Decision Making in Manufacturing and Services
Vol. 6 • 2012 • No. 1 • pp. 5–23

Optimization in Supply Chain Management

An Incremental Approach for Storage and Delivery Planning Problems

Kazutoshi Sakakibara*, Yajie Tian**, Ikuko Nishikawa*

Abstract. We consider a logistic planning problem for simultaneous optimization of the storage and the delivery. This problem arises in the consolidate shipment using an intermediate storage in a supply chain, which is typically found in the automobile industry. The vehicles deliver the items from the origin to the destination, while the items can be stored at some warehousing facilities as the intermediate storage during the delivery. The delivery plan is made for each day separately, but the storage at a warehouse may last for more than one day. Therefore, the entire logistic plan should be considered over a certain period for the total optimization. We formulate the storage and delivery problem as a mixed integer programming. Then, we propose a relax-and-fix type heuristic method, which incrementally fixes decision variables until all the variables are fixed to obtain a complete solution. Moreover, a semi-approximate model is introduced to effectively fix the variables. Based on the formulation, the delivery plan can be solved for each day separately. This has the advantage especially in the dynamic situation, where the delivery request is modified from the original request before the actual delivery day. Numerical experiments show that the simultaneous optimization gives the effective storage plan to reduce the total logistic cost, and the proposed heuristics efficiently reduce the computational time and are robust against the dynamic situation.

Keywords: simultaneous optimization, logistic terminal, consolidate shipment, mixed integer programming, relax-and-fix

Mathematics Subject Classification: 90B06 – Transportation, logistics, 90C11 – Mixed integer programming

Revised: December 09, 2011, Accepted by Dirk Briskorn and Florian Jaehn special issue editors

* College of Information Science and Engineering, Ritsumeikan University; Noji-Higashi, Kusatsu city, Shiga 525-8577, JAPAN; E-mail: {sakaki, nishi}@ci.ritsumei.ac.jp
** Nihon Unisys, Ltd.; Koto ward, Tokyo 135-8560, JAPAN; E-mail: yjtian@bj-unity.com

DOI: http://dx.doi.org/10.7494/dmms.2012.6.1.5

1. INTRODUCTION

Logistic terminals which provide warehousing services are recently attracting more attention for the efficient and adaptive logistics. For example, public logistic terminals are widely utilized as temporal warehousing facilities for the delivery from the parts suppliers to the assembly plants of half-finished products, such as the parts of an automobile. Consolidate shipment from multiple suppliers is becoming common in such a supply chain, instead of a traditional style of a direct transportation from one supplier to one plant. Then, the consolidate shipment with an intermediate storage is expected to reduce the cost and to make the logistics more adaptive to the change. Efficient logistics also leads to the decrease of the carbon dioxide discharge by decreasing the number of the vehicle and the lead-time for the delivery. This is the background of the simultaneous optimization of the storage plan and the delivery plan based on a practical requirement. However, the logistic planning problem becomes more complex if we consider the storage and the delivery simultaneously. We investigate this *storage and delivery problem* in this paper.

Several studies have been reported on the optimization regarding the logistic terminal. One traditional problem is how to locate the logistic terminals appropriately, and several mathematical models are developed to determine the optimal size and location of the terminal (Campbell 1990, Daganzo 1996, Taniguchi, Noritake, Yamada & Izumitani 1999). Onoyama et al. (Onoyama, Maekawa, Kubota, Tsuruta & Komoda 2008) proposed the fast solution method using genetic algorithms which is applicable to realistic size problems where a one-day delivery plan is optimized. On the other hand, we focus on the simultaneous optimization of the storage and delivery planning under the condition that the logistic terminals are already given. We consider that a fleet of vehicles deliver the items from the origins to the destinations exactly on each due date, namely, by a just-in-time delivery. Warehousing facilities are available on the way to the final destination for a temporary storage of some items. The storage may last for more than one day before another vehicle picks up the stored items for the further delivery. Therefore, the storage plan ranges over multiple days, while the delivery plan is made for each day. As the result, the total logistic plan extends over a long period of time.

Logistic terminal also serves for consolidate shipments, as the items collected from several suppliers are once stored and consolidated at an intermediate terminal before the final delivery to an assembly plant. Several studies have been reported which consider the storage terminals for the consolidate shipment. Chen et al. (Chen, Guo, Lim & Rodrigues 2006) studied a logistic system with multiple storage terminals, or, crossdocks. Their model determines the amount, supplier, route and crossdock of each item to fulfill the requests from customers. However, a vehicle is not considered explicitly, while our model expresses each vehicle with a finite capacity, which enables the study on the efficiency of the consolidate shipment using vehicles. Musa et al. (Musa, Arnaout & Jung 2010) also consider a logistic system with multiple crossdocks, and both the vehicle assignment and the vehicle routing are optimized. However, they do not consider the storage at the crossdock for multiple days, though the storage over multiple days is essential in our model. Another request for the consolidate shipment arises from Vendor Managed Inventory (VMI), which has been successfully introduced in several companies (Dong & Xu 2002). Though the present problem definition given in Section 2 assumes that the request is made from the destination side of the delivery, which differs from VMI, we should notice that the cost reduction by the consolidate shipment with an intermediate storage is the key issue in both frameworks.

We have developed a mixed integer programming (MIP) model for the above problem (Sakakibara, Tian & Nishikawa 2011), which explicitly considers each delivery request and contains the decision variables corresponding to the storage plan and to the delivery plan separately. As a result, the model has many integer variables with complex constraints, and its computational cost becomes rather huge to obtain the optimal solution for a large size problem. Therefore, we use relax-and-fix type heuristics based on the MIP structure to acquire near-optimal solutions in a reasonable computational time. Relax-and-fix heuristics (Pochet & Wolsey 2006) incrementally optimize each sub-problem step by step using a mathematical programming method, to fix decision variables incrementally until all the variables are fixed to obtain the solution of the original problem. Relax-and-fix is effective for a special type of MIP problem. It has been especially used for production planning (Pochet & Vyve 2004, Kelly & Mann 2004, Ferreira, Morabito & Rangel 2009, Akartunali & Miller 2009), which focused on the lot sizing problem (Comelli, Gourgand & Lemoine 2008). There are also some hybrid methods, which use both relax-and-fix heuristics and some meta heuristics, such as tabu search or nested partition (Pedroso & Kubo 2005, Wu & Shi 2009). Here we proposed a semi-approximate model for a sub-problem in applying relax-and-fix heuristics to our *storage and delivery problem*.

The paper is organized as follows. Section 2 gives the formulation of the present logistic problem by MIP model. Section 3 describes the incremental optimization method based on relax-and-fix heuristics. Then, Section 4 shows the results of computational experiments both in a static and a dynamic environments. Section 5 summarizes the present results.

2. STORAGE AND DELIVERY PROBLEM

2.1. DESCRIPTION OF THE PROBLEM

Let us consider the following problem. There are N delivery requests O_i $(i = 1, \dots, N)$ during the planning period T. Request O_i requires that the number m_i of size p_i item should be delivered from its supplier to a designated assembly plant on the due date t_i^{D}. There are three kinds of facilities, that is, a supplier as an origin, an assembly plant as a destination, and a logistic terminal as a mid-point storage. We consider an arbitrary number of suppliers and assembly plants, while only one logistic terminal, for the simplicity in the present paper. M vehicles V_k $(k = 1, \dots, M)$ are available for the delivery each day. A vehicle can visit multiple suppliers and assembly plants, and a route from the first visited supplier to the last visited assembly plant must be completed within one day. Traveling time by a vehicle between any pair of facilities is preliminary given as a constant value. The items can be stored at a logistic terminal for more than one day. And a storage cost is charged per unit size of an item per day.

The constraints in making the logistic plan are given as follows:

(a) Items of request O_i can be shipped from its supplier only after its completion date t_i^{R}.

(b) Items of request O_i should be delivered to its designated assembly plant exactly on the due date t_i^{D}, that is, a just-in-time delivery is required.

(c) Vehicle V_k has capacity $\bar{q}_k^{\mathrm{V}}$.

(d) Vehicle cannot visit more than 3 facilities in one day.

The constraint (d) is based on the following observation on a real data from an automobile company. Various parts are delivered daily by a long-distance massive transportation, which should be completed within one day by each vehicle. Then, one vehicle can visit only a few facilities in one day.

The objective of the storage and delivery planning is to minimize both the delivery cost by vehicles and the storage cost at a logistic terminal. Additionally, we consider the load balance over the planning period. The details of the objective function are given in **2.2**.

2.2. MIXED INTEGER PROGRAMMING MODEL

We formulate the above problem using a mixed integer programming (MIP) model. The problem is composed of two sub-problems: One is a storage problem denoted *problem* $\mathcal{P}$, and the other is a delivery problem denoted *problem* $\mathcal{Q}$. Then, $\mathcal{P}$ ranges over multiple days, while $\mathcal{Q}$ can be further decomposed into the delivery problem for each day. Delivery problem $\mathcal{Q}$ includes both vehicle assignment problem and vehicle routing problem (Lawler, Lenstra, Kan & Shmoys 1985). To avoid the computational complexity of the vehicle routing problem, here we instead solve a route selection problem, where a vehicle only needs to select a route from a set of candidate routes. This approach is based on the consideration that appropriate vehicle routes under given orientations and destinations are limited from the geographic and traffic conditions in an actual transportation system, in general. The general procedure to generate a set of candidate routes is taken as follows: first, to prepare a number of pairs of the orientation and destination facilities; then, for each pair, to obtain a number of routes which connect the orientation and destination facilities in ascending order of the lengths. The routes are obtained by solving a shortest path problem with resource constraints, which is a problem to find a minimum cost route under several constraints on a vehicle such as a minimum-maximum load capacity or a time window (Irnich & Desaulniers 2005). For real problems, road and traffic conditions or labor conditions may also be included as the constraints. Here, we give a set of L candidate routes in advance, which satisfy the constraint (d) given in **2.1**.

First, let us introduce the following notations and parameters to describe the MIP model. The uniqueness of the logistic terminal makes the notation simpler, while the formal generalization to multiple terminals is straightforward.

- Day t ($t = 1, \ldots, T$)
- Request O_i ($i = 1, \ldots, N$)
  - m_i : The number of items,
  - p_i : The size of one item,
  - t_i^{R} : The earliest shipping date,
  - t_i^{D} : The due date for the delivery,
  - The supplier,
  - The designated assembly plant.
- Logistic terminal
  - c^{K} : The storage cost per unit size per day.
- Route R_ℓ ($\ell = 1, ..., L$)
  - $b_{i\ell}^{\mathrm{S}} \in \{1, 0\}$: 1 if the supplier of O_i is on route R_ℓ, 0 otherwise.
  - $b_{i\ell}^{\mathrm{F}} \in \{1, 0\}$: 1 if the assembly plant of O_i is on route R_ℓ, 0 otherwise.
  - $b_{\ell}^{\mathrm{K}} \in \{1, 0\}$: 1 if the logistic terminal is on route R_ℓ, 0 otherwise.
- Vehicle V_k ($k = 1, \ldots, M$)
  - $\overline{q}_k^{\mathrm{V}}$: The capacity,
  - c_k^{I} : The initial cost,
  - $c_{k\ell}^{\mathrm{V}}$: The delivery cost taking R_ℓ.

Next, let us introduce the decision variables defined for $\mathcal{P}$ and $\mathcal{Q}$, individually.

- Variables for $\mathcal{P}$:
  - $x_{iu} \in \mathbb{Z}_+$ [1] : The number of items of O_i which are stored at a logistic terminal for u days [2].
  - v_t : Total size of all items which are shipped from the suppliers on day t.
  - v_t^{B} : Load balancing index of day t defined by,

$$v_t^{\mathrm{B}} \triangleq \left| v_t - \frac{1}{T} \sum_{t=1}^{T} v_t \right|. \tag{1}$$

    This is introduced to balance the daily shipping amount from the suppliers over the planning period, which is sometimes used as an efficiency index in a heuristic planning.
- Variables for $\mathcal{Q}$:
  - $y_{ik\ell t} \in \mathbb{Z}_+$: The number of items of O_i which are loaded into V_k taking R_ℓ on t.
  - $y_{ik\ell t}^{\mathrm{SF}} \in \mathbb{Z}_+$: The number of items of O_i loaded into V_k taking R_ℓ on t, which are directly delivered from suppliers to assembly plants without storage.
  - $y_{ik\ell t}^{\mathrm{SK}} \in \mathbb{Z}_+$: The number of items of O_i loaded into V_k taking R_ℓ on t, which are delivered from suppliers to a logistic terminal.

[1] $\mathbb{Z}_+$ represents a set of non negative integers.
[2] x_{i0} represents the number of items of O_i which are directly delivered without storage.

- $y_{ik\ell t}^{\mathrm{KF}} \in \mathbb{Z}_+$: The number of items of O_i loaded into V_k taking R_ℓ on t, which are delivered from a logistic terminal to assembly plants.
- r_{it}^{SF} : The number of items of O_i, which are directly delivered from suppliers to assembly plants without storage on t.
- r_{it}^{SK} : The number of items of O_i, which are delivered from suppliers to a logistic terminal on t.
- r_{it}^{KF} : The number of items of O_i, which are delivered from a logistic terminal to assembly plants on t.
- $z_{k\ell t} \in \{1, 0\}$: 1 if V_k takes R_ℓ on t, 0 otherwise.

Here, $\boldsymbol{x}$ and $\boldsymbol{y}$ are decision variables of $\mathcal{P}$ and $\mathcal{Q}$, respectively, and the other variables are dependent on $\boldsymbol{x}$ and $\boldsymbol{y}$.

Finally, the evaluations of each problem are defined by the following functions.

- Evaluation of $\mathcal{P}$
  - f^{K} : The storage cost at a logistic terminal.
  - f^{B} : The load balancing index defined by Eq.(1).
- Evaluation of $\mathcal{Q}$
  - f_t^{V} : The delivery cost on t, $t = 1, \ldots, T$.

The objective function for the total logistic problem is given by a weighted linear summation of the above evaluation functions. And, this formulation makes the problem solvable by the mathematical programming.

Now, we can describe the MIP model ($\mathcal{M}$) as the followings. The explanations on the equations are given in the succeeding paragraph.

minimize

$$w^{\mathrm{K}} f^{\mathrm{K}} + w^{\mathrm{B}} f^{\mathrm{B}} + w^{\mathrm{V}} \sum_{t=1}^{T} f_t^{\mathrm{V}}, \tag{2}$$

subject to

$$\sum_{u=0}^{t_i^{\mathrm{D}} - t_i^{\mathrm{R}}} x_{iu} = m_i, \qquad i = 1, \ldots, N \tag{3}$$

$$x_{iu} \in \mathbb{Z}_+, \qquad i = 1, \ldots, N,\ u = 0, \ldots, T-1 \tag{4}$$

$$v_t = \sum_{\substack{i=1,\\ t_i^{\mathrm{D}} \geqslant t}}^{N} p_i x_{i,\, t_i^{\mathrm{D}} - t}, \qquad t = 1, \ldots, T \tag{5}$$

$$v_t^{\mathrm{B}} \geqslant v_t - \frac{1}{T} \sum_{t=1}^{T} v_t, \qquad t = 1, \ldots, T \tag{6}$$

$$v_t^{\mathrm{B}} \geqslant \frac{1}{T}\sum_{t=1}^{T} v_t - v_t, \qquad t = 1, \ldots, T \tag{7}$$

$$f^{\mathrm{K}} = \sum_{i=1}^{N} \sum_{u=1}^{t_i^{\mathrm{D}}-t_i^{\mathrm{R}}} c^{\mathrm{K}} p_i u \, x_{iu}, \tag{8}$$

$$f^{\mathrm{B}} = \frac{1}{T}\sum_{t=1}^{T} v_t^{\mathrm{B}}, \tag{9}$$

$$r_{it}^{\mathrm{SK}} = \begin{cases} x_{i,t_i^{\mathrm{D}}-t} \ (t_i^{\mathrm{R}} \leqslant t < t_i^{\mathrm{D}}), \\ 0 \qquad\qquad (t < t_i^{\mathrm{R}},\, t \geqslant t_i^{\mathrm{D}}), \end{cases} \qquad i = 1, \ldots, N,\, t = 1, \ldots, T \tag{10}$$

$$r_{it}^{\mathrm{SF}} = \begin{cases} x_{i0} \ (t = t_i^{\mathrm{D}}), \\ 0 \quad (t \neq t_i^{\mathrm{D}}), \end{cases} \qquad i = 1, \ldots, N,\, t = 1, \ldots, T \tag{11}$$

$$r_{it}^{\mathrm{KF}} = \begin{cases} m_i - x_{i0} \ (t = t_i^{\mathrm{D}}), \\ 0 \qquad\qquad (t \neq t_i^{\mathrm{D}}), \end{cases} \qquad i = 1, \ldots, N,\, t = 1, \ldots, T \tag{12}$$

$$\sum_{i=1}^{N}\sum_{\ell=1}^{L} p_i y_{ik\ell t}^{\mathrm{SK}} + \sum_{i=1}^{N}\sum_{\ell=1}^{L} p_i y_{ik\ell t}^{\mathrm{SF}} \leqslant \bar{q}_k^{\mathrm{V}}, \qquad k = 1, \ldots, M,\, t = 1, \ldots, T \tag{13}$$

$$\sum_{i=1}^{N}\sum_{\ell=1}^{L} p_i y_{ik\ell t}^{\mathrm{KF}} + \sum_{i=1}^{N}\sum_{\ell=1}^{L} p_i y_{ik\ell t}^{\mathrm{SF}} \leqslant \bar{q}_k^{\mathrm{V}}, \qquad k = 1, \ldots, M,\, t = 1, \ldots, T \tag{14}$$

$$\sum_{k=1}^{M}\sum_{\ell=1}^{L} b_{i\ell}^{\mathrm{S}} b_{\ell}^{\mathrm{K}} y_{ik\ell t}^{\mathrm{SK}} = r_{it}^{\mathrm{SK}}, \qquad i = 1, \ldots, N,\, t = 1, \ldots, T \tag{15}$$

$$\sum_{k=1}^{M}\sum_{\ell=1}^{L} b_{i\ell}^{\mathrm{S}} b_{i\ell}^{\mathrm{F}} y_{ik\ell t}^{\mathrm{SF}} = r_{it}^{\mathrm{SF}}, \qquad i = 1, \ldots, N,\, t = 1, \ldots, T \tag{16}$$

$$\sum_{k=1}^{M}\sum_{\ell=1}^{L} b_{\ell}^{\mathrm{K}} b_{i\ell}^{\mathrm{F}} y_{ik\ell t}^{\mathrm{KF}} = r_{it}^{\mathrm{KF}}, \qquad i = 1, \ldots, N,\, t = 1, \ldots, T \tag{17}$$

$$y_{ik\ell t} = y_{ik\ell t}^{\mathrm{SK}} + y_{ik\ell t}^{\mathrm{SF}} + y_{ik\ell t}^{\mathrm{KF}}, \qquad \begin{array}{l} i = 1, \ldots, N,\, k = 1, \ldots, M, \\ \ell = 1, \ldots, L,\, t = 1, \ldots, T \end{array} \tag{18}$$

$$y_{ik\ell t},\, y_{ik\ell t}^{\mathrm{SF}},\, y_{ik\ell t}^{\mathrm{SK}},\, y_{ik\ell t}^{\mathrm{KF}} \in \mathbb{Z}_+, \qquad \begin{array}{l} i = 1, \ldots, N,\, k = 1, \ldots, M, \\ \ell = 1, \ldots, L,\, t = 1, \ldots, T \end{array} \tag{19}$$

$$\sum_{i=1}^{N} y_{ik\ell t} \Big/ \sum_{i=1}^{N} m_i \leqslant z_{k\ell t} \leqslant \sum_{i=1}^{N} y_{ik\ell t}, \qquad \begin{array}{l} k = 1, \ldots, M,\, \ell = 1, \ldots, L, \\ t = 1, \ldots, T \end{array} \tag{20}$$

$$\sum_{\ell=1}^{L} z_{k\ell t} \leqslant 1, \qquad k=1,\dots,M,\ t=1,\dots,T \tag{21}$$

$$f_t^{\mathrm{V}} = \sum_{k=1}^{M}\sum_{\ell=1}^{L}\Big(c_k^{\mathrm{I}} z_{k\ell t} + c_{k\ell}^{\mathrm{V}} z_{k\ell t}\Big), \quad t=1,\dots,T \tag{22}$$

$$z_{k\ell t} \in \{1,\, 0\}. \tag{23}$$

Eq.(3) gives the number of items of O_i as the sum over the storage period u at a logistic terminal, which ranges from 0 to $t_i^{\mathrm{R}} - t_i^{\mathrm{D}}$. Eq.(4) is the non negative integer constraint on $\boldsymbol{x}$. Eq.(5) gives the total size of the items shipped from suppliers on day t as the sum of $\boldsymbol{x}$. The second subscript of x, namely, $t_i^{\mathrm{D}} - t$, corresponds to the storage period u of items of O_i. Thus, the total size of all items v_t are given by the sum of the items which are delivered to a logistic terminal ($u > 0$) and those which are directly delivered to the assembly plants ($u = 0$). Eqs.(6) and (7) represent the linear expression of Eq.(1). Eq.(8) gives the total storage cost at a logistic terminal as the sum over the storage periods u. Eq.(9) is the evaluation on the load balancing, defined as the average of the load balancing index for each day. Eqs.(10), (11) and (12) show how $\boldsymbol{r}^{\mathrm{SK}}$, $\boldsymbol{r}^{\mathrm{SF}}$ and $\boldsymbol{r}^{\mathrm{KF}}$ are dependent on $\boldsymbol{x}$, respectively. Eqs.(13) and (14) ensure that the total load on each vehicle cannot exceed the vehicle capacity. Eq.(13) is the restriction for the vehicle from suppliers, while Eq.(14) is that for the vehicle heading to assembly plants. In Eqs.(15), (16) and (17), the numbers of items $\boldsymbol{r}^{\mathrm{SK}}$, $\boldsymbol{r}^{\mathrm{SF}}$ and $\boldsymbol{r}^{\mathrm{KF}}$ are given by the sums of the vehicle loads calculated from $\boldsymbol{y}^{\mathrm{SK}}$, $\boldsymbol{y}^{\mathrm{SF}}$ and $\boldsymbol{y}^{\mathrm{KF}}$, respectively. Eq.(18) gives the relation among various $\boldsymbol{y}$. Eq.(19) is the non negative integer constraint on $\boldsymbol{y}$. Eq.(20) sets the value of $z_{k\ell t}$ depending on $y_{ik\ell t}$. That is, Eq.(20) restricts the value of $z_{k\ell t}$ to 1 when there is at least one i satisfying $y_{ik\ell t} > 0$, under the binary constraint given by Eq.(23) , as the left most side $\sum_{i=1}^{N} y_{ik\ell t} / \sum_{i=1}^{N} m_i$ is the positive value smaller than 1. On the other hand, if $y_{ik\ell t} = 0$ for all i, then Eq.(20) sets the value of $z_{k\ell t}$ to 0. Eq.(21) states that each vehicle can select at most one route. Eq.(22) gives the delivery cost on day t, which consists of the initial cost c_k^{I} and the delivery cost $c_{k\ell}^{\mathrm{V}}$ taking route R_ℓ. Eq.(23) is the binary constraint on $\boldsymbol{z}$.

2.3. STRUCTURE OF MIP MODEL

Storage problem $\mathcal{P}$ is described as the optimization of the objective function given by the first and second terms of Eq.(2), under the constraints of Eqs.(3)-(9), while delivery problem $\mathcal{Q}$ is described as the optimization of the objective function given by the third term of Eq.(2), under the constraints of Eqs.(10)-(23). $\mathcal{P}$ and $\mathcal{Q}$ make up the original problem $\mathcal{M}$.

Regarding the storage problem $\mathcal{P}$, Eq.(3) imposes the constraint over multiple days, and evaluations f^{K} and f^{B} are expressed as linear sums over the planning period. Therefore, we cannot decompose the optimization problem $\mathcal{P}$ into individual day. On the other hand, once $\boldsymbol{x}$ are fixed by solving $\mathcal{P}$, $\mathcal{Q}$ can be decomposed into a sub-problem for each t, which we denote $\mathcal{Q}_t$. That is, the delivery can be optimized for each day separately, once the storage plan is determined. Here, $\mathcal{Q}_t$ has a strong similarity with the general assignment problem, where an item and its route in the delivery problem correspond to a job and its processor in the general assignment problem, respectively. The general assignment problem is known to be NP-hard. Therefore, it is supposed that the rigorous approach as a branch-and-bound algorithm becomes hardly applicable for the large size problem due to its computational complexity.

The structure of the MIP $\mathcal{M}$ is illustrated in Figures 1 and 2 by using $\mathcal{P}$ and $\mathcal{Q}$. In Figure 2, x_{iu} which satisfies $u = t_i^{\mathrm{D}} - t$ is described by a circle located in column t. More precisely, variable x_{iu} for $\mathcal{P}$, which denotes the number of the items stored at a logistic terminal for u days starting from t, is written in column t, to indicate that those items should be delivered to the logistic terminal on day t. And this information is passed to problem $\mathcal{Q}_t$ as a predefined delivery request for day t.

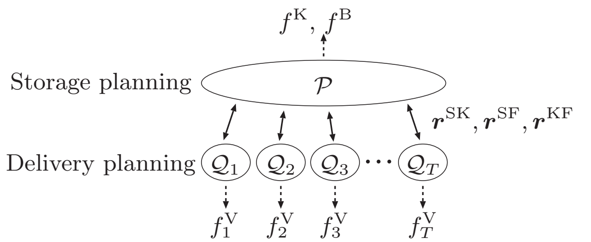


Fig. 1. *Storage planning $\mathcal{P}$ and delivery planning $\mathcal{Q}_t$.*

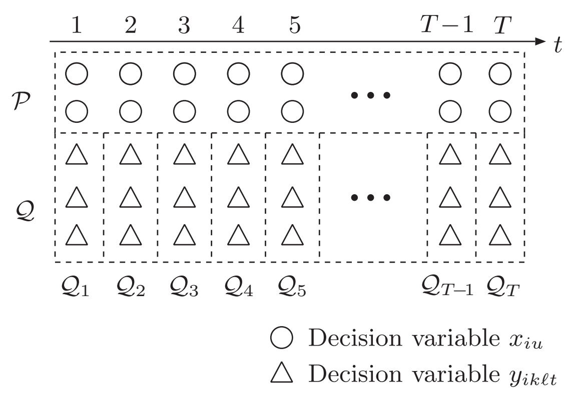


Fig. 2. *Structure of the MIP model.*

3. INCREMENTAL OPTIMIZATION METHOD

The MIP model $\mathcal{M}$ described in **2.2** can be solved rigorously by mathematical programming techniques, such as branch-and-bound algorithms (Nemhauser, Kan & Todd 1989). However, the rigorous approach requires a huge computational cost for this MIP problem, which includes many integer variables and constraints, as is already discussed in the previous section. Therefore, we prefer the heuristic approach to acquire near-optimal solutions in a reasonable computational time. Here we use relax-and-fix type heuristics, which is effective for the MIP structure. Relax-and-fix heuristics incrementally optimize each sub-problem using mathematical programming to fix decision variables step by step with day t until all the variables are fixed to obtain the complete solution of the original problem. Moreover, we introduce a semi-approximate model for a sub-problem to further reduce the computational cost, based on the problem structure which is decomposed into $\mathcal{P}$ and $\mathcal{Q}$. In the following, first the framework of relax-and-fix algorithm is given in **3.1**, and then a semi-approximate model for the present problem is described in more detail in **3.2**.

3.1. ALGORITHM OF RELAX-AND-FIX

Relax-and-fix heuristics reduce the number of the decision variables by fixing the values of some variables, which are called *fixed variables*, while the remaining variables are called *free variables*. In the original problem, all the variables are *free variables*. On the contrary, a complete solution of the original problem is obtained when all the variables become *fixed variables*.

We fix the variables in the temporal order, namely, along day t, in the application of relax-and-fix to $\mathcal{M}$. Following procedure $1° - 4°$ describes how to fix the variables incrementally along t, using the iteration index $\hat{t}$ for the incremental step. At each step of $\hat{t}$, sub-problem $\mathcal{M}_{\hat{t}}$ is solved rigorously by integer programming method. $\mathcal{M}_{\hat{t}}$ contains both *fixed variables* and *free variables* until the final step. By solving $\mathcal{M}_{\hat{t}}$, a part of *free variables* become *fixed variables*. The detailed definition of $\mathcal{M}_{\hat{t}}$ is given in **3.2** with parameter T^{S}.

The framework of the relax-and-fix is given by the following procedure:

1° **Initialization:** Set counter $\hat{t} := 1$.

2° **Solving a sub-problem:** Construct sub-problem $\mathcal{M}_{\hat{t}}$ according to the definition given in **3.2**. Set all the variables in $\mathcal{M}_{\hat{t}}$ as *free variables*. Solve $\mathcal{M}_{\hat{t}}$ by the integer programming method to obtain the optimal solution.

3° **Fixing incrementally:** If $\hat{t} > T - T^{\mathrm{S}}$, go to 4°. Otherwise, the variables with indices $t = \hat{t}$ become fixed to the values of the optimal solution obtained in step 2° [3]. Set $\hat{t} := \hat{t} + 1$ and return to 2°.

4° **Completion:** Fix all the variables to the values of the optimal solution obtained in step 2°. Output all the fixed values as a complete solution.

3.2. SEMI-APPROXIMATE MODEL

Sub-problem $\mathcal{M}_{\hat{t}}$ to be solved at step $\hat{t}$ is defined in this section. Moreover, we introduce an approximate model for the further reduction of the computational complexity,

[3] As for x_{iu}, the variables with indices $u > t_i^{\mathrm{D}} - \hat{t}$ become fixed.

thus the computational cost. Therefore, the following is the definition of an approximated sub-problem, denoted *semi-approximate model*, which we also call $\mathcal{M}_{\hat{t}}$ for simplicity.

$T^{\mathrm{S}}(< T)$ is a parameter introduced for *the semi-approximate model*. In the approximate model, $\mathcal{M}_{\hat{t}}$ is divided into the former part and the latter part by a temporal axis. T^{S} is the temporal length of the former part, for which all the variables and constraints are rigorously treated, to obtain a detailed plan. The remaining is the latter part, for which the constraints for delivery problem $\mathcal{Q}$ are all removed, and the relating variables are all deleted, to obtain an approximate plan. In short, *the semi-approximate model* describes the former part with length T^{S} precisely, while the latter part is approximated without solving the corresponding $\mathcal{Q}$. In this connection, relax-and-fix heuristics originally proposed by Pochet et al. (Pochet & Vyve 2004) used a linear relaxation for *the semi-approximate model* of $\mathcal{M}_{\hat{t}}$.

According to the procedure described in **3.1**, at step $\hat{t}$, the variables with indices $t \in [1,\, \hat{t}-1]$ have already been fixed by the preceding step. Then, $\mathcal{M}_{\hat{t}}$ is divided into the former part with $t \in [\hat{t},\, \hat{t}+T^{\mathrm{S}}-1]$ and the latter part with $t \in [\hat{t}+T^{\mathrm{S}}, T]$. The former part has the same variables, constraints and evaluations as $\mathcal{M}$ ($\mathcal{P}$ and $\mathcal{Q}$). On the contrary, the latter part has only the variables, constraints and evaluations which belong to $\mathcal{P}$, and those for $\mathcal{Q}$ are all deleted. Instead, an approximate evaluation for $\mathcal{Q}$ is introduced, which can be calculated from the solution of $\mathcal{P}$. Here, we use the total load size f^{W} and the number of visited facilities f^{G} as an approximation of the original evaluation f^{V}_t, the delivery cost.

$\mathcal{M}_{\hat{t}}$ is defined by the followings:

minimize

$$w^{\mathrm{K}} f^{\mathrm{K}} + w^{\mathrm{B}} f^{\mathrm{B}} + w^{\mathrm{V}} \sum_{t=\hat{t}}^{\hat{t}+T^{\mathrm{S}}-1} f^{\mathrm{V}}_t + w^{\mathrm{W}} f^{\mathrm{W}} + w^{\mathrm{G}} f^{\mathrm{G}}, \tag{24}$$

subject to

Eqs. (3)–(9),

Eqs. (10)–(23), while $t = \hat{t}, \ldots, \hat{t}+T^{\mathrm{S}}-1$,

$$f^{\mathrm{W}} = \sum_{t=\hat{t}+T^{\mathrm{S}}}^{T} \sum_{\substack{i=1,\\ t^{\mathrm{D}}_i > t}}^{N} p_i x_{i,t^{\mathrm{D}}_i - t} + \sum_{t=\hat{t}+T^{\mathrm{S}}}^{T} \sum_{i=1}^{N} p_i m_i d_{it}, \tag{25}$$

$$f^{\mathrm{G}} = \sum_{t=\hat{t}+T^{\mathrm{S}}}^{T} v_t, \tag{26}$$

$$x_{iu} = \tilde{x}_{iu}, \qquad u > t^{\mathrm{D}}_i - \hat{t},\ i = 1, \ldots, N,\ u = 0, \ldots, T-1 \tag{27}$$

$$y_{ik\ell t} = \tilde{y}_{ik\ell t}. \quad i = 1, \ldots, N,\ k = 1, \ldots, M,\ \ell = 1, \ldots, L,\ t = 1, \ldots, \hat{t}-1 \tag{28}$$

d_{it} in Eq.(25) is defined as 1 if the due date of O_i is t, 0 otherwise. w^{W} and w^{G} in Eq.(24) are non negative values for the weights. $\tilde{x}$ and $\tilde{y}$ on the right-hand sides of Eqs.(27) and (28) are *fixed variables*, the values of which are obtained by the preceding step of $\hat{t}$.

Figure 3 illustrates the structure of $\mathcal{M}_{\hat{t}}$ in the similar style with Figure 2. x_{iu} which satisfies $u = t_i^{D} - t$ is described by a circle located in column t. Table 1 lists the variables and the objective functions in $\mathcal{M}_{\hat{t}}$.

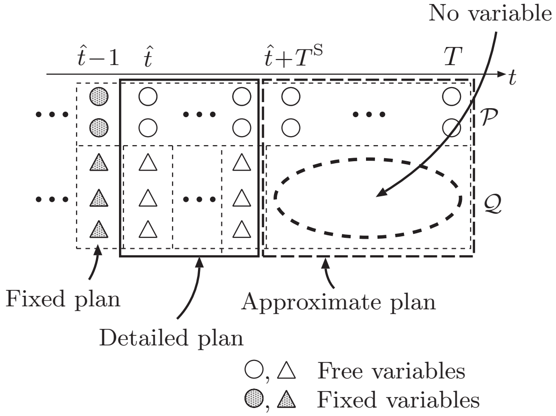


Fig. 3. *Structure of the semi-approximate model.*

Table 1. *Semi-Approximate Model $\mathcal{M}_{\hat{t}}$.*

	Detailed plan	Approximate plan
x_{iu}, v_t, v_t^{B}	included	included
f^{K}, f^{B}	included	included
$y_{ik\ell t}$, r_{it}^{SK}, r_{it}^{SF}, r_{it}^{KF}, $y_{ik\ell t}^{SK}$, $y_{ik\ell t}^{SF}$, $y_{ik\ell t}^{KF}$, $z_{k\ell t}$	included	not included
f_t^{V}	included	not included
f^{W}, f^{G}	not included	included

In the approximate plan for the latter part $[\hat{t} + T^{S}, T]$, the decision variables $y_{ik\ell t}$ are deleted together with their constraints and the variables depending on them. Thus, $\mathcal{M}_{\hat{t}}$ may derive an infeasible solution because of its relaxed or deleted variables, but it is a tentative solution of a sub-model obtained through the incremental process described in the previous section.

The relaxed variable, which may take an infeasible value, always remains a free variable in the present incremental step $\hat{t}$, and it is again included in the next updated sub-problem $\mathcal{M}_{\hat{t}+1}$ to be solved in the next step. When the iterative procedure has completed, all the variables become fixed variables and take feasible values. On the other hand, the number of integer variables in a sub-problem is reduced by a semi-approximated model, which becomes proportional to parameter T^{S}. Therefore, we can control the balance between the computational cost and the quality of the solution by choosing T^{S} value according to the requirement in the application.

The proposed idea to incrementally fix the solution can be easily extended to the infinite T problem. Then, sub-problem $\mathcal{M}_{\hat{t}}$ has the fixed length T^{P} ($T^{\mathrm{S}} < T^{\mathrm{P}}$), and $\mathcal{M}_{\hat{t}}$ is composed of the former part of length T^{S} and the latter part of length $T^{\mathrm{P}} - T^{\mathrm{S}}$. This $\mathcal{M}_{\hat{t}}$ with fixed length $T^{\mathrm{P}}$is also applicable to the finite T problem ($T^{\mathrm{P}} < T$), to reduce the computational complexity. Figure 4 illustrates an example with $T^{\mathrm{S}} = 2, T^{\mathrm{P}} = 5$ and $T = 6$. The figure also shows how the variables for each day are fixed, namely, the plan for each day is obtained, by each step of the iteration. Thus, this method produces the plan for each day incrementally, taking into account the future plan for the next T^{P} days.

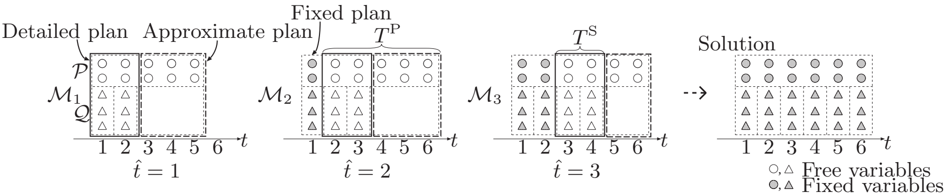


Fig. 4. *Incremental procedure using the semi-approximate model ($T^{\mathrm{P}} = 5$, $T^{\mathrm{S}} = 2$).*

4. COMPUTATIONAL EXPERIMENTS

Numerical experiments on several examples are executed to study the performance of the proposed algorithm. The following results are obtained by the computations executed on a Xeon 3.16GHz PC with 32GB of RAM, and CPLEX11 (IBM 2008) is used as a mathematical programming solver.

4.1. PERFORMANCE EVALUATION OF THE PROPOSED METHODS

Four examples E1,. . .,E4 are prepared as the storage and delivery problem. Virtual data are used for the preliminary study on the performance of the proposed algorithm. The parameters which define the problem size are given as shown in Table 2, including the number of integer variables. And the distribution of the daily total size of all items v_t, which are shipped from the suppliers, is given according to the real data obtained from one Japanese automobile company for one week. That is, the distribution with the same average and variance with the real size distribution is used to generate the data in the four examples. The number of the candidate routes L is 30. Each route connects no more than 3 facilities with the shortest path, and each facility is ensured to be included in multiple routes. The weighting factors $\boldsymbol{w}$ in the objective function are chosen empirically so that each of five terms in the objective function takes a comparative value. That is, as the ratios of the five objective values f^{K}, f^{B}, f^{V},

f^{W} and f^{G} are found to be approximately 20 : 5000 : 1000 : 300 : 1 in the present examples, therefore, w^{K}, w^{B}, w^{V}, w^{W} and w^{G} are set inversely proportional to the ratios; 1/20, 1/5000, 1/1000, 1/300 and 1, respectively.

Table 2. *Parameters of Examples E1,. . .,E4.*

	T	N	M	L	Number of integer variables
E1	7	10	4	30	34479
E2	7	15	5	30	64106
E3	14	25	8	30	339566
E4	14	30	8	30	406802

4.1.1. BASIC PERFORMANCE AND $\boldsymbol{T}^{\mathbf{S}}$ DEPENDENCY

First, we evaluate the performance of the proposed method by comparing with the following two methods, which also solve a sub-problem by a semi-approximated model but with different ways of approximation:

Method(a): Semi-approximate model without f^{W} and f^{G} for the evaluation

Method(b): Semi-approximate model by a linear relaxation

The solutions are also obtained by branch-and-bound algorithm using CPLEX for the comparison. The optimal values are obtained for the smaller size examples E1 and E2, but not for the larger size examples E3 and E4 due to the running out of the computer memory. Parameter T^{S} dependency is also studied for all methods, while parameter T^{P} is set T for all cases.

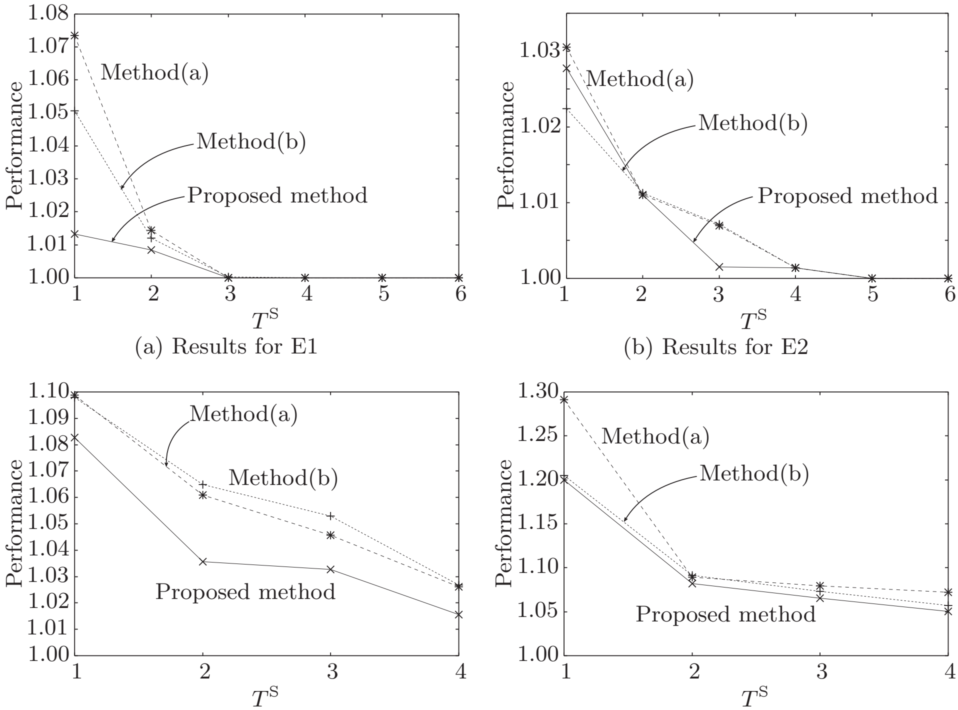
Figure 5 (a),. . .,(d) give the results for E1,. . .,E4, respectively. The vertical axis shows the performance by the objective function value, which is normalized by the optimal value. For the larger size examples E3 and E4, the best value obtained through 150,000 seconds computational time by CPLEX is used for the normalization. Therefore, the smaller value indicates the better performance, with the best at 1.0. The horizontal axis is T^{S}, which ranges from 1 to 6 for the small size examples, and from 1 to 4 for the larger size examples.

Figure 5 shows the following results:

(i) The proposed method gives the better results compared with Method(a) in all cases. This shows the effectiveness of the approximate evaluations f^{W} and f^{G} for $\mathcal{Q}$.

(ii) The proposed method gives similar or better results compared with Method(b). This shows the effectiveness of the proposed approximate model.

(iii) The results are improved with increasing T^{S} for all methods. Especially, the optimal values are acquired by the proposed algorithm with $T^{\mathrm{S}} \geqslant 5$ for the small size examples. This shows the proposed approximate model becomes precise enough by increasing T^{S}.



(a) Results for E1 (b) Results for E2

(c) Results for E3 (d) Results for E4

Fig. 5. *Comparison with other methods and T^{S} dependency of the performance.*

4.1.2. COMPUTATIONAL TIME

Next, we compare the computational time of the proposed method by comparing it with Method(b) and the branch-and-bound method.

Table 3 shows the results for the same examples as in **4.1.1**, which lead to the following observations:

(iv) Computational time grows exponentially with increasing T^{S} for all methods.

(v) Both the proposed method and Method(b) require shorter computational time compared with the branch-and-bound method for the larger size examples.

(vi) The proposed algorithm requires longer computational time compared with Method(b) in all cases. This is caused by the integer variables contained in the proposed approximate model, which Method(b) avoids by a linear relaxation.

Table 3. *Comparison of Computational Time.*

T^{S}	1	2	3	4	5	6
(a) Results for E1						
Proposed method	2.22	9.57	28.30	70.55	356.26	1187.70
Method(b)	1.81	12.93	31.58	58.53	415.83	1679.01
Branch-and-bound	1135.64					
(b) Results for E2						
Proposed method	7.29	62.94	322.93	1621.60	27462.18	33704.42
Method(b)	7.79	97.04	513.39	2028.93	52946.98	355219.20
Branch-and-bound	173416.46					
(c) Results for E3						
Proposed method	114.91	560.28	1379.86	4268.52		
Method(b)	50.70	219.21	960.69	2707.48		
Branch-and-bound	150000					
(d) Results for E4						
Proposed method	163.66	887.46	3185.88	8392.51		
Method(b)	54.19	363.47	2138.71	4199.27		
Branch-and-bound	150000					

4.1.3. $\boldsymbol{T}^{\mathbf{P}}$ DEPENDENCY

Finally, we evaluate the performance of the proposed method for a larger size example with T^{P} and T^{S} dependencies. Here, E5: $T = 30$, $N = 55$, $M = 5$ and $L = 30$, is used as a larger size example. Figure 6 shows that performance is improved with increasing T^{P} at each fixed T^{S}. This implies the effectiveness of the approximate model with a longer planning period T^{P}, by which $\mathcal{P}$ can produce a longer storage plan, though $\mathcal{Q}$ is approximated.

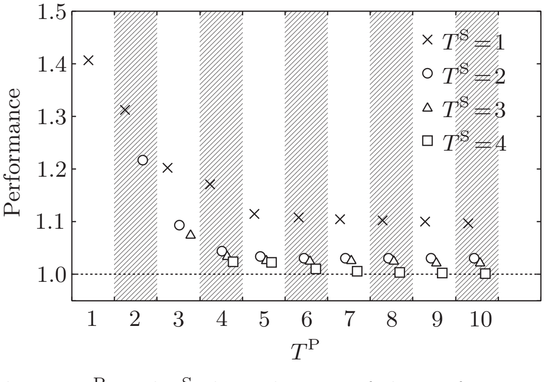


Fig. 6. *T^{P} and T^{S} dependencies of the performance.*

4.2. PERFORMANCE IN DYNAMIC ENVIRONMENTS

In this section, we evaluate the performance of the proposed algorithm in a dynamic situation, where the delivery order is changed from the original request before the actual delivery day. Here we only consider the change in the number of the items, namely m_i of order O_i. The modification of the number Δm_i from the original value m_i is informed during the incremental procedure of the proposed method. It can be informed only after the original order day t_{0i} and before the due date t_i^{D}. Then, let us consider the occurrence distribution of the modification Δm_i during the period $[t_{0i}, t_i^{\mathrm{D}}]$. We consider two types of the temporal distribution. Type $\mathcal{D}_1$: The modification occurs less frequently approaching the due date. Type $\mathcal{D}_2$: The modification occurs more frequently approaching the due date. Figure 7 illustrates the two types by linearly decreasing or increasing the function.

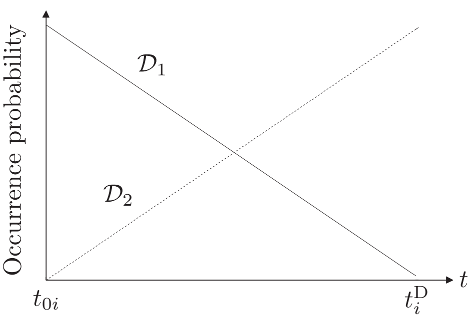
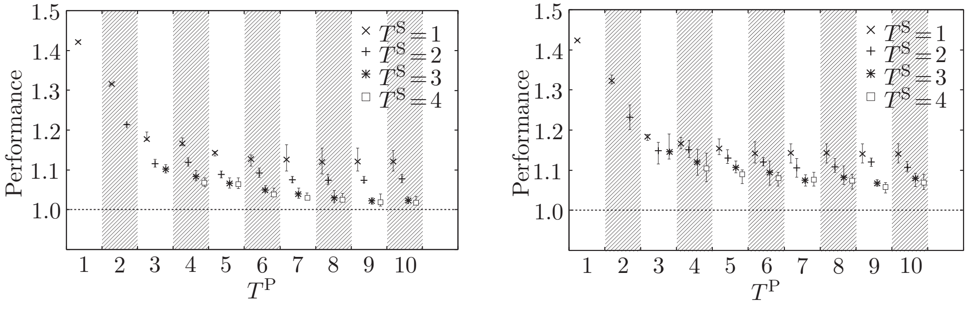


Fig. 7. *Probability distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ of the order modification.*

Larger size example E5 is used for the following computational experiments. Linear distribution $\mathcal{D}_1$ or $\mathcal{D}_2$ shown in Figure 7 is used as the temporal profile of the modification occurrence probability. According to the distribution, five realizations of the order modification are generated from E5, each of which becomes E5 after the modification. The modification amount Δm_i is given by the uniform distribution $[-0.2m_i,\ 0.2m_i]$.

Figure 8 shows the performance of the obtained plans for various values of T^{S} and T^{P}. Error bars indicate the variance among five realizations of the modification. First, the obtained solutions are feasible in all cases, though the modification can occur even after the first shipping for order O_i is started. Next, the Figure shows that performance is improved with increasing T^{P} at each fixed T^{S}, similar to the static environment shown in **4.1.3**. Moreover, Figure shows the better performance for $\mathcal{D}_1$ compared with for $\mathcal{D}_2$. This is because $\mathcal{D}_1$ causes less changes in a semi-approximate model compared with $\mathcal{D}_2$ during the incremental procedure. As it is known that the real logistics environment in the production planning undergoes $\mathcal{D}_1$ type changes in general, the proposed algorithm has the potential to robustly generate an efficient plan in the actual logistic systems.



(a) Results in environment $\mathcal{D}_1$ (b) Results in environment $\mathcal{D}_2$

Fig. 8. *T^P and T^S dependencies of the performance in the dynamic environments.*

5. CONCLUSIONS

We considered the logistic planning problem where the storage and the delivery are optimized simultaneously. First, we formulated the problem as a mixed integer programming (MIP) model, which is composed of the storage problem and the delivery problem. The storage problem ranges over multiple days, while the delivery problem can be solved for each day separately. Then, we applied relax-and-fix heuristics to optimize sub-problems incrementally by the mathematical programming method. Each incremental step generates the plan for each day. The sub-problem is described by a semi-approximate model, which partially approximates the delivery problem. The incremental approach and the approximate model lead to the effective reduction of the computational complexity, which becomes fatal for the large size problem. The computational experiments show that simultaneous optimization gives the effective storage plan to reduce the total logistic cost, and the proposed heuristics efficiently reduce the computational time and are robust against the dynamic situation.

As the further investigation, the semi-approximate model can be improved. And the robustness in the dynamic situation should be studied in more detail. Moreover, the substantial problem remains in the model extension to the multiple terminals, especially regarding computational complexity, which will be studied in the next step.

REFERENCES

Akartunali, K. & Miller, A. J. (2009), 'A heuristic approach for big bucket multi-level production planning problems', *European Journal of Operational Research* **193**(2), 396–411.

Campbell, J. (1990), 'Locating transportation terminals to serve an expanding demand', *Transportation Research* **B 24**(3), 173–193.

Chen, P., Guo, Y., Lim, A. & Rodrigues, B. (2006), 'Multiple crossdocks with inventory and time windows', *Computers & Operations Research* **33**(1), 43–63.

Comelli, M., Gourgand, M. & Lemoine, D. (2008), 'A review of tactical planning models', *Journal of Systems Science and Systems Engineering* **17**(2), 204–229.

Daganzo, C. (1996), *Logistics Systems Analysis*, Springer.

Ferreira, D., Morabito, R. & Rangel, S. (2009), 'Solution approaches for the soft drink integrated production lot sizing and scheduling problem', *European Journal of Operational Research* **196**(2), 697–706.

IBM (2008), *CPLEX11*, http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

Irnich, S. & Desaulniers, G. (2005), *Column Generation*, Springer-Verlag, chapter 2.

Kelly, J. D. & Mann, J. L. (2004), 'Flowsheet decomposition heuristic for scheduling: a relax-and fix method', *Computers and Chemical Engineering* **28**, 2193–2200.

Lawler, E. L., Lenstra, J. K., Kan, A. H. G. R. & Shmoys, D. B., eds (1985), *The Traveling Salesman Problem*, John Wiley & Sons.

Musa, R., Arnaout, J.-P. & Jung, H. (2010), 'Ant colony optimization algorithm to solve for the transportation problem of cross-docking network', *Computers & Industrial Engineering* **59**, 85–92.

Nemhauser, G., Kan, A. R. & Todd, M., eds (1989), *Handbooks in Operations Research and Management Science Volume 1*, Elsevier.

Onoyama, T., Maekawa, T., Kubota, S., Tsuruta, S. & Komoda, N. (2008), 'A selfish constraint satisfaction genetic algorithm for planning a long-distance transportation network', *Electronics and Communications in Japan* **91**(9), 1–10.

Pedroso, J. & Kubo, M. (2005), Hybrid tabu search for lot sizing problems, *in* 'Hybrid Metaheuristics, Proceedings', Vol. 3636 of *Lecture Notes in Computer Science*, Springer-Verlag, 66–77.

Pochet, Y. & Vyve, M. V. (2004), 'A general heuristic for production planning problems', *INFORMS Journal on Computing* **16**, 316–327.

Pochet, Y. & Wolsey, L. A. (2006), *Production planning by mixed integer programming*, Springer.

Sakakibara, K., Tian, Y. & Nishikawa, I. (2011), 'Mixed integer programming model and incremental optimization for delivery and storage planning using truck terminals (in Japanese)', *Transactions of the Institute of Systems, Control and Information Engineers* **24**(3), 88–96.

Taniguchi, E., Noritake, M., Yamada, T. & Izumitani, T. (1999), 'Optimal size and location planning of public logistics terminals', *Transportation Research Part E: Logistics and Transportation Review* **35**(3), 207–222.

Wu, T. & Shi, L. (2009), Hybrid nested partitions and relax-and-fix approach for capacitated multi-item lot sizing problem, *in* 'IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics', 359–364.