Omega 39 (2011) 64–72

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

Crossdocking distribution networks with setup cost and time window constraint

Hong Ma [a], Zhaowei Miao [b,*], Andrew Lim [a], Brian Rodrigues [c]

[a] Department of Management Sciences, City University of Hong Kong, Kowloon, Hong Kong
[b] School of Management, Xiamen University, Fu Jian 361005, PR China
[c] Lee Kong Chian School of Business, Singapore Management University, Singapore 178899, Singapore

ARTICLE INFO

Article history:
Received 18 March 2008
Accepted 9 March 2010
Available online 16 March 2010

Keywords:
Transportation
Integer program
Heuristics
Crossdocking

ABSTRACT

In this work, we study a new shipment consolidation and transportation problem in crossdocking distribution networks that considers trade-offs between transportation costs, inventory and time scheduling requirements. Transportation costs include time costs, truck setup costs, and the number of trucks used. The model is formulated as an integer program, and shown to be $\mathcal{NP}-$complete in the strong sense. Moreover, a solution approach is provided which consists of two stages. First, a reduced problem is solved for a truckload transportation plan. This is followed by a heuristic solution approach to the remaining less-than-truckload problem. Computational experiments are conducted to test the effectiveness and efficiency of the heuristics. The various cost parameters and time window settings of the distribution network are also discussed.

1. Introduction

As companies seek more profitable supply chain management, there has been a desire to optimize distribution networks to reduce logistics costs. This includes finding the best locations for facilities, minimizing inventory, and minimizing transportation costs. While there is a vast literature available on facility location, crossdocking strategies which minimize inventory by processing goods quickly for reshipment has recently attracted the attention of researchers [1–5]. In this work, we focus on reducing transportation costs in distribution networks with the added objective that transshipment centers achieve the quick reshipment goal of crossdocks. In particular, we study trucking consolidation since trucking accounts for 83% of freight transportation in the US alone [6]. Among companies that have benefited from trucking consolidation, Nabisco Inc., for example, has used consolidation to reduce transportation costs by half, and to bring down inventory levels and improve on delivery [3,7]. In the case of Wal-Mart, crossdocking is often regarded as a key driver of the retailer's superior logistics management [40].

Typically, if a carrier is used, quantity discounts on freight are provided. It is less costly to ship quantities in a full truckload (TL) than to ship partially, using less-than truckload (LTL) shipments. In many situations demand can be less than TL and items delivered may have to wait before reshipment or delivery to customers, thus incurring higher inventory costs at transshipment centers and warehouses. Although shipments can be combined to fill trucks, controlling a distribution system requires detailed knowledge of orders and demand forecasts. The trade-off between transportation and inventory costs is difficult and cannot always be eliminated [8]. To control the trade-off, using crossdocks can help since they allow for TL consolidation from different manufacturers to the customer. In the case of a direct shipment from manufacturer to customer, consolidation can be performed by the manufacturer for goods destined to the customer to reduce transportation and inventory costs.

Manufacturers prefer to control shipout times to achieve better productivity and economies of scale. On the other hand, delivery times must meet customer preferences within short lead times. In making supply meet demand in the supply chain, transportation and inventory cost therefore need to be optimized in coordination with manufacturer and customer time constraints. In this work, we include a time dimension into the distribution network model we study, which takes into account manufacturer and customer time requirements.

The model we study is a global optimization problem which identifies the best choices in distribution network logistics, i.e. trade-offs between transportation costs, inventory and time scheduling requirements at all stages of the supply chain. Transportation costs include time costs, truck setup costs, and the number of trucks used. The model is formulated as an integer program, for which we provide a solution approach which consists of two stages. In the first stage, a reduced problem is

* Corresponding author. Tel.: +86 592 2182618.
E-mail addresses: miaozhaowei@gmail.com, miaozhaowei@xmu.edu.cn (Z. Miao).

0305-0483/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.omega.2010.03.001

solved for a TL transportation plan, and in the second stage, the remaining LTL problem is solved iteratively, using a meta-heuristic, to complete the solution. Computational experiments are conducted to test the algorithms and comparisons made with exact solutions where these are available.

2. Literature review

The model we study has its roots in the classical transshipment problem which is concerned with shipping quantities and routes to be taken through transshipment centers. The problem has been studied in the context of network flow [9] to find the quickest transshipment time. For the general n location transshipment model, Robinson [10], who developed a large LP by discretizing demand, provided heuristic solutions, while Tayur [11] used a gradient-based approach for the problem, also after discretizing demand. Although these studies considered inventory and transshipment costs, they did not address time constraints which are present during the transshipment process, for example, constraints imposed by transportation schedules or time window constraints at supply and demand nodes. When time is a critical factor, distribution through crossdocks, which have become synonymous with rapid consolidation and processing, have been studied. Napolitano [12] has described manufacturing, transportation, distribution, retail and opportunistic crossdocking, all of which have the common feature of consolidation and short cycle times made possible by known pick-up and delivery times. Physical operations which reduce labor costs in LTL crossdocking has been studied by Bartholdi and Gue [13]. The only papers on distribution and system design which include crossdocks are Donaldson et al. [39], Ratliff et al. [14], Gümüs and Bookbinder [3], Li et al. [2] and Lim et al. [15]. In the more recent studies, Gümüs and Bookbinder [3] model location-distribution networks which include crossdock facilities to determine the impact on the supply chain. Both Li et al. [2] and Lim et al. [15] studied transshipment networks where transportation schedules and time constraints at manufacturers and customers are included.

In related work, Grahovac and Chakravarty [16], Herer and Tzur [17], Herer et al. [18], Axsater [19,20] studied the replenishment and inventory policy issues which impact costs in supply and demand management. In other transshipment-inventory models, a frequent assumption is that demand (usually stochastic) which cannot be met from one supply point can be fulfilled through some other point. Work on this subject has been extensive and can be found in, for example, Karmarkar and Patel [21], Karmarkar [22], Tangaras [23], and Rudi et al. [24]. Other related work on the fixed-charge transportation problem which is an extension of the classical transportation problem can be found in Adlakha et al. [25,26] and Kowalski and Lev [27]. Moreover, Waiel et al. [28] presented an interactive fuzzy goal programming approach to determine the preferred compromise solution for the multi-objective transportation problem.

3. Modeling the shipment consolidation problem

Cost parameters: *Transportation costs*. Transportation costs incurred by shippers are typically dependent on variable cost per unit product shipped per unit distance travelled, and on a fixed cost per truck which varies according to the number of trucks used [3,29]. In models which incorporate schedules, a cost per unit product per unit *time* parameter is generally more preferred. This is since fixed pickup and delivery times at manufacturers and customers, respectively, impact transportation times, and ultimately transportation costs. Distance costs can always be included into time costs, whereas time costs do not always arise only from travel distance; for example, additional time costs incurred by driver rest periods, or through recurring congestion delays on specific routes. Hence, the total transportation cost depends on the number of trucks used as well as the total period of time trucks are deployed.

Each truck deployed incurs a setup cost, including administrative cost, driver cost, depreciation cost and so on, which is independent of travel time costs. In this work, we consider two types of routes: direct shipping route (manufacturer-to-customer) and indirect shipping route (manufacturer-to-crossdock and crossdock-to-customer). We make the simplifying assumption that truck setup costs only depend on route type, which means that all direct shipping routes incur the same truck setup cost, so do all indirect shipping routes.

Transshipment center costs. Additional holdover costs is incurred on a per unit basis at transshipment centers, including crossdocks when shipments are delayed, i.e., are not shipped out immediately once received. Since the key motivation for a crossdock (vs. a transshipment center or warehouse) is to achieve zero holdover, a penalty cost per unit product held over in a crossdock is used in the objective function. This principle can be applied to any transshipment center. The approach of introducing holding costs at crossdocks is different from that taken in recent research (see, for example, Gümüs and Bookbinder [3]) on crossdocking distribution networks where holding costs are not incurred and shipments assumed to transit in relatively short times.

Manufacturer/customer inventory costs and time windows. In this study, we assume that manufacturers specify shipping time windows. Likewise, customers specify time windows in which they expect to receive shipments. This allows both manufacturers and customers to optimize their inventory flow by shipping or receiving the product exactly within scheduled times to minimize holdovers. Inventory holdover costs at the manufacturer and customer ends of the supply chain are therefore not assumed in the model (cf. Gümüs and Bookbinder [3]).

Fixed time windows benefit both manufacturers and customers because it can reduce the uncertainty of daily operation and improve the service level. On the other hand, from the constraining point of view, time windows can be required since manufacturers and customers may need to stick to fixed schedules when their serviced routes are made available by transportation vendors [15]. This is especially so for smaller manufacturers who ship through available transportation and do not benefit from made-to-order transportation service.

Scheduling of trucks. Truck pickup and delivery times are, in many cases, determined by trucking providers. On the other hand, manufacturers and customers commonly specify time windows for pickup and delivery. In both cases, finding the best cost transportation plan depends on meeting trucking schedules and time window constraints imposed at the manufacturer, transshipment point and customer (see, for example, Lim et al. [15]). In this work, we assume time windows which determine when trucks are scheduled are set by manufacturers and customers, and time constraints at the transshipment centers are a result of these requirements.

Consolidation. Shipment consolidation (or freight consolidation) seeks the systematic coordination of inventory and transportation decisions at outbound warehouses. We can mention, among many others, the work by Hall [30], Bookbinder and Higginson [31] and Çetinkaya and Lee [32]. In a distribution network studied in this work that includes crossdocks as transshipment points, we assume the consolidation can take place at the manufacturer and/or at transshipment points. In the case of manufacturer, consolidation is initiated by accumulating small orders received from customers before shipment. At transshipment centers, consolidation is performed because inbound shipments are frequently broken down and/or combined before being shipped out to customers. In both cases, consolidation is done to reduce outbound transportation costs.

3.1. The model

In this article, any retailer or cluster of retailers at a location supplied directly from the manufacturers or through transshipment centers is referred to as a "customer" (cf. Gümüs and Bookbinder [3]). We refer to transshipment centers and crossdocks by the collective term "crossdock". We assume that each customer is at a predetermined location $k \in \Delta \equiv \{1,\ldots,m\}$, and demands d_k units of a single product which can be shipped from any manufacturer situated at $i \in \Phi \equiv \{1,\ldots,n\}$, either directly or through a crossdock $j \in \mathbf{X} \equiv \{1,\ldots,l\}$. The value d_k represents the total demand of the retailer or cluster of retailers at location k. Each $i \in \Phi$, has a supply capacity of s_i units of the product, which can only be shipped (released) in the time window $[b_i^r, e_i^r]$ $(i \in \Phi)$ and must be delivered (accepted) within the time window $[b_k^a, e_k^a]$ $(k \in \Delta)$. Also, total supply can meet demand, i.e., $\sum_{i \in \Phi} s_i \geq \sum_{k \in \Delta} d_k$, and, at each crossdock, a holding cost, given by h_j $(j \in \mathbf{X})$ per unit per time, takes effect. This implicitly allows for shipments to be delayed at crossdocks. Fig. 1 illustrates the problem for the multiple-manufacturer, -crossdock and -customer case.

In the model, we take the time horizon to be composed of discrete intervals and the time variable to be discrete (see Ahuja et al. [33], Hoppe and Tardos [9] and Lim et al. [15]). Shipment times can then be represented as non-negative integers. The following parameters are used:

- p setup cost for each truck on arc (i,k) $(i \in \Phi, k \in \Delta)$
- p' setup cost for each truck on arc (i,j) $(i \in \Phi, j \in \mathbf{X})$
- p'' setup cost for each truck on arc (j,k) $(j \in \mathbf{X}, k \in \Delta)$
- h_j inventory holding cost per unit product per unit time in crossdock j $(j \in \mathbf{X})$
- r_{ik} total shipping time on arc (i,k) $(i \in \Phi, k \in \Delta)$
- r'_{ij} total shipping time on arc (i,j) $(i \in \Phi, j \in \mathbf{X})$
- r''_{jk} total shipping time on arc (j,k) $(j \in \mathbf{X}, k \in \Delta)$
- c_{ik} shipping cost per unit product per unit time on segment (i,k) $(i \in \Phi, k \in \Delta)$
- c''_{ij} shipping cost per unit product per unit time on segment (i,j) $(i \in \Phi, j \in \mathbf{X})$
- c''_{jk} shipping cost per unit product per unit time on segment (j,k) $(j \in \mathbf{X}, k \in \Delta)$
- T_i set of feasible shipping time points from manufacturer i, i.e., $T_i = \{b_i^r, b_i^r+1, \ldots, e_i^r\}$ $(i \in \Phi)$
- T_{jk} set of feasible shipping time points from crossdock j to demand point k, i.e., $T_{jk} = \{b_k^a - r''_{jk}, b_k^a - r''_{jk}+1, \ldots, e_k^a - r''_{jk}\}$ $(j \in \mathbf{X}, k \in \Delta)$

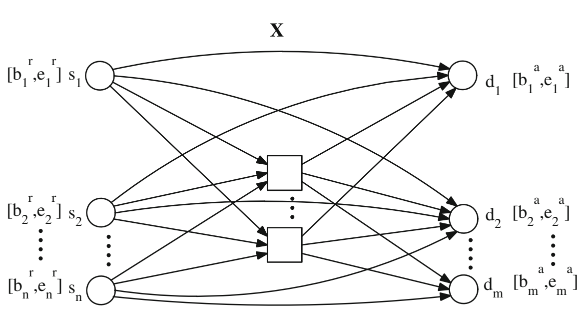

Fig. 1. Distribution network with crossdocks and time windows.

- T'_j set of feasible time points at which shipments can arrive at or depart from crossdock j, i.e. $T'_j = \left\{\min_{i \in \Phi}\{e_i^r + r'_{ij}\},\right.$ $\min_{i \in \Phi}\{e_i^r + r'_{ij}\}+1, \ldots, \max_{i \in \Phi}\{b_i^r + r'_{ij}\}\}(j \in \mathbf{X})$
- Q capacity of each truck
- S set of manufacturer sites
- D set of retailer sites

The decision variables required for the model are:

- x_{ikt} quantity of product shipped on arc (i,k) at time t $(i \in \Phi, k \in \Delta, t \in T_i)$
- x'_{ijt} quantity of product shipped on arc (i,j) at time t $(i \in \Phi, j \in \mathbf{X}, t \in T_i)$
- x''_{jkt} quantity of product shipped on arc (j,k) at time t $(j \in \mathbf{X}, k \in \Delta, t \in T_{jk})$
- v_{ikt} number of trucks used on arc (i,k) at time t $(i \in \Phi, k \in \Delta, t \in T_i)$
- v'_{ijt} number of trucks used on arc (i,j) at time t $(i \in \Phi, j \in \mathbf{X}, t \in T_i)$
- v''_{jkt} number of trucks used on arc (j,k) at time t $(j \in \mathbf{X}, k \in \Delta, t \in T_{jk})$
- I_{jt} quantity of inventory in crossdock j at time t $(j \in \mathbf{X}, t \in T''_j)$

For distribution network that includes crossdocks for transshipment, we typically assume there exists certain level of cooperative relationship between the customers and manufacturers. In industry, this is usually called vertical integration of the supply chain, whereas a task force including both customers and manufacturer representatives seek to achieve some system-wide efficiencies through this partnership [34]. Therefore, the objective in the shipment consolidation problem (SCP) is to minimize the system's total cost including transportation and inventory costs while satisfying time window constraints. Transportation costs include time costs, truck setup costs and the number of trucks deployed. With the above notation, the SCP can be given by the following integer program:

$$\min \sum_{i \in \Phi}\sum_{k \in \Delta}\sum_{t \in T_i}(p v_{ikt} + c_{ik} r_{ik} x_{ikt}) + \sum_{i \in \Phi}\sum_{j \in X}\sum_{t \in T_i}(p' v'_{ijt} + c'_{ij} r'_{ij} x'_{ijt})$$
$$+ \sum_{j \in X}\sum_{k \in \Delta}\sum_{t \in T_{jk}}(p'' v''_{jkt} + c''_{jk} r''_{jk} x''_{jkt}) + \sum_{j \in X}\sum_{t \in T''_j} h_j I_{jt}$$

s.t.

$$\sum_{j \in X}\sum_{t \in T_i} x'_{ijt} + \sum_{k \in \Delta}\sum_{t \in T_i x_{ikt}} \leq s_i \quad (i \in \Phi) \tag{1}$$

$$\sum_{j \in X}\sum_{t \in T_{jk}} x''_{jkt} + \sum_{i \in \Phi}\sum_{t \in T_i} x_{ikt} = d_k \quad (k \in \Delta) \tag{2}$$

$$I_{j,t-1} + \sum_{i \in \Phi} x'_{ij,t-r'_{ij}} - \sum_{k \in \Delta} x''_{jkt} = I_{jt} \quad (j \in \mathbf{X}, t \in T'_j) \tag{3}$$

$$I_{j,t} = 0 \quad (j \in \mathbf{X}, t \in \{\min\{T'_j\}-1, \max\{T'_j\}\}) \tag{4}$$

$$Q(v_{ikt}-1) \leq x_{ikt}, x_{ikt} \leq Q v_{ikt} \quad (i \in \Phi, k \in \Delta, t \in T_i) \tag{5}$$

$$Q(v'_{ijt}-1) \leq x'_{ijt}, x'_{ijt} \leq Q v'_{ijt} \quad (i \in \Phi, j \in \mathbf{X}, t \in T_i) \tag{6}$$

$$Q(v''_{jkt}-1) \leq x''_{jkt}, x''_{jkt} \leq Q v''_{jkt} \quad (j \in \mathbf{X}, k \in \Delta, t \in T_{jk}) \tag{7}$$

All decision variables are nonnegative integers (8)

In the objective function, the first term gives direct transportation costs from manufacturers to customers, including all truck setup costs and time costs; the second and third terms are similar to the first, and represent costs between manufacturers and crossdocks and crossdocks and customers, respectively. The last term represents total holding cost. Constraints (1) ensure the total quantity of the product shipped from manufacturers is no greater than the available supply. Similarly, constraints (2) ensure that total quantity of product received meets demand. Constraints (3) require that, for each crossdock, the inventory at time t is equal to the inventory heldover at time $t-1$ plus the total quantity received at time t minus the quantity shipped out at time t. Constraints (4) are initial and terminal conditions, respectively, of the inventory level at each crossdock. Constraints (5)–(7) ensure that the number of trucks used on any route is a minimum. For practical reasons, we assume the above IP model always has a feasible solution with respect to the time window constraints, i.e. a fall-back shipment plan that circumvents all crossdocks and uses direct shipments from manufacturers to customers is always guaranteed to be feasible.

The following Theorem 1 shows that the problem is $\mathcal{NP}-$complete in the strong sense.

Theorem 1. *The SCP is $\mathcal{NP}$-complete in the strong sense.*

Proof. See Appendix.

In view of the computational complexity of the problem, we provide in the next section, a heuristic approach to its solution.

4. A two-stage heuristic algorithm

In order to find solutions for the SCP, we need to decide the quantity to be shipped, shipment times and routes to be taken. Once TL quantities are decided, LTL shipments at crossdocks need to be consolidated to reduce truck setup costs. The solution approach we take has two components: (1) a full truck load plan (TL Plan), and (2) a less than truck load (LTL Plan). The basic idea is to separate trucks that can be fully loaded at the supply nodes in the beginning, and then search for good solutions that link together various smaller shipments which would have been dispatched in partially loaded trucks. Hence, in the first stage, a solution is found which consists of a TL Plan and an LTL Plan. The TL Plan is found by solving a network flow problem. In the second stage, meta-heuristics are used to improve the LTL Plan.

Stage 1: *Initialization*. A TL Plan is developed, which is used to obtain an initial LTL Plan.

(1) *TL Plan*: A transportation plan is found by solving a network flow problem determined as follows. Split the demand at each customer into a TL part and an LTL part, and remove the LTL part (remainders) from the demand. Next, form the network G with $|\Phi|$ supply nodes and $|\Delta|$ customer nodes, without crossdocks, where the demand is the TL demand. Each arc in G is taken to be the least cost route available for a TL shipment from a supply node to a demand node, which does not violate time window constraints. Each arc represents a direct route from a manufacturer to a customer, or a route through a crossdock.

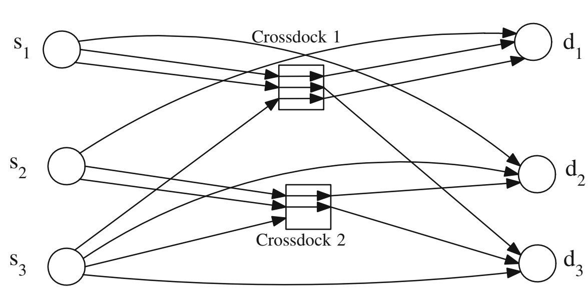

Fig. 2. Example—least cost routes.

A network flow algorithm is utilized to allocate full truck load flow in G to generate the TL Plan. The TL Plan cannot guarantee all demand is satisfied; we leave unsatisfied demand to be satisfied by the LTL Plan.

To illustrate this, Fig. 2 shows a simple example of least cost routes from manufacturers to customers, and Fig. 3 shows a network G constructed from this example. Assume manufacturer nodes, S_1, S_2, S_3, have $s_1=103$, $s_2=10$, $s_3=28$, and customer nodes, D_1, D_2, D_3, have $d_1=25$, $d_2=27$, $d_3=82$, and that the truck capacity $Q=20$. In G, the demand becomes $d'_1=20$, $d'_2=20$, $d'_3=80$. The resulting optimal transportation plan will be to use one truck from S_1 to D_1 and S_1 to D_2, three trucks from S_1 to D_3, and one truck from S_3 to D_3. Hence, the TL Plan for this example can be represented by: {(S_1, *Crossdock*$_1$, D_1, 20), (S_1, *Crossdock*$_2$, D_2, 20), (S_1, *Crossdock*$_1$, D_3,60), (S_3, *Directly*, D_3, 20)}.

Once the TL Plan is decided, it is not altered. Next, we construct an initial LTL Plan.

(2) *LTL Plan*: With a TL Plan in hand, the quantities remaining at supply and demand nodes can be determined. Using the example above, remaining supply and demand is $s'_1=3$, $s'_2=10$, $s'_3=8$, and $d''_1=5$, $d''_2=7$, $d''_3=2$.

Take a *lane* to be an ordered pair of supply and demand nodes and a *route* on a lane to be a crossdock through which shipments pass (possibly empty). Each lane can have multiple routes, i.e., crossdocks through which the product can transit. Take *quantity* to be the number of units of the product shipped on a lane, and let $t\,1$ be the time the product leaves a supply node, and $t\,2$ the time the product leaves the crossdock, if it passes through the crossdock. To overcome the difficulty of time window constraints, the LTL Plan is generated lane by lane, i.e., the LTL Plan is represented by $\{(lane_{ik}, route_{ik}, quantity_{ik}, t1_{ik}, t2_{ik}) : i \in \Phi, k \in \Delta\}$. For example, the element ((1,2),1,13,7,12) obtained in an LTL Plan indicates that we should ship 13 units of the product from supply node 1 to demand node 2 through crossdock 1, where the time the product leaves the supply node 1 is 7 and the time it leaves crossdock 1 is 12.

Since the TL Plan is fixed, the initial solution depends only on the LTL Plan. This is generated through the following three steps:

Step 1. Preallocate: A plan is found greedily as follows. Sort supply and demand in decreasing order of the remaining quantities and then allocate quantities to be shipped from largest demand to the smallest demand. In the example, $s'_1=3$, $s'_2=10$, $s'_3=8$ and $d''_1=5$, $d''_2=7$, $d''_3=2$; hence $s'_2>s'_3>s'_1$ and $d''_2>d''_1>d''_3$. Selecting the largest supply and demand, i.e., $s'_2=10$ and $d''_2=7$, we first satisfy D_2 with S_2, and then satisfy the second largest demand $d''_1=5$, so that D_1 is satisfied with the remaining amount 3 at S_2 and the second largest supply at S_3. Continue in this way until all the demand is satisfied. With the assumption that supply is not less than demand, this can always be done and all lanes allocated with flow.

Step 2. Generate a timetable: With preallocation, once routes are found for each lane, a timetable is constructed taking into

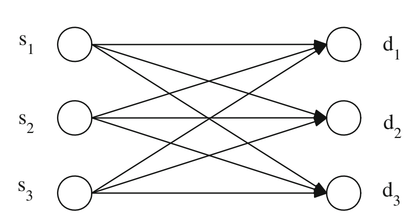

Fig. 3. Network G constructed from the Example.

account the time window constraints. To avoid holding cost, feasible time intervals at each crossdock for each route of each lane are maintained, i.e., for a route of lane (S_i,D_k), the time the product can remain at crossdock j is calculated according to the time windows at S_i and D_k, the shipping time from S_i to j and from j to D_k.

Step 3. Consolidate: Once a timetable has been constructed, consolidation is performed at each supply point and crossdock according to the route and timetable available, and the times t 1 and t 2 are determined to complete the plan to obtain a feasible initial solution.

Fig. 4 shows routes and quantities to be shipped in a plan. In this case, flow from S_1 to D_1 and from S_1 to D_2 can be consolidated at S_1, if the time window constraint is not violated, so that there need only be one truck to transport the product from S_1 to Crossdock 1. Similarly, flow from S_2 to D_2 and flow from S_3 to D_2 can be consolidated at Crossdock 2, and there need only be one truck to transport the product from Crossdock 2 to D_2. There may be a holding cost if the product does not arrive crossdock 2 simultaneously, so that the shipment arriving earlier will have to hold for the later shipment. The flow from S_3 to D_3 does not transit any crossdock.

Stage 2: *Iteration*. Once an initial LTL Plan is found, it can be improved iteratively by using a metaheuristic. We chose a (1) "Squeaky Wheel Optimization" (SWO) [35] and a (2) Genetic Algorithm (GA) [36] since they are representative of techniques available. SWO uses a simple greedy algorithm to construct a solution. The solution is then analyzed to decide on priorities to be used for the greedy algorithm to construct the next solution. This Construct—Analyze—Prioritize loop continues until a limit is reached. Applications of SWO are given in graph coloring problems [35], crane scheduling [37], retail shelf-space optimization [38] for example. In some cases and with less programming effort SWO can achieve comparable or even better results than other general purpose search techniques [35]. GA, on the other hand, is based on the idea of natural selection and usually requires considerable computation for the gradual improvement of solution quality over generations. In this work, each algorithm is used to refine the LTL Plan.

(1) *Using SWO*: In each iteration, the LTL Plan is re-constructed as follows:

Step 1. Calculate transportation costs which satisfy each demand and give the demand with highest cost priority in choosing low cost supply points.

Step 2. Consolidate again at the supply points, where different routes for lanes can be chosen according to their costs. In order to reduce setup costs, give shipments from the same supply point a higher chance to be sent to the same crossdock.

Step 3. To avoid being trapped in a local optimum, randomly reset the priority given to each demand point in choosing manufacturers and routes.

Following this, the transportation plan and timetable are both generated again, similar to how they were obtained in finding an initial solution.

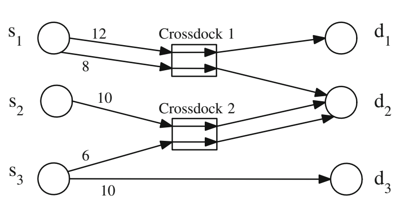

Fig. 4. Consolidation.

The following gives an outline of the SWO algorithm:

Initialize: Get the initial solution by greedy method
for *iter* ← 1 to ♯*iter* do
calculate the cost to satisfy each demand
find the demand point D_m with highest cost to be satisfied
set D_m to have the highest priority to choose the supply points and routes
if D_m has been given the highest priority more than 3 times
randomly choose another demand and set it to have the highest priority
end if
re-generate the LTL Plan according to the new priority
end for
Output: LTL Plan with the lowest cost

(2) *Using GA*: Once a TL Plan is generated, a GA can be used to improve the LTL Plan.

Step 1. Each solution is mapped into a sequence $S=(a_1,a_2,\ldots,a_n)$, where n is the number of customers, and S is a permutation of $1,2,\ldots,n$. The sequence is a customer priority for choosing manufacturers. To begin, sequences are generated randomly.

Step 2. Selection: The population is randomly divided into $M/2$ disjoint pairs, where M is the size of the population. Each pair of individuals are used in the crossover step.

Step 3. Crossover: Each offspring is generated from its parents with equal probability. Let the parents be $P_1=(a_1,a_2,\ldots,a_n)$ and $P_2=(b_1,b_2,\ldots,b_n)$, then, a temporary offspring of P_1 and P_2 is given by $C_{temp}=(rand(a_1,b_1), rand(a_2,b_2),\ldots,rand(a_n,b_n))$ with Prob $(rand(a_1,b_1)=a_1)=$ Prob$(rand(a_1,b_1)=b_1)=0.5$.
After crossover, normalize C_{temp} into a permutation of $1,2,\ldots,n$, by sorting priorities and re-assigning a value from 1 to n to get an offspring C. For example, $C_{temp}=(3,1,2,1)$ becomes $C=(4,2,3,1)$ or $C=(4,1,3,2)$ after normalization.

Step 4. Mutation: A new individual is produced from the offspring by the 2-swap neighborhood with a given probability. For example, $C=(4,1,3,2)$ mutates to $C=(2,1,3,4)$, where the priority of customer 1 and customer 4 is interchanged.

Step 5. Population update: The new individual is inserted into the population replacing one with the highest cost.

The GA algorithm is outlined as follows:

Initialize Population with size M
for *iter* ← 1 to ♯*iter* **do**
for *off* ← 1 to $\frac{M}{2}$ **do**
randomly select parent A and parent B
crossover parent A and parent B to produce offspring C
end for
for each newly-produced individual *indv* **do**
mutate *indv* with probability p_1
end for
select the best $\frac{M}{2}$ individuals from the old generation, and merge them
with the newly-produced $\frac{M}{2}$ individuals to form a new population of size M
discard the rest individuals
update current best solution
end for

5. Computational experiments

The heuristic algorithms are coded in Java and executed on a set of Intel Pentium(R) 2.0 GHz PCs with 512 Mb memory. For comparison, we also test the IP model given in Section 3 in CPLEX, a popular off-the-shelf optimization software package. The purpose of the experiments is to check the effectiveness of the proposed heuristic methods by comparing the results with what CPLEX is able to provide within a reasonable time limit. The test set generation, parameter settings, and detailed computational results are described in the following sections.

5.1. Test data generation

The test cases are generated with three sets of parameters. The actual numbers of the parameters in the test sets are distributed uniformly as Unif $\left[\frac{4}{5}param, \frac{6}{5}param\right]$.

(1) (p,c,h)—(setup cost, transportation cost, inventory holding cost). By setting the ratio of $\frac{p}{c}$, normal setup cost $\left(\frac{1000}{5}\right)$, high setup cost $\left(\frac{2000}{5}\right)$, and very high setup cost $\left(\frac{3000}{5}\right)$ test cases are generated; by setting the ratio for $\frac{p}{h}$, low holding cost $\left(\frac{1000}{50}\right)$ and high holding cost $\left(\frac{1000}{300}\right)$ cases are generated.

(2) (q,cap)—(quantity of demand, truck capacity). By setting the ratio of $\frac{q}{cap}$, we generate normal demand quantity $\left(\frac{30}{20}\right)$, high demand quantity $\left(\frac{50}{20}\right)$, and very high demand quantity $\left(\frac{70}{20}\right)$ cases.

(3) (t,d)—time window length and distances between manufacturers, crossdocks and customers. By setting the ratio of $\frac{t}{d}$, we generate "normal time window" $\left(\frac{40}{30}\right)$ and "narrow time window" $\left(\frac{20}{30}\right)$ cases. A "wide time window" setting is not preferred as it usually simplifies the problem when the surplus "wide time window" constraints can be taken out easily for a given test set before the start of optimization.

5.2. Results from the experiments

The name of a test set "s-d-cd-attr" indicates the number of manufacturers (s), the number of customers (d), the number of crossdocks (cd), and the attribute of the parameters (attr). We first set the attribute of parameters to "normal level" denoted by "(n)". Two heuristics are, respectively, denoted by "NF+SWO" (network flow with SWO) and "NF+GA" (network flow with GA).

Each test set contains five randomly generated test cases. The average resulting costs of the test sets generated by different methods are listed in Tables 1–4 for different parameter settings. The costs in the table are integer numbers, as the insignificant float parts ($\varepsilon < 0.01\%$, $\forall$ costs $> 10\,000$) are conveniently truncated. The terms $T(s)$ correspond to the run times in seconds.

For SWO, ♯ *iter* (maximum iteration) are set to 1000. For GA, M (population) is set to 300. The mutation probability p1 is set to 0.2. The maximum iteration is 10 000 and the termination condition is when the best solution does not improve within 500 iterations. The maximum CPLEX runtime is set to 45 min (2700 s) by "trial and error", because for many test cases CPLEX turned out to run out of memory after around 50 min, which made it quite difficult to summarize or to take average of the computational results.

From Table 1 and Fig. 5, we find that CPLEX is not able to solve any of the sets to optimality within a runtime limit of 45 min, but it performs quite well for small test sets, as compared with the two heuristics. As the size of test sets increases, performance of NF+SWO is close to CPLEX and performance of NF+GA surpasses CPLEX gradually. While NF+SWO is a faster heuristic method in terms of runtime, NF+GA generates better results on average. Both NF+SWO and NF+GA complete within a few minutes, if not in seconds. We then take a close look at the best solutions, i.e. the CPLEX solutions for the first four small test sets and NF+GA solutions for the large sets. We find out that on average, the number of trucks arriving at or departing from the crossdocks accounts for 47–65% of the total number of trucks in the schedules, which we reckon can be a good measurement for the crossdock utilization. The percentage of the transportation cost and holding cost related to crossdocks varies significantly between 39% and 79% of the total cost. While it is difficult to determine a total cost structure, the importance of crossdocks is proved by the busy truck activities.

5.2.1. Using different parameters

High and extremely high setup cost: Computational results for "high (hs)" and "extremely high (es)" setup costs are listed in Table 2. Different cost levels in truck setup can arise through the use of different common carriers or customer's own dedicated truck fleet such as the Wal-Mart case for example. From Fig. 6, we see that both heuristics perform better than CPLEX. This implies that increasing setup costs can have an effect that results in an expanded IP solution space, which in turn has a negative influence on the "branch & bound (cut)" strategy CPLEX employs. Correspondingly, the two heuristic approaches are less affected. This also suggests that to better illustrate effects of the "higher" (or "more constrained") parameter settings, further experiments be focused on small test sets, such as those with 10 supply nodes (manufacturers) and less than four demand nodes (customers), for which CPLEX performs well. When we analyze the NF+GA solutions, which are the best in quality among three methods, we find the percentage of truck operations related to crossdocks decreases slightly. The number of trucks arriving at or departing from the crossdocks, in the high and extremely high setup cost setting, accounts for 45–62% of the total truck usage. This small change can be explained in the NF+GA procedure because while the TL plan is not affected by the truck setup cost, in the LTL plan, direct shipping has its advantage in using a smaller number of trucks unless the truck setup cost at the crossdocks is different from a manufacturer. Since the experiments are based on

Table 1
Results for normal parameters.

Test set	CPLEX	LB	Gap (%)	T (s)	NF+SWO	Gap (%)	T (s)	NF+GA	Gap (%)	T (s)
10-20-3-n	214371	201 070	6.6	2700	225 658	12.2	1.7	224 814	10.6	16.5
10-40-3-n	455933	418 897	8.8	2700	461 525	10.2	4.5	460 037	9.8	19.0
10-60-4-n	688435	630 463	9.2	2700	703 416	11.6	10.1	701 519	11.2	37.6
10-80-4-n	1352060	1 189 643	13.7	2700	1 408 208	18.4	15.3	1 428 825	20.1	49.9
20-60-6-n	1 052 382	896 869	17.3	2700	1 093 639	21.9	24.5	1045889	16.6	68.3
20-60-8-n	1 095 538	926 347	18.3	2700	1 110 580	19.9	28.7	1056577	14.1	71.4
20-80-6-n	1 389 823	1 219 187	14.0	2700	1 433 150	17.5	35.2	1381104	13.3	115.3
20-80-8-n	1 556 685	1 355 616	14.8	2700	1 595 046	17.6	41.8	1530916	12.9	133.5

Table 2
Results for high and extremely high setup cost.

Test set	CPLEX	LB	Gap (%)	*T* (s)	NF+SWO	Gap (%)	*T* (s)	NF+GA	Gap (%)	*T* (s)
10-40-3-hs	566 539	509 928	11.1	2700	560 765	10.0	4.1	559446	9.7	22.9
10-60-4-hs	921 802	800 943	15.1	2700	886 914	10.7	13.1	884728	10.5	39.2
10-80-4-hs	1 328 027	1 160 959	14.4	2700	1 280 280	10.3	18.4	1277915	10.1	51.4
20-80-6-hs	2 128 752	1 768 865	20.3	2700	2 130 497	20.4	37.4	2112374	19.4	118.6
10-40-3-es	766 911	679 050	12.9	2700	762 911	12.3	7.1	762219	12.2	22.4
10-60-4-es	1 149 195	1 004 250	14.4	2700	1115198	10.9	15.7	1115198	10.9	39.1
10-80-4-es	1 586 560	1 391 935	14.0	2700	1 524 003	9.5	21.2	1523557	9.4	55.1
20-80-6-es	3 090 487	2 535 703	21.9	2700	3 107 764	22.5	38.2	3014955	18.9	122.7

Table 3
Results for high holding cost and narrow time windows.

Test set	CPLEX	LB	Gap (%)	*T* (s)	NF+SWO	Gap (%)	*T* (s)	NF+GA	Gap (%)	*T* (s)
10-20-3-h	231120	215 530	7.2	2700	234 409	8.7	3.6	233 925	8.5	17.7
10-40-3-h	470 522	409 870	14.8	2700	4 491 752	9.6	11.1	448214	9.3	19.8
10-50-3-h	591 606	515 379	14.8	2700	565 993	9.8	16.4	564226	9.5	24.1
10-60-4-h	717 343	632 305	13.4	2700	683 283	8.1	17.1	683109	8.0	43.8
10-20-3-ntw	228 021	204 396	11.6	2700	224 034	9.6	14.7	223658	9.4	19.5
10-40-3-ntw	471 610	413 867	14.0	2700	453 807	9.7	18.1	452584	9.4	28.9
10-50-3-ntw	554 405	502 480	10.3	2700	549 425	9.3	20.3	548192	9.1	30.3
10-60-4-ntw	749 092	641 115	16.8	2700	702 390	9.6	17.6	701428	9.4	44.8

Table 4
Results for large and extremely large demand.

Test set	CPLEX	LB	Gap (%)	*T* (s)	NF+SWO	Gap (%)	*T* (s)	NF+GA	Gap (%)	*T* (s)
10-20-3-ld	557 810	492 838	13.2	2700	538 229	9.2	3.9	537498	9.0	21.5
10-40-3-ld	1 023 318	908 388	12.7	2700	1 006 463	10.8	5.1	1005588	10.7	24.1
10-50-3-ld	1 337 733	1 151 537	16.2	2700	1 277 052	10.9	11.8	1276293	10.8	28.5
10-60-4-ld	1 597 592	1 394 411	14.6	2700	1 531 300	9.8	15.6	1530649	9.7	40.4
10-20-3-ed	773 528	685 227	12.9	2700	749 569	9.4	7.7	749560	9.4	23.5
10-40-3-ed	1503290	1 369 771	9.7	2700	1513199	10.5	9.1	1 512 328	10.4	24.7
10-50-3-ed	2 067 313	1 780 236	16.1	2700	1966829	10.4	12.0	1966829	10.4	35.6
10-60-4-ed	2 412 512	2 068 616	16.6	2700	2 306 647	11.5	14.7	2305570	11.4	39.6

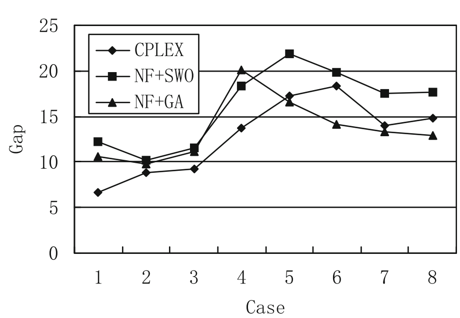

Fig. 5. Gap of results for normal parameters.

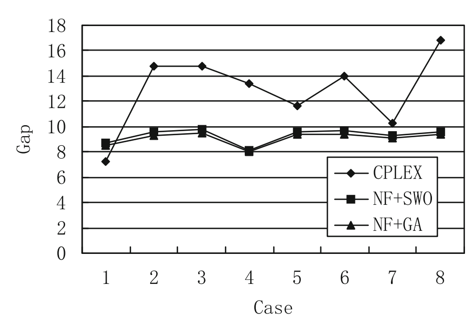

Fig. 7. Gap of results for high holding cost and narrow time windows.

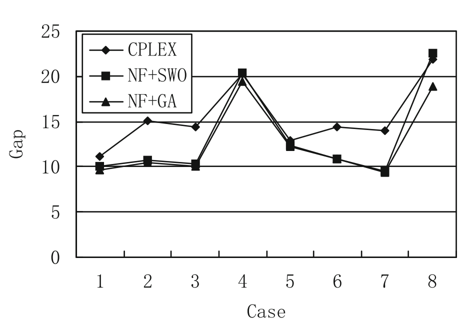

Fig. 6. Gap of results for high and extremely high setup cost.

randomly generated data sets with a few other parameters, such as time windows, direct comparison of the cost structure in different setup costs is not feasible although we reckon that the holding cost at the crossdocks will decline with the number of arriving trucks. For multi-commodity problems where a single truck at the crossdock can carry different type of shipments up to full truck load, the above findings may no longer hold since greater savings on truck setup can be achieved by consolidating shipments of multiple types at crossdocks.

High holding cost: Table 3 shows the heuristics in general provide better solutions as compared with CPLEX in the “high holding cost (h)” setting, and Fig. 7 plots the gap of different methods.

Narrow time windows: Table 3 also shows the heuristics outperform CPLEX in test sets with the "narrow time windows (ntw)" setting. Comparing with high truck setup cost, high holding cost and narrow time windows both have a stronger and immediate effect on the truck setup at crossdocks. While a higher holding cost works against an extended shipment consolidation process in which different shipments are reorganized at crossdocks, narrow time windows force a tighter schedule especially on the manufacturer and customer nodes, resulting in more direct shipments due to solution feasibility. Since stock outs at the customer nodes are not allowed in the model, a solution that contains routes with less travel times to meet narrow time windows has an advantage. We observe from the best solutions generated by NF+GA that on average the number of trucks operating at the crossdocks accounts for 25–53% of the total in these two settings, showing a significant reduction in crossdock utilization.

Large and extremely large demand: Table 4 shows the results when test set parameter is set to "large demand (ld)" and "extremely large demand (ed)". From the table and Fig. 8, we find that both heuristic approaches give more competitive results than CPLEX does, as compared to the normal demand setting, which implies that larger demand settings have resulted in harder problem instances. This is mainly due to the increase of number of trucks required in transportation and more room given for reorganization work at crossdocks. With regard to the percentage of trucks at the crossdocks, we have not found evident changes.

In summary, the two heuristic approaches we propose are shown to be efficient in terms of both runtime and solution quality. The results are very competitive, as compared with what the popular commercial solver CPLEX has provided.

6. Conclusion and future work

In this work, we study a new shipment consolidation problem in distribution networks that include crossdocks as well as time windows in manufacturers and customers, whereas a single product can be shipped directly or through crossdocks. The problem examines trade-offs among transportation costs, inventory holding costs, and scheduling requirements. The problem is proved to be $\mathcal{NP}$-complete in the strong sense, and then formulated as an Integer Program (IP) model. A two-stage heuristic framework, which combines a network flow algorithm and a meta-heuristic, is proposed. We develop a Squeaky Wheel Optimization (SWO) heuristic and a Genetic Algorithm (GA) as the second stage algorithm for the consolidation of LTL shipments. The idea behind the heuristics is to split the demand into TL and LTL components, and solve the TL component as a standard network flow problem. The key point in the second stage is to find a way to represent lanes from manufacturers to customers that are feasible with respect to the time windows constraints. The problem can then be reduced as

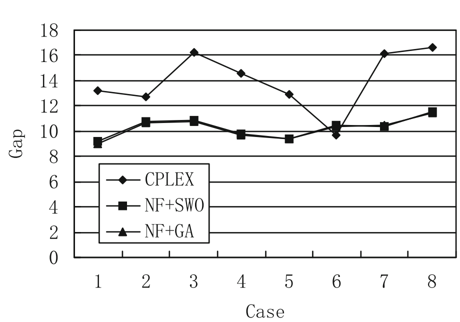

Fig. 8. Gap of results for large and extremely large demand.

the time constraints are removed in subsequent procedures. Computational experiments are carried out to test the two-stage heuristic approaches as well as the IP model in CPLEX. The heuristic approaches are shown to be efficient in terms of both runtime and solution quality, and can be conveniently applied in situations when the costly CPLEX suit is unavailable on the spot, or when optimal numerical solutions are not rigorously required in a practical context of supply chain management.

Future work can be carried out in a few directions. One of them is to conduct more extensive computational testing to determine the cost structure of the distribution network as well as the average shipment "delay time" at crossdocks under different cost and time window parameter settings. The current time-expanded model has a large number of decision variables partly due to the discrete time points that map different shipping time windows. Work can be done to design a more efficient solving procedure that simplifies the problem based on the given time window limits. Infeasible time windows at the manufacturers and crossdocks can first be identified, and then related shipment decision variables are picked out and removed, which will reduce the number of idle variables and hopefully lead to performance boosts on exact methods.

It is also worth to note that the feasible shipping time windows in this work are defined as "hard time windows", i.e. predetermined and fixed. Thus, the solutions are likely to contain some less efficient shipping schedules just in order to meet these fixed time windows. More work can be done to consider the soft time window scenario, i.e. time windows that can be violated at a cost. In this scenario, different penalty costs are incurred when the time windows are missed on the early or late sides. We may also consider a scenario where an upper limit is set for the numbers of trucks leaving a particular manufacturer at the same feasible time point to simulate a realistic road capacity or congestion issues. At last, the current problem can also be extended to the multi-commodity consolidation problem with time windows, in which various type of goods or freight are considered in a given supply chain transshipment network.

Acknowledgements

The authors would like to express their thanks to the anonymous referees whose comments and suggestions helped improve the work. This research is supported by a grant from City University of Hong Kong (Project no. 6341009) and grants from the National Natural Science Foundation of China (NSFC) (Project nos. 70802052 and 70971111) and the National Ministry of Education of China (Humanities and Social Science Project no. 07JC630047—"Research on Green Logistics System Design and Optimization Strategies").

Appendix

Proof of Theorem 1. We provide a reduction of the strongly $\mathcal{NP}$-complete **3- PARTITION** problem: Given positive integers, w, D and $\Gamma=\{1,2,\ldots,3w\}$ with a positive integer values $\gamma(i)$ where, for each $i\in\Gamma$, $\sum_{i\in\Gamma}\gamma(i)=wD$ and $D/4<\gamma(i)<D/2$ for $i\in\Gamma$, can Γ be partitioned into w disjoint sets $\Gamma_1,\Gamma_2,\ldots,\Gamma_w$ such that $|\Gamma_j|=3$ and $\sum_{i\in\Gamma_j}\gamma(i)=D$ for $j=1,\ldots,w$? From an arbitrary instance of **3-PARTITION**, we consider a polynomial reduction to an instance of the SCP and ask if there exists a feasible solution whose objective value is no greater than $4w$. For $3w$ manufacturers given in Φ, and w customers $\Delta=\{1,2,\ldots,w\}$, let s_i be the supply and $s_i=\gamma(i)$ for $i\in\Phi$, while for each $j\in\Delta$, let D be the demand and $[j,j]$ be the time window; relax the time windows of manufacturers. Let Q and setup cost p' (p'') of each truck be equal to D and 1,

respectively, and set p to be a large number. Exactly one crossdock, χ, say, with holding cost 1 per unit product per time exists linking manufacturers with customers, for which transportation cost is zero for all arcs. We now show that a feasible schedule exists whose objective value is no greater than 4w if and only if the **3-PARTITION** has a feasible solution. On one hand, if **3-PARTITION** has a feasible solution $\Gamma_1,\ldots,\Gamma_w$,we can ship all goods provided by manufacturers in Φ_i (letting $\Phi_i=\Gamma_i$) to χ at time j, which satisfies the demand D for demand j in its time window $[j,j]$, where $j=1,2,\ldots,w$. It is easy to verify that such a schedule is feasible and total cost is 4w. On the other hand, if a feasible schedule exists with objective no greater than 4w, then it is optimal since it is easy to prove that 4w is the lower bound of our instance. The optimal solution with this objective must satisfy the following conditions: (1) there is no direct positive flow between manufacturers and customers; (2) no more than one truck is used on each arc; (3) there is no inventory in crossdock. We can then construct a partition by setting Φ_j to be the subset of $i \in \Phi$ where a positive flow is shipped from i to χ at time j, for $1 \le j \le w$. Because of conditions (1)–(3) and the demands are D at time j, we have $\sum_{i \in \Phi_j} s_i = D$. Since $D/4 < s(i) < D/2$ for $i \in \Phi$, we have $|\Phi_j| = 3$. Hence, $\Phi_1,\ldots,\Phi_w$ is a feasible partition for the instance of **3-PARTITION** and this completes the proof. □

References

[1] Brockmann T. 21 warehousing trends in the 21st Century. IIE Solutions 1999;31(7):36–40.
[2] Li Y, Lim A, Rodrigues B. Crossdocking: JIT scheduling with time windows. Journal of the Operational Research Society 2004;55(12):1342–51.
[3] Gümüs M, Bookbinder JH. Cross-docking and its implication in location-distribution systems. Journal of Business Logistics 2004;25(2):199–228.
[4] Boysen N, Fliedner M. Cross dock scheduling: classification, literature review and research agenda. Omega 2009; doi:10.1016/j.omega.2009.10.008.
[5] Tang SL, Yan H. Pre-distribution vs. post-distribution for cross-docking with transshipments. Omega 2010;38:192–202.
[6] Wilson RA. Transportation in America, 19th ed. Washington, USA: Eno Transportation Foundation; 2002.
[7] Quinn FJ. The payoff!. Logistics Management 1997;36(12):37–41.
[8] Simchi-Levi D, Kaminsky P, Simchi-Levi E. Designing and managing the supply chain, 2nd ed.. Boston: McGraw-Hill—Irwin Publishers; 2003.
[9] Hoppe B, Tardos E. The quickest transshipment problem. In: Proceedings of the sixth annual ACM-SIAM symposium on discrete algorithms, San Francisco, California, United States, 1995.
[10] Robinson LW. Optimal and approximate policies in multiperiod, multilocation inventory models with transshipments. Operations Research 1990;38:278–95.
[11] Tayur S. Computing optimal stock levels for common components in an assembly system. Working paper, Carnegie-Mellon University, USA, 1995.
[12] Napolitano M. Making the move to crossdocking. The WERC-sheet, Warehousing Education and Research Council; 2000.
[13] Bartholdi III JJ, Gue KR. Reducing labor costs in an LTL crossdocking terminal. Operations Research 2000;48(6):823–32.
[14] Ratliff DH, van de Vate J, Zhang M. Network design for load-driven cross-docking systems. Research report, Georgia Institute of Technology, 1999.
[15] Lim A, Miao Z, Rodrigues B, Xu Z. Transshipment through crossdocks with inventory and time windows. Naval Research Logistics Quarterly 2005;52(8):724–33.
[16] Grahovac J, Chakravarty A. Sharing and lateral transshipment of inventory in a supply chain with expensive low-demand items. Management Science 2001;47(4):579–94.
[17] Herer YT, Tzur M. The dynamic transshipment problem. Naval Research Logistics Quarterly 2001;48:386–408.
[18] Herer YT, Tzur M, Yücesan E. Transshipments: an emerging inventory recourse to achieve supply chain leagility. International Journal of Production Economics 2002;80:201–12.
[19] Axsater S. A new decision rule for lateral transshipments in inventory systems. Management Science 2003;49:1168–79.
[20] Axsater S. Evaluation of unidirectional lateral transshipments and substitutions in inventory systems. European Journal of Operational Research 2003;149:438–47.
[21] Karmarkar US, Patel NR. The one-period n-location distribution problem. Naval Research Logistics Quarterly 1977;24:559–75.
[22] Kamarkar US. The multilocation multiperiod inventory problem: bounds and approximations. Management Science 1987;33(1):86–94.
[23] Tangaras G. Effects of pooling on the optimization and service levels of two-location inventory systems. IIE Transactions 1989;21:250–7.
[24] Rudi N, Kapur S, Pyke DF. A two-location inventory model with transshipment and local decision making. Management Science 2001;47:1668–80.
[25] Adlakha V, Kowalski K. A simple heuristic for solving small fixed-charge transportation problems. Omega 2003;31:205–11.
[26] Adlakha V, Kowalski K, Vemuganti RR, Lev B. More-for-less algorithm for fixed-charge transportation problems. Omega 2007;35:116–27.
[27] Kowalski K, Lev B. On step fixed-charge transportation problem. Omega 2008;36:913–7.
[28] Waiel F, El-Wahed A, Lee SM. Interactive fuzzy goal programming for multi-objective transportation problems. Omega 2006;34:158–66.
[29] Daganzo CF. Physical distribution from warehouse: vehicle coverage and inventory levels. Transportation Research—Part B 1985;19B(5):397–407.
[30] Hall RW. Consolidation strategy: inventory, vehicles and terminals. Journal of Business Logistics 1987;8(2):57–73.
[31] Bookbinder JH, Higginson JK. Probabilistic modeling of freight consolidation by private carriage. Transportation Research Part E: Logistics and Transportation Review 2002;38:305–18.
[32] Çetinkaya S, Lee CY. Optimal outbound dispatch policies: modeling inventory and cargo capacity. Naval Research Logistics 2002;49:531–56.
[33] Ahuja RK, Magnanti TL, Orlin JB. Network flows: theory, algorithms, and applications. New Jersey: Prentice-Hall; 1993.
[34] Buzzell RD, Ortmeyer G. Channel partnerships streamline distribution. Sloan Management Review 1995;36:85.
[35] Joslin DE, Clements DP. Squeaky wheel optimization. Journal of Artificial Intelligence 1999;10:353–73.
[36] Glover F, Kochenberger GA. Handbook of metaheuristics. Boston, Dordrecht, London: Kluwer Academic Press; 2003.
[37] Lim A, Rodrigues B, Xiao F, Zhu Y. Crane scheduling with spatial constraints. Naval Research Logistics 2004;51(3):386–406.
[38] Lim A, Rodrigues B, Zhang X. Metaheuristics with local search techniques for retail shelf-space optimization. Management Science 2004;50(1):117–31.
[39] Donaldson H, Johnson EL, Ratliff HD, Zhang M. Schedule-driven crossdocking networks. Research report, Georgia Institute of Technology, 1998.
[40] Hammer M. Deep change: how operational innovation can transform your company. Harvard Business Review 2004;82(4):84–93.