.

6. Summary

We considered the problem of selecting the appropriate distribution strategy for delivering a family of products from a set of suppliers to a set of plants so that the total transportation, pipeline inventory, and plant inventory costs are minimized. Two heuristics and an exact algorithm are presented. The algorithms were tested on randomly generated data and their performances are satisfactory.

Some directions for future work are:

- To extend the model to include the strategy of milk-run.
- In practice, management may require that the number of shipments in a week is integer so that the same pattern can be repeated each week. We suggest extending the model to take this into account.
- To extend the model by including different types of truck and less-than-full-truckload strategies. We may have a truck fleet with different capacities. Choosing the type of truck for each route is also an important decision faced by management. Furthermore, sometimes it may be beneficial using the less-than-full-truckload strategy, especially for expensive products with small physical sizes.
- To include more details about the operation of the cross-dock, for example, the coordination issue and the inventory-holding issue at the cross-dock.

Acknowledgments

This research was supported by a grant from the National Sciences and Engineering Research Council of Canada. The authors are very grateful to one associate editor and two anonymous referees for their detailed and constructive com-

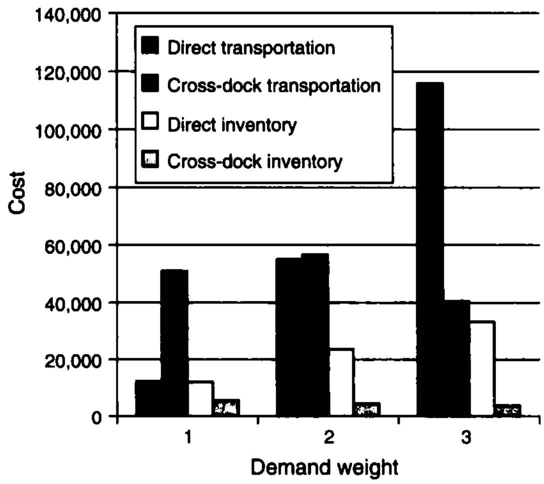

Figure 4 Sensitivity of Cost Distribution to the Change of Demand Weight

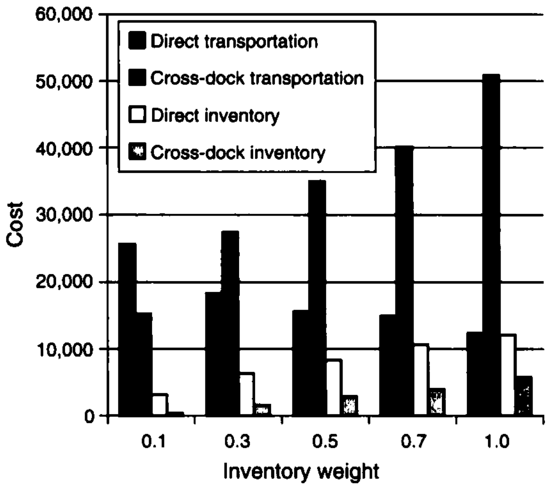

Figure 5 Sensitivity of Cost Distribution to the Change of Inventory Weight

ments and suggestions, which helped them to improve the presentation of the paper significantly.

References

Aykin, T. 1995. Networking policies for hub-and-spoke systems with application to the air transportation system. *Transportation Sci.* **29** 201–221.

Baita, F., W. Ukovich, R. Pesenti, D. Favaretto. 1998. Dynamic routing-and-inventory problems: A review. *Transportation Res. Part B* **32**(8) 585–598.

Beasley, J. E. 1993. Lagrangian relaxation. C. R. Reeves, ed. *Modern Heuristic Techniques for Combinatorial Problems*. John Wiley & Sons, New York, 243–303.

Bertazzi, L., M. G. Speranza. 1999. Models and algorithms for the minimization of inventory and transportation costs: A survey. *Lecture Notes Econom. Math. Systems* **480** 137–157.

Bertazzi, L., M. G. Speranza. 2005. Worst-case analysis of the full load policy in the single link problem. *Internat. J. Production Econom.* **93–94** 217–224.

Blumenfeld, D. E., L. D. Burns, J. D. Diltz, C. F. Daganzo. 1985. Analyzing trade-offs between transportation, inventory and production costs on freight networks. *Transportation Res. Part B* **19**(5) 361–380.

Blumenfeld, D. E., L. D. Burns, C. F. Daganzo, M. C. Frick, R. W. Hall. 1987. Reducing logistics costs at General Motors. *Interfaces* **17**(1) 26–37.

Bookbinder, J. H., N. S. Fox. 1998. Intermodal routing of Canada-Mexico shipments under NAFTA. *Transportation Res. Part E* **34**(4) 289–303.

Burns, L. D., R. W. Hall, D. E. Blumenfeld, C. F. Daganzo. 1985. Distribution strategies that minimize transportation and inventory costs. *Oper. Res.* **33**(3) 469–490.

Campbell, J. F. 1993. One-to-many distribution with transshipments: An analytic model. *Transportation Sci.* **27**(4) 330–340.

Campbell, J. F. 1996. Hub location and the *p*-hub median problem. *Oper. Res.* **44** 923–935.

Cetinkaya, S., C. Y. Lee. 2000. Stock replenishment and shipment scheduling for vendor-managed inventory systems. *Management Sci.* **46**(2) 217–232.

Chan, L. M. A., A. Muriel, Z. J. M. Shen, D. Simchi-Levi, C. P. Teo. 2002. Effective zero-inventory-ordering policies for the single-warehouse multiretailer problem with piecewise linear cost structures. *Management Sci.* **48** 1446–1460.

Daganzo, C. F. 1999. *Logistics Systems Analysis*, 3rd ed. Springer, Heidelberg, Germany.

Lapierre, S. D., A. B. Ruiz, P. Soriano. 2004. Designing distribution networks: Formulations and solution heuristic. *Transportation Sci.* **38**(2) 174–187.

Liu, J. Y., C. L. Li, C. Y. Chan. 2003. Mixed truck delivery systems with both hub-and-spoke and direct shipment. *Transportation Res. Part E* **39** 325–339.

O'Kelly, M. E., D. L. Bryan. 1998. Hub location with flow economies of scale. *Transportation Res. Part B* **32** 605–616.

Popken, D. A. 1994. An algorithm for the multiattribute, multicommodity flow problem with freight consolidation and inventory costs. *Oper. Res.* **42** 274–286.

Shen, Z. J., C. Coullard, M. S. Daskin. 2003. A joint location-inventory model. *Transportation Sci.* **37** 40–55.

Snyder, L. V., M. S. Daskin, C. P. Teo. 2006. The stochastic location model with risk pooling. *Eur. J. Oper. Res.* Forthcoming.

Speranza, M. G., W. Ukovich. 1994. Minimizing transportation and inventory costs for several products on a single link. *Oper. Res.* **42** 879–894.

Zhao, Q. H., S. Y. Wang, K. K. Lai, G. P. Xia. 2004. Model and algorithm of an inventory problem with the consideration of transportation cost. *Comput. Indust. Engrg.* **46** 389–397.