Synchronization of inventory and transportation under flexible vehicle constraint: A heuristics approach using sliding windows and hierarchical tree structure

Zhendong Pan \textsuperscript{a}, Jiafu Tang \textsuperscript{a,\*}, Richard Y.K. Fung \textsuperscript{b}

\textsuperscript{a} Department of Systems Engineering, Key Laboratory of Integrated Automation of Process Industry of MOE, Northeastern University, Shenyang 110004, China
\textsuperscript{b} Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong, China

Received 30 July 2006; accepted 8 October 2007
Available online 17 October 2007

Abstract

This paper investigates the integrated inventory and transportation planning under flexible vehicle constraint. To offer better services at lower prices, more and more companies turn to outsource transportation functions to other professional service providers, namely 3rd party logistics companies. Under these vehicle rental arrangements, the number of vehicles is a decision variable instead of a fixed number, and the transportation cost includes not only the delivery cost but also the cost of vehicle rental that is proportional to the number of vehicles rented in a given planning horizon. In this paper, the problem is formulated as a mixed integer programming problem. A heuristic algorithm is developed, in which sliding windows are applied to approximate the problem by repeatedly solving a series of overlapping short-term subproblems, and a hierarchical tree structure is used to evaluate the closeness of different groups of retailers. Numerical experiments show that a better tradeoff between the inventory cost and transportation cost can be achieved through the proposed heuristic algorithm.

Keywords: 3rd Party logistics; Flexible vehicle constraint; Mixed integer program; Sliding window; Hierarchical tree

1. Introduction

Today’s market, where products are designed, manufactured, and distributed in a speed faster than ever seen before, are forcing companies to continuously improve their supply chain management (SCM). Many models, algorithms, and decision tools are developed for every echelon of the supply chain, and contribute a lot to the management of many companies’ operations both economically and efficiently [17]. However, since logistics constitutes such a large percentage of many companies’ expenses, further efforts on SCM are still appreciated, and two popular methodologies recently proposed and widely applied in industries are integrated SCM and 3rd party logistic (3PL).

Traditionally, synchronization of different echelons is accomplished in a sequential way, in the sense that outputs of the upper echelon are regarded as inputs of the latter echelon. This way cannot yield an optimal plan for a company of which operations cover more than one echelon of a supply chain. This problem motivates researchers to come up with the idea of integrated SCM, which is to optimize the supply chain as a whole and consider the planning of different echelons simultaneously.
The integrated SCM provides an important source of cost savings for companies’ operation management, particularly for inventory and transportation, which are the two most common operations of many companies.

Nowadays, the role of the 3PL companies becomes more and more important in industry. The activities of 3PL companies can be seen almost everywhere in the supply chain. By properly purchasing the cheap and reliable services provided by 3PL companies, many companies increase their profits and improve customer relationship in the meantime. Among all the fields that 3PL companies have entered, transportation is one of the firsts. With the anticipation of 3PL transportation companies, i.e. professional transportation enterprises, the situations are more complex. Dynamic and flexible number of vehicles rather than constant are available to cater for quickly changed market demands, and the number of vehicles to be rented is thus viewed as an operational decision simultaneously with inventory and transportation plan [20]. However, most of the former researches are based on the assumption that vehicles are maintained by the companies themselves. Hence, these models assume that the number of vehicles is constant and hence the cost in renting vehicles is neglected in the objective function on operational level decision making.

Due to the huge needs for effective solution methodologies of integrated SCM in industries, many researchers have dedicated so much effort in this area. However, because of the complexity of these problems, optimal solutions are impossible to obtain. Within integrated SCM, different considerations and levels of analyses with proper simplification have been proposed along with heuristic solution approaches, and unified classifications are found in the paper [3,6,8,17]. Here, two categories are identified according to their replenishment modes: frequency-domain models and time-domain models. This classification can include most of known literatures and adopted by Baita et al. [3]. Comprehensive reviews can also be found in [6,8,17].

The frequency-domain models are models of which decisions of replenishment are made in a frequency-based way, and these decisions include the frequency and quantity of replenishment. Under the situation of one warehouse and multi-retailers, Federgruen and Zipkin [12] consider a one-period problem in which the amount of product at the depot is limited. Their work is the first to integrate the problems of product allocation and vehicle routing into a single model. They propose a heuristic solution method which decomposes the main problem into a non-linear inventory allocation subproblem and a number of traveling salesmen subproblems (one for each vehicle considered). Burns et al. [7] consider an infinite horizon problem and develop an analytical method to minimize the overall cost. They compare two different distribution strategies: the direct shipping (i.e. shipping separate loads to each customer) and the peddling shipping (i.e. dispatching vehicles that deliver items to more than one customer per load). For the direct shipping, they obtain an Economic Order Quantity (EOQ) type of solution by trading off inventory and transportation cost. If the shipment size obtained from the solution is greater than the capacity of the truck then a full-truck load is scheduled. For the peddling strategy, a full-truck load is the optimal shipment size. Anily and Federgruen [1] consider a problem very similar in structure to that of Burns et al. [7]. They restrict themselves to a class of strategies in which a collection of regions (set of retailers) is specified which cover all outlets and they call it replenishment strategies $\phi$. The upper and lower bounds for the system-wide long run average cost are derived, and these bounds are asymptotically optimal under weak probabilistic conditions. A solution procedure of which computational requirements grow roughly linearly with the number of locations considered is developed. Anily and Federgruen [2] extend their former work [1] to the case where central inventories may be kept in the warehouse. Viswanathan and Mathur [22] consider the same problem as Anily and Federgruen [1] with the generalization of multiple products in the system. They develop a heuristic algorithm for a joint replenishment problem to obtain a stationary nested joint replenishment policy.

All the above works are based on the famous EOQ model. A major drawback of EOQ-based models is that they provide headways that may be non-integer (in fact, non-rational): it would be hard to make deliveries every $\sqrt{5}$ days. Some researchers, for instance Hall [14], acknowledge that this is an impractical result. Based on the assumption that shipments done on a set of specific frequencies (e.g. 1, 3, 10 days compose a set) may be more appropriate and executable by the transportation company [4,5,18,19]. Speranza and Ukovich [19] consider the product shipping strategy to determine shipping frequencies at which each product have to be shipped in a way that the sum of transportation cost and inventory cost is minimized. The cases of single frequency (SF) and multiple frequencies (MF) and the ways of time consolidation (TC) and frequency consolidation (FC) are discussed. They further discuss MFFC from the theoretical and the computational point of view and pointed out that the formulation is NP-hard from the perspective of computational complexity [22]. This problem is extended by Bertazzi et al. [4] from single destination to multiple destinations under several given shipping frequencies and is solved by decomposing into several subproblems each with a heuristic algorithm. Bertazzi and Speranza [5] consider design of periodic shipping strategy to minimize the total cost of transportation and inventory in a network with one origin, some intermediates and one destination with given frequencies. As in Yano and Gerchak [23], they find that shipping less-than-full truckloads may lead to improved policies, and they consider the following two cases for shipping frequencies: (1) each product is assigned a single frequency, and (2) each product can be assigned to more than one frequency (order splitting).

The other kind of models is time-domain models. In these models, decisions are taken at the beginning of each period (e.g. every day or week), knowing the state of the system (i.e. the inventory levels). This is a closed loop approach in the sense that there is a feedback of the decisions taken in one period, which affect the decisions of the following period.
Time-domain model is firstly introduced by Dror and Ball [9], Dror and Levy [10], Dror and Trudeau [11], and Trudeau and Dror [21]. In each time interval, only the customers who will reach their safety stock level during this interval are serviced. A single item has to be delivered from one depot to many customers, whose demand is different in each period and is deterministic in Dror and Ball [9] and Dror and Levy [10], stochastic in Trudeau and Dror [21], and both deterministic and stochastic in Dror and Trudeau [11]. They consider a fixed number of identical trucks. Dror and Levy [10], and Dror and Trudeau [11] present heuristic algorithms for solving the problem. They focus on the maximization of operational efficiency (average number of units delivered in 1 h of operation) and the minimization of the average number of stockouts in one period.

Jaillet et al. [16] discuss an extension of this idea. They take a rolling horizon approach to the problem by determining a schedule for two weeks, but only implementing the first week. At the beginning of the next week, the problem will be solved again for the next two weeks. The system includes not only a central depot and customers to be replenished to prevent stockouts, but also several satellite facilities. Campbell [8] adopt a similar method to solve an inventory routing problem. Fleischmann [13] considers transportation of several products on a single link when shipment is conducted only at discrete times. It aims to determine the timing and the quantities of the shipments and the inventory level on a link in a specified planning horizon to minimize total cost of transportation and inventory.

This paper investigates the problem of synchronization of inventory and transportation in a system where there are one warehouse and multiple retailers facing a constant and retailer-specific demand for a single product. In addition, the number of available vehicles is flexible and needed to be decided simultaneously with the inventory and transportation plan. A heuristic algorithm using sliding windows and hierarchical tree is developed to solve the problem.

The contributions of this paper include:

1. investigate vehicle-rental the transportation where vehicle are outsourced from the 3rd party logistic companies,
2. provide a “sliding windows” method to approximate the long-term problem by a series of short-term overlapping subproblems,
3. propose a hierarchical tree to exploit the geographical information of retailers, and
4. examine the benefits obtained from synchronizing inventory and transportation plan.

The rest of this paper is organized as follows. In Section 2, the characteristics of the problem are analyzed and a mixed integer programming (MIP) mathematical model is formulated. How to decide the inventory and transportation plan are discussed in Section 3. A heuristic algorithm using sliding windows and hierarchical tree is developed in Section 4. Numerical experiments are conducted in Sections 5 and 6 concludes this paper.

2. Mathematical model for integrated inventory and transportation problem in 3rd logistic environment

In a distribution network, a certain kind of product is shipped from a central warehouse to several geographically dispersed retailers over a finite planning horizon. The planning horizon consists of a number of discrete periods which may be months if the planning horizon is a year, or days if the planning horizon is a week. Each retailer faces a constant but retailer-specific demand. Stocks are kept at the retailers but not at the warehouse, which means that the distribution system is “coupled” [1]. There is no limit on the inventory of retailers; and backlogging is allowed, but a penalty must be charged.

Moreover, this paper considers a situation completely different from former models that assume the number of vehicles is constant. In practical applications, if the vehicles are maintained by the warehouse itself, on the one hand, it may bring great expenses due to redundant vehicles when the market demand is largely lower; on the other hand, it would bring penalty when the market demand reaches out of the number of vehicles. Therefore, flexible and dynamic number of vehicles is more suitable for warehouse to respond to the variation of market demands with less expenses and higher customer satisfaction. This is the motivation for companies to subcontract their transportation operations to 3PL transportation companies. At this time, transportation plan is made by the companies, and the actual distribution is undertaken by contracted 3PL transportation companies. The decision maker must ensure that there are always enough vehicles to fulfill the transportation plan over the planning horizon. Therefore, it is reasonable to pay an additional rental fee (denoted by $C_{\text{rent}}$) to the professional transportation enterprise for reservation of a fixed number of vehicles so that there would not be “shortage” for vehicles. This cost is called vehicle rental cost.

In the follows, deliveries are assumed to be made by identical trucks (in terms of capacity). Due to the participation of 3PL transportation companies, the transportation cost is made up of three parts:

1. Travel cost, which is proportional to the distance that the trucks travel.
2. Dispatch cost, the setup cost of dispatching trucks.
3. Truck rental cost.
Obviously, the structure of the transportation cost is different from those of former research. This difference, as shown later, can affect the transportation plan.

For simplicity, the deliveries are assumed “the first thing in the morning”. In addition, split delivery is not allowed in this paper, which means each retailer can be served no more than once in each period.

Before the analysis of this problem, notation is defined below:

\[ T \] the length of planning horizon;
\[ N \] the number of retailers;
\[ D_i \] the demand of retailer \( i, i = 1, \ldots, N; \)
\[ h_{it} \] the inventory holding cost rate at retailer \( i \) during period \( t, i = 1, \ldots, N, t = 1, \ldots, T; \)
\[ p_{it} \] the inventory penalty cost due to backorders of retailer \( i \) during period \( t, i, t = 1, \ldots, N, t = 1, \ldots, T; \)
\[ D_{it} \] the distance node \( i \) and node \( i', i = 0, \ldots, N, i' = 0, \ldots, N; \)
\[ C \] the truck capacity;
\[ \alpha \] the transportation charge per unit distance of a truck;
\[ \beta \] the fixed cost of dispatching a truck;
\[ \gamma \] the truck rental fee through the planning horizon of one truck.

The decision maker needs to decide in which period, in what quantity and in which route a delivery should be made, in order to minimize the total costs. Hence, the decision variables include:

\[ V_t \] the number of trucks dispatched in period \( t, t = 1, \ldots, T; \)
\[ V_{\text{rent}} \] the number of trucks rented in the planning horizon;
\[ Z_{it} \in \{0, 1\} \text{ indicate if truck } v \text{ travel between node } i \text{ and } i' \text{ in period } t; Z_{it} = 0 \text{ else}; \]
\[ Q_{it} \] the delivery quantity of retailer \( i \) in period \( t, i = 1, \ldots, N, t = 1, \ldots, T; \)
\[ I_{it} \] the inventory level of retailer \( i \) in period \( t, i = 1, \ldots, N, t = 1, \ldots, T; \)
\[ B_{it} \] the backorder level of retailer \( i \) at the end of period \( t, i = 1, \ldots, N, t = 1, \ldots, T; \)

Based on the above notation, the problem can be formulated as:

\[
\begin{align*}
\text{Min} & \quad \sum_{t=1}^{T} \left[ \alpha \sum_{i=1}^{N} \sum_{v=1}^{V_t} \sum_{i' \neq i} D_{it} Z_{it}^v + \beta \sum_{i=1}^{N} V_t + \sum_{i=1}^{N} (h_{it} I_{it} + p_{it} B_{it}) \right] \\
\text{s.t.} & \quad \sum_{i=1}^{N} \sum_{v=1}^{V_t} \sum_{i' \neq i} Z_{it}^v \leq 1, \quad \forall i = 0, \ldots, N; \quad t = 1, \ldots, T; \\
& \quad \sum_{i=1}^{N} Z_{it}^v - \sum_{m=0, m \neq i}^{N} Z_{it}^v = 0, \quad \forall i = 1, \ldots, N; \quad t = 1, \ldots, T; \quad v = 1, \ldots, V_t; \\
& \quad V_t \leq V_{\text{rent}}, \quad \forall t = 1, \ldots, T; \\
& \quad \sum_{i=1}^{N} \sum_{v=1}^{V_t} \sum_{i' \neq i} Q_{it} Z_{it}^v \leq C, \quad \forall t = 1, \ldots, T; \quad v = 1, \ldots, V_t; \\
& \quad C \sum_{i=1}^{N} \sum_{v=1}^{V_t} \sum_{i' \neq i} Z_{it}^v \geq Q_{it}, \quad \forall i = 1, \ldots, N; \quad t = 1, \ldots, T; \\
& \quad I_{it} = \begin{cases} 
\sum_{k=1}^{S} Q_{it} & \text{if } \sum_{k=1}^{S} Q_{it} - D_{it} > 0, \\
0 & \text{otherwise,} \\
\end{cases} \quad \forall i = 1, \ldots, N; \quad t = 1, \ldots, T; \\
& \quad B_{it} = \begin{cases} 
- \left( \sum_{k=1}^{S} Q_{it} - D_{it} \right) & \text{if } \sum_{k=1}^{S} Q_{it} - D_{it} < 0, \\
0 & \text{otherwise,} \\
\end{cases} \quad \forall i = 1, \ldots, N; \quad t = 1, \ldots, T.
\end{align*}
\]
This is a mixed integer programming problem. Formula (1) is the objective function including transportation cost (travel cost, dispatch cost, truck rental cost) and inventory cost. Constraints (2) and (3) are the typical traveling salesman problem (TSP) constraints; constraint (4) provides the lower bound of needed trucks; constraint (5) is the capacity constraint; constraint (6) expresses the relationship of $Q$ and $Z$; constraint (7) and (8) show how the inventory and backorder levels are calculated, respectively. This problem is clearly a NP-hard problem, since it contains TSP as subproblems.

3. Analysis of inventory and transportation plans

In this section, the characteristics of inventory and transportation plan are analyzed, respectively. In addition, a sufficient condition to obtain an optimal plan is obtained.

3.1. Inventory plan

Obviously, the optimal inventory plan is to replenish the retailers’ inventory in every period with the deliveries that equal to the demands of retailers. Under this situation, the inventory cost is zero.

3.2. Transportation plan

The transportation cost is composed of three components, so there are three kinds of transportation plan that minimize every cost, respectively.

The first one is to minimize the dispatch cost. Under this consideration, all trucks are loaded as full as possible, so the number of dispatched trucks is minimized. At this time, only the direct shipping strategy can be adopted. Hence, the minimal number of dispatched trucks can be calculated as

$$
(V_{t})_{\text{min}} = \left\lfloor \frac{\sum_{i=1}^{n} D_{i}}{C} \right\rfloor,
$$

where the load on a truck could be for multiple retailers, the peddling shipping should be considered.

The second kind is to minimize truck rental cost. Since the demands are constant for all retailers over the planning horizon, the optimal inventory plan could minimize the truck rental cost too, where the number of trucks dispatched is the same. Under this situation, the transportation mode could be a mix of direct shipping and peddling shipping. The least number of trucks is:

$$
(V_{\text{rent}})_{\text{min}} = \left\lfloor \frac{\sum_{i=1}^{n} D_{i}}{C} \right\rfloor.
$$

The third is to minimize the travel cost. This kind of transportation plan is hard to express explicitly, since there exist a list of TSPs to solve.

3.3. How to obtain an optimal plan

Based on the above discussion, the following property of the optimal plan can be found.

**Proposition 1.** There exists an optimal plan, in which for any retailer $i$ whose demand $D_{i}$ exceed the truck capacity, there exist $\lfloor D_{i}/C \rfloor$ trucks dispatched to retailer $i$ directly with $\lfloor D_{i}/C \rfloor$ products.

According to Proposition 1, an initial plan can be obtained by letting initial quantity of products delivered to retailer $i$ in period $t$ equal to its demand:

$$
Q_{it}^{\text{ini}} = D_{i},
$$

If $Q_{it}^{\text{ini}} = nC (n$ is an integer), the optimal delivery quantity is $Q_{it}^{\text{ini}}$, otherwise, $Q_{it}^{\text{ini}}$ is split into full-load delivery $Q_{it}^{f}$ and non-full delivery $Q_{it}^{nf}$,

$$
Q_{it}^{f} = C \times \lfloor Q_{it}^{\text{ini}} / C \rfloor; \quad Q_{it}^{nf} = Q_{it}^{\text{ini}} - C \times \lfloor Q_{it}^{\text{ini}} / C \rfloor.
$$

The shipping strategy is different for $Q_{it}^{f}$ and $Q_{it}^{nf}$. For $Q_{it}^{f}$, the optimal shipping strategy is direct shipping. All the trucks sent to them are loaded full, so there is no need for combing deliveries to different retailers into a route. The number of truck
dispatched to retailer $i$ in period $t$ is $\lfloor Q_{it}^{mi} / C \rfloor$. For $Q_{it}^{nf}$, the optimal shipping strategy is peddling shipping. The cost incurred by shipping all $Q_{it}^{nf}$ is $T \sum_{n=1}^{N} [(2aD_{in} + b)Q_{it}^{mi} / C_i] + \gamma \sum_{n=1}^{N} Q_{it}^{nf} / C_i$.

The remaining problem, named as non-ful delivery scheduling (NDS), is how to schedule the deliveries for the non-ful delivery. Three kinds of decisions must be made:

1. when the trucks should be sent to fulfill non-ful deliveries,
2. which routes trucks should travel on, and
3. how many trucks should be dispatched in each period.

3.4. Trade-off between inventory and transportation

It is intuitive that the total cost can be reduced if the plans of inventory and transportation can be considered together. Besides, the transportation cost is composed of travel cost, as well as dispatch cost and truck rental cost that are of strong relevance to the number of trucks used in each period. Combining different deliveries into one route can reduce the transportation cost. The following example could illustrate such a scenario.

In Fig. 1, the transportation cost is definitely reduced, and the inventory cost can either increase or decrease. If inventory cost decrease, the total cost is obviously reduced; and if inventory cost increase, the total cost could be reduced as well (if decrease of the transportation cost is higher than the increase of inventory cost).

Note that there are many other scenarios that the total cost could be reduced by rescheduling the inventory and transportation plan. For example, under the peddling shipping strategy, by being “brought forwards or postponed” from one period to the other, a delivery may be combined with other deliveries to be finished on one truck; this certainly increase the inventory cost, but decrease the transportation cost too. The more the trucks carry, the less the number of dispatched trucks is. It is easy to observe that loading each dispatched truck as full as possible may have a chance to reduce the transportation cost.

4. A heuristic algorithm using sliding windows and hierarchical tree

In this section, a heuristic algorithm using sliding windows and hierarchical tree (HASH) is developed to solve NDS. As discussed in the above sections, a trade-off between inventory and transportation could help reduce the cost of NDS.

Specifically, three factors below can directly affect cost of NDS.

1. The number of dispatched trucks in each period.
2. The distance that the trucks travel in each period.
3. The maximal number of dispatched trucks over the planning horizon.

In the following, key techniques in HASH are discussed.

4.1. Delivery parts

When a delivery is rescheduled, it may be too large to be combined with other deliveries into a truck when there are still many non-ful trucks, so little quantity deliveries are more convenient for rescheduling. Because when they are smaller, the deliveries have better chance to be “inserted” into a truck that still has remaining capacity. Out of this consideration, this
paper adopts a method that divides every non-ful delivery into several “delivery parts”. However, since the aim is to load the trucks as full as possible, so the quantity of delivery parts must be proportion to the truck capacity. Here, a parameter \( e (0 < e < 1) \) is defined, which is named as vehicle ratio coefficient and set by decision maker and can help split each \( Q_{nf}^i \) into a number of delivery parts. According to \( e \), the quantity of a delivery part can be calculated by formula (14) as follow

\[
C' = C \times e.
\]  

(14)

Using \( e \) and \( C' \), \( Q_{nf}^i \) of retailer \( i \) can be divided into \( \lceil Q_{nf}^i / C' \rceil \) delivery parts. Of course, one of delivery parts is maybe less than \( C' \). For \( C' \) is small enough, it is acceptable not to distinguish this “small delivery” with other delivery parts equal to \( C' \).

With the concept of delivery parts, the trade-off between inventory and non-ful deliveries is transformed into the trade-off between inventory and delivery parts.

4.2. Sliding window

Even the planning horizon is finite, it seems still computational impossible to consider the complete planning horizon in one time. It is known that two delivery parts that are originally scheduled in two separate periods far away from each other are unlikely to be combined together, due to the existence of inventory cost. Based on this observation, the sliding window technique is introduced here. Sliding window is a concept widely used in computer science, such as in Internet flow control protocol. When the volume of data is too large to be handled together, sliding windows can help deal with data in blocks, which both improve the efficiency and keep the relationship between the data in the meantime.

Conceptually, a sliding window is a set of continuous units. Here, a period represents a unit. There are two important characteristics in a sliding window used in this paper: the length and the step size. The length of a sliding window specifies how many periods are considered together; and the step size determines the distance of two consecutive sliding windows. The computational sequence of sliding windows is in an ascending order of their starting periods.

With sliding window, the solution of NDS can be performed in a “window-by-window” way, and the process can be illustrated in Fig. 2.

There is a \( T \)-period planning horizon in Fig. 2, the length of a sliding window is set to three, and the step size is one. Instead of being considered together, the \( T \) periods are decomposed into a list of intercrossing 3-period “shorter” planning horizons that are the so-called sliding windows. As shown in the figure, the current sliding window is \{2,3,4\}. After the subproblem of this sliding window is solved, the plan in period 2 is applied, and the subproblem of next sliding window \{3,4,5\} is considered.

With sliding windows, a balance of computational complexity and the quality of solution could be kept.

4.3. Hierarchical tree

However, to solve the subproblem within a sliding window, there are still too many possible delivery parts to be considered (Assume the number of delivery parts is \( N \), and the length of sliding window is \( L \), the maximal number of combination is \( L^N \)); and for each combination, there are still a list of TSPs to solve. Here a hierarchical tree is constructed to help decide the transportation plan.

The hierarchical tree mentioned here is a binary tree, and is used for two purposes: One is to find out which the group of delivery parts are mostly possible be combined together; and the other is to approximate the shortest distances a truck should travel in order to visit all the retailers in the above found groups.

Intuitively, retailers that stay in the neighborhood are most likely to be served together, and the hierarchical tree is adopted based on this observation. In every level of hierarchical tree, retailers that are close to each other are grouped together, and a group can be viewed as a “retailer” from the standpoint of the upper level group. This idea originates from cluster analysis. The building process of a hierarchical tree can be found in [15].
An example of a hierarchical tree structure is illustrated as follows:

In Fig. 3, a circle represents a retailer node, in which the first number is the index of some retailer and the second the shortest distance traveled by a truck departing from and returning to warehouse. In addition, an ellipse represents a group of retailers that are close to each other geographically. The “closeness” is measured by the distance. The distance between a group of retailers and a retailer is the smallest distance between the retailers in the group and the other retailer; and the distance between two groups of retailers is the smallest distance between the retailers from the two groups.

Based on this tree, the closeness level between retailers can be evaluated; so which retailers are most likely to be visited by the same truck is known. For the delivery parts that stay within a sliding window, an algorithm should be run on the hierarchical tree to find out of which the delivery parts are the most possible ones to be combined with other delivery parts.

In the process of HASH, it is not preferable that a single truck serves too much retailers. This requirement can be carried out by not allowing too many retailers in a group when these retailers are searched through the tree. In the heuristic, this limit is set to $\sqrt{N}$ retailers. This limit also guarantees retailers far away from each other are not served together.

The algorithm to find all possible routes in a sliding window is described as follows:

FindRouteSets (A sliding window)

**Step 1:** Find which retailers are in need of non-ful deliveries in the sliding window.
**Step 2:** Sum up the delivery quantity of retailers found.
**Step 3:** For each retailer found, calculate the full-load and non-ful deliveries according to (12), (13). Each of the full-load deliveries is treated as a route (Direct shipping), and split the non-ful deliveries into delivery parts.
**Step 4:** According to the hierarchical tree, group the retailers without violating the truck capacity constraint, and the number of retailer in a group cannot exceed $\sqrt{N}$. Every group of retailers is a route.
**Step 5:** Return all the routes.

The second purpose of the hierarchical tree is to approximate the travel distance in a group of retailers. The basic idea is as follows: first, the truck is sent to the nearest retailer, and then the retailers in the same level, and the nearest retailer in the upper level, and so on. An example is given below.

In Fig. 4, retailer 1, 2, and 3 are grouped together, a truck should be sent to serve these retailers. Suppose the retailer 1 is nearest to the warehouse and the travel distance of the truck can be approximated as:

$$D(1, 2, 3) = (D_{01} + D_{12}) + D_{23} + D_{30}.$$ 

The algorithm to compute the travel distance of a group of retailers $S$ can be described as:

CalculateRoutedistance ($S$: a group of retailers)

//dis represents the distance traveled by the truck; $G$ represents a group.

**Step 1:** Set $x = 0, dis = 0, G = \phi$.
**Step 2:** Search for $x$=the nearest retailer to the warehouse, $G = \{x\}$.
**Step 3:** $dis = dis + D_{0x}$, add $x$ on the route.
**Step 4:** Update the dis.
Step 4.1: Search for \( y \) = the index of retailer that is nearest to \( x \) in \( G \) and not on the route.

Step 4.2: If (\( y \) exists) \( \text{dis} = \text{dis} + D_{xy}, x = y \), add \( y \) on the route.

Step 4.3: If (There exists a retailer in \( G \) that is still not on the route) go to Step 4.1.

Step 4.4: If (There exists no retailer in \( G \) that is still not on the route and \( G \neq S \))

Step 4.4.1: \( G = G \)’s direct father node.

Step 4.4.1: goto Step 4.1.

Step 5: \( \text{dis} = \text{dis} + D_{x0} \).

Step 6: Return \( \text{dis} \).

There are literatures on applying clustering or partition techniques in similar problems. However, the clustering is done either in once-and-for-all way, or repeatedly in each period. Since clustering is quite a time-consuming process, it is preferable that this process executed just once; however, this way loses certain flexibility, for each retailer can only appear in a certain cluster, and the demands in the group can sometimes exceed the capacity of a truck. With hierarchical tree, on one hand, the efficiency is guaranteed for the hierarchical tree just need building once; on the other hand, the flexibility is not kept for any retailer can belong to several groups.

4.4. The overall procedure of the HASH

Finally, HASH can be described as follows:

The heuristics (\( L \): the length of sliding window)

Step 1: Initialize the transportation according to (11)–(13). Let \( V_{rent} \) = the largest number of used truck in a single period.

Step 2: Build a hierarchical tree, and compute the travel distance for each group of retailers.

Step 3: For each period, implement the following two substeps

Step 3.1: Find the retailer groups of which retailers are near to each other.

Step 3.2: Design the routes for groups of which the sums of non-ful delivery quantity are less than truck capacity and larger then \( C \times (1 - \varepsilon) \).

Step 4: Build a new hierarchical tree (eliminate the retailers that have already been served in Step 3).

Step 5: Divide the delivery of \( Q_{i}^{\text{ret}} \) into several delivery parts according to the number of delivery parts of retaileri.

Step 6: Initialize the first sliding window by setting the start of the window to period 1 and the end of the window to period \( L \).

Step 7: Reschedule the “deliveries” for delivery parts within current sliding window.

Step 7.1: FindRouteSets (the current sliding window).

Step 7.2: Re-schedule the transportation plan: for every possible transportation plan,

If (the maximal number of trucks is less than \( V_{rent} \))

Calculate the inventory and transportation cost in the sliding window.

Step 7.3: Update the transportation plan according to the plan with minimal cost.

Step 7.4: Update \( V_{rent} \).
Step 7.5: Add 1 to the start of window and the end of window.

Step 8: If the end of the current sliding window ≠ the end of planning horizon, go to Step 7; else, return the best solution.

Note that each time a solution is obtained, it is not necessary to calculate the total cost; the cost within the sliding window and the according $V_{rent}$ is enough for comparison.

Even a plan in some sliding window is optimal, it may fail to be optimal in the planning horizon. Therefore, it is reasonable to choose several “good” solutions in a single sliding window, then pass them all as input of the next sliding window, and repeat the process until the sliding window reach the end of planning horizon. However, for simplicity, only the “best” solution is chosen in each sliding window in the above description.

5. Numerical experiments

In this section, numerical experiments are conducted to test HASH. Due to different environment companies are facing, various ways for synchronization of inventory and transportation planning are adopted in practice. However, in spite of their external difference, these methods are based on one of the following two methodologies: just-in-time (JIT) transportation and fully loaded truck (FLT) transportation. Under JIT transportation, the delivery quantities to each retailer are exactly equal to its demands, so there is no stock hold. This is the near optimal ways when the inventory much more expensive than the transportation. Another typical method is FLT, in which vehicle are always loaded as full as possible. Contrary to JIT, FLT is suitable for scenarios where transportation cost is much higher then inventory cost.

To examine the solutions of HASH, comparisons with JIT and FLT is made. Note that JIT has an insertion algorithm inside to solve its TSP subproblems.

Some data in the following experiments is collected from a glass company in China with proper modification. Other data are generated randomly within a reasonable range. This company has 10 retailers in the nearby cities indexed from 1 to 10. The company is located at (0,0), and the locations of retailers are listed in Table 1.

A 20-week planning horizon is considered. The demand, inventory cost, and penalty cost at each retailer are shown in Table 2:

The company has signed a contract with a transportation enterprise for its distribution of products; the contents of the contract are in Table 3:

The cost incurred by using different methods are given as shown in Table 4:

Even though this is a simple example, the main differences of the three methods can be observed. JIT incur no inventory cost, but the transportation JIT made is far from optimal. FLT offers the best transportation plan, since most of trucks are made direct shipping. Both the inventory and transportation yielded by HASH is not the best; however, HASH achieves a good balance between inventory and transportation, so that the total costs are minimal.

The delivery quantities and rented trucks in each period are shown in Table 5.
In Table 5, “110 + 10” means some retailer receives two deliveries in the period, 110 means a fully loaded truck is sent to the retailer; 10 is a non-ful delivery.

Moreover, the delivery routes in each period are shown in Table 6.

Moreover, to compare the above three methods under different demands, the following experiments are conducted (ten retailers are considered). When demands of retailers is between \([0,1C]\), \([1C,2C]\), \([2C,3C]\), \([4C,5C]\), \([7C,8C]\), \([9C,10C]\) (\(C\) is the truck capacity), respectively, results of different methods are shown in the Table 7.

It can be easily observed from Table 7 that when the demands increase, the advantage of HASH seems less clear (the differences do not change much while the total cost increase). This phenomenon happens because when the demands are far larger than the capacity, most of trucks dispatched are already fully loaded. Hence, it is noticeable that HASH performs quite well when peddling shipping is the major source of cost; and when direct shipping is the major form of transportation, the benefits obtained from HASH is not so significant. However, HASH can still significantly reduce the cost incurred by peddling shipping in all cases.

As shown in Table 8 that HASH does not perform well on reducing the number of rented trucks, and sometimes it is even the worst choice; and FLT is the best choice.

In conclusion, when a decision maker has to choose between these three methodologies, the following rule can be adopted. When the demands are far larger than the truck capacity, JIT and FLT may be good choices (if the inventory cost is high, choose JIT; or, chose FLT), since they are much easier to understand and execute; and HASH seems to be a good choice when the demands are lower, which is especially common in retailer business.

Through the above discussion, one can conclude that HASH exceeds JIT and FLT when the demands are not very large. HASH seems a good choice for decision makers. However, a decision maker may be concerned with the computational cost of the algorithm in many real-time environments, although the qualities of solutions yielded by HASH are better. Since the major factor is the number of retailers. The computation time of JIT, FLT and HASH under different number of retailers is compared (only the case where demands are between \([0,1C]\) are considered, since larger demands will only incur fully loaded trucks which don’t affect the efficiency too much).

As shown in Fig. 5, FLT is the fastest method, since no vehicle routing process is executed in it. JIT is the second fastest, and HASH is the slowest. However, HASH is still computational acceptable, since it rarely happens in industry that one warehouse supplies more than 40 retailers.

Table 5
The delivery quantities of each retailer and the number of rented trucks in each period

<table>
<thead>
<tr>
<th>Week 1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>Rented trucks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>80</td>
<td>100</td>
<td>40</td>
<td>40</td>
<td>30</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>110</td>
<td>110</td>
<td>110 + 10</td>
<td>30</td>
<td>110</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>110</td>
<td>110</td>
<td>30</td>
<td>110</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>110</td>
<td>110</td>
<td>30</td>
<td>110</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td>110</td>
<td>110</td>
<td>30</td>
<td>110</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>30</td>
<td>110</td>
<td>110</td>
<td>30</td>
<td>110</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>30</td>
<td>110</td>
<td>110</td>
<td>30</td>
<td>110</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>30</td>
<td>110</td>
<td>110</td>
<td>30</td>
<td>110</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>30</td>
<td>110</td>
<td>110</td>
<td>30</td>
<td>110</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>30</td>
<td>110</td>
<td>110</td>
<td>30</td>
<td>110</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rented trucks</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
This paper investigates the synchronization of inventory and transportation planning in a one-warehouse multiple-retailer distribution network under flexible vehicle constraint. The transportation vehicles are outsourced from the 3rd party logistic companies. After an analysis of vehicle rental mode of transportation, a mixed integer program is formulated. Based on the characteristics of the problem, a mixed transportation mode of direct shipping and peddling shipping is adopted. A heuristic algorithm using sliding windows and hierarchical tree is developed to solve the problem in an acceptable computation complexity and accuracy. Numerical experiments show that in practical settings where the number of non-ful trucks is huge, our solution methodology can yield better plans compared to JIT and FLT based approaches. Finally, it is noticeable that the meth-

| Table 6 |
The routes in each period |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1:</td>
<td>0-7-0</td>
<td>0-5-3-0</td>
<td>0-8-9-10-0</td>
</tr>
<tr>
<td>Week 2:</td>
<td>0-2-0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 3:</td>
<td>0-7-0</td>
<td>0-4-3-5-0</td>
<td>0-8-9-10-0</td>
</tr>
<tr>
<td>Week 4:</td>
<td>0-6-0</td>
<td>0-5-3-0</td>
<td></td>
</tr>
<tr>
<td>Week 5:</td>
<td>0-7-0</td>
<td>0-8-9-10-0</td>
<td></td>
</tr>
<tr>
<td>Week 6:</td>
<td>0-1-0</td>
<td>0-2-0</td>
<td>0-4-0</td>
</tr>
<tr>
<td>Week 7:</td>
<td>0-7-0</td>
<td>0-5-3-0</td>
<td>0-8-9-10-0</td>
</tr>
<tr>
<td>Week 8:</td>
<td>0-1-0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 9:</td>
<td>0-2-0</td>
<td>0-7-0</td>
<td>0-4-3-5-0</td>
</tr>
<tr>
<td>Week 10:</td>
<td>0-5-3-0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 11:</td>
<td>0-7-0</td>
<td>0-8-9-10-0</td>
<td></td>
</tr>
<tr>
<td>Week 12:</td>
<td>0-4-0</td>
<td>0-6-0</td>
<td>0-10-0</td>
</tr>
<tr>
<td>Week 13:</td>
<td>0-2-0</td>
<td>0-7-0</td>
<td>0-5-3-0</td>
</tr>
<tr>
<td>Week 14:</td>
<td>0-1-0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 15:</td>
<td>0-7-0</td>
<td>0-4-3-5-0</td>
<td>0-8-9-10-0</td>
</tr>
<tr>
<td>Week 16:</td>
<td>0-5-3-0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 17:</td>
<td>0-2-0</td>
<td>0-7-0</td>
<td>0-5-3-0</td>
</tr>
<tr>
<td>Week 18:</td>
<td>0-4-0</td>
<td>0-5-0</td>
<td>0-10-0</td>
</tr>
<tr>
<td>Week 19:</td>
<td>0-7-0</td>
<td>0-1-3-0</td>
<td>0-8-9-10-0</td>
</tr>
<tr>
<td>Week 20:</td>
<td>0-6-0</td>
<td>0-2-5-0</td>
<td></td>
</tr>
</tbody>
</table>

| Table 7 |
Cost of different methods under different demands |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[0,1C]</td>
<td>[1C,2C]</td>
<td>[2C,3C]</td>
<td>[4C,5C]</td>
<td>[9C,10C]</td>
<td></td>
</tr>
<tr>
<td>JIT</td>
<td>215,809</td>
<td>375,244</td>
<td>690,825</td>
<td>1,085,386</td>
<td>1,943,095</td>
</tr>
<tr>
<td>FLT</td>
<td>173,647</td>
<td>340,073</td>
<td>678,774</td>
<td>942,858</td>
<td>1,908,323</td>
</tr>
<tr>
<td>HASH</td>
<td>164,179</td>
<td>338,339</td>
<td>634,828</td>
<td>924,898</td>
<td>1,893,763</td>
</tr>
</tbody>
</table>

| Table 8 |
The number of rented trucks of different methods under different demands |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[0,1C]</td>
<td>[1C,2C]</td>
<td>[2C,3C]</td>
<td>[4C,5C]</td>
<td>[9C,10C]</td>
<td></td>
</tr>
<tr>
<td>JIT</td>
<td>3</td>
<td>15</td>
<td>26</td>
<td>45</td>
<td>97</td>
</tr>
<tr>
<td>FLT</td>
<td>3</td>
<td>13</td>
<td>24</td>
<td>42</td>
<td>93</td>
</tr>
<tr>
<td>HASH</td>
<td>5</td>
<td>15</td>
<td>25</td>
<td>46</td>
<td>97</td>
</tr>
</tbody>
</table>

Fig. 5. Computation time of different methods.

6. Conclusion

This paper investigates the synchronization of inventory and transportation planning in a one-warehouse multiple-retailer distribution network under flexible vehicle constraint. The transportation vehicles are outsourced from the 3rd party logistic companies. After an analysis of vehicle rental mode of transportation, a mixed integer program is formulated. Based on the characteristics of the problem, a mixed transportation mode of direct shipping and peddling shipping is adopted. A heuristic algorithm using sliding windows and hierarchical tree is developed to solve the problem in an acceptable computation complexity and accuracy. In the heuristic algorithm, sliding windows are used to approximate the problem by repeatedly a list of intersecting subproblems, and hierarchical tree is used to measure the closeness of different group of retailers. Numerical experiments show that in practical settings where the number of non-ful trucks is huge, our solution methodology can yield better plans compared to JIT and FLT based approaches. Finally, it is noticeable that the meth-
odology proposed by this paper can indeed be applied into other multiple-period vehicle scheduling problems with minor revisions.

Acknowledgements

The authors are grateful to two anonymous referees for their suggestions on improving the quality of this paper. The research is supported by the Natural Science Foundation of China (NSFC 70625001, 70721001, 70601004), the Key Project (104064) of Research, Scientific of Ministry of Education (MOE) in China, the program of New Century Excellent Talents in University (NCET-04-280) of MOE of China and a Strategic Research Grant (SRG) from City University of Hong Kong (7002078).

References