Author's Accepted Manuscript

An Integrated Strategy for a Production Planning and Warehouse Layout Problem: Modeling and Solution Approaches

Guoqing Zhang, Tatsushi Nishi, Sarina D.O. Turner, Keisuke Oga, Xindan Li

Cite this article as: Guoqing Zhang, Tatsushi Nishi, Sarina D.O. Turner, Keisuke Oga and Xindan Li, An Integrated Strategy for a Production Planning and Warehouse Layout Problem: Modeling and Solution Approaches, *Omega*, http://dx.doi.org/10.1016/j.omega.2016.06.005

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting galley proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

An Integrated Strategy for a Production Planning and
Warehouse Layout Problem: Modeling and Solution Approaches

Guoqing Zhang[1*], Tatsushi Nishi[2], Sarina D. O. Turner[3], Keisuke Oga[2], Xindan Li[4]

[1]Supply Chain and Logistics Optimization Research Center,
Department of Mechanical, Automotive & Materials Engineering, University of Windsor, Canada

[2]Graduate School of Engineering Science, Osaka University, Japan

[3]Department of Mechanical and Industrial Engineering, University of Toronto, Canada

[4]Research Center of Management Science and Engineering, Nanjing University, China

Abstract We study a real-world production warehousing case, where the company always faces the challenge to find available space for its products and to manage the items in the warehouse. To resolve the problem, an integrated strategy that combines warehouse layout with the capacitated lot-sizing problem is presented, which have been traditionally treated separately in the existing literature. We develop a mixed integer linear programming model to formulate the integrated optimization problem with the objective of minimizing the total cost of production and warehouse operations. The problem with real data is a large-scale instance that is beyond the capability of optimization solvers. A novel Lagrangian relax-and-fix heuristic approach and its variants are proposed to solve the large-scale problem. The preliminary numerical results from the heuristic approaches are reported.

Keywords: warehouse layout, capacitated lot-sizing, integer linear programming, Lagrangian heuristics

*Corresponding author: gzhang@uwindsor.ca, tel: 5192533000 ext 2637, fax: 519-973-7062.

1

1   Introduction

The warehouse layout problem is one of the key issues in warehouse management, which involves all stages of a supply chain. An effective warehouse layout can distinctly reduce the operation costs. The storage location assignment problem addressed in this paper is to determine the locations for different items in a warehouse; thus we consider warehouse layout and storage location assignment as interchangeable in this research. Production planning is the process of determining how to use and allocate the resources efficiently to satisfy the customer demands. The capacitated lot-sizing problem is a medium-term production planning, which decides how much to produce for each product so that the total cost of production, setup, and inventory is minimized. Usually, warehouse layout problems and production planning are analyzed separately. However, while warehouse space is one of the key resources for production, enhancing the utilization of the space can improve the production output and increase the manufacturer's profit.

Our research is motivated by a real-world production warehousing problem. We perform the study at a food company that processes and packages rice products. The company always faces the challenges of finding available space for their products and managing the items in the warehouse. The lack of coordination between production planning and warehouse management leads to significant inefficiency in both departments. Thus we investigate a production planning that considers the warehouse space resource and a warehouse layout with new storage assignment. In essence, we propose an integrated strategy to combine the capacitated lot-sizing problem with warehouse layout problem so that production process and warehousing can be coordinated effectively.

The warehouse layout problem and the capacitated lot-sizing problem have been studied extensively in research. However, to our knowledge, these two problems have been addressed separately in the current literature and they have not been considered together in a mathematical model. Jans and Degraeve (2008) review the modeling for industrial lot-sizing problems and discuss the four extensions of lot-sizing models. It is indicated in their discussion on new research directions that the integration of lot-sizing, scheduling, and storage constraints affects the planning and scheduling. Buschkühl *et al.* (2010) presents a review of four decades of research on dynamic capacitated lot-sizing and focuses on modeling and algorithm for the problems. Although they highlight that production planning is strongly related to the layout type and organizational structure of a production system, the warehouse layout is not taken into account in their reviewed literature on the capacitated lot-sizing problem. Existing literature indicates that most of the lot-sizing problems are hard to solve. Zhang and Lai (2010) review the multi-level warehouse layout problem with a dedicated storage policy without considering the lot-sizing problem.

2

Various researchers have combined the warehouse layout problem with different inventory models such as Economic Order Quantity (EOQ) models (e.g., Chu and Chu (2008), Wilson (1997), and Malmborg and Deutsch (1988)), and the replenishment policy models (e.g., Hassini (2008), and Kulturel *et al.* (1999)). It is noted that in existing research where the warehouse layout problem and inventory models are considered joinly, only static inventory models are utilized. The model presented in this paper not only combines a storage location assignment problem with dynamic inventory, but also takes into account production planning. To our knowledge, the dynamic joint multi-item storage location assignment combined with production planning has not been considered in the current literature.

The main contributions of the work include: 1) a new integrated strategy that combines storage location assignment and the capacitated lot-sizing problem, which were solved separately in previous research; 2) a new mixed integer programming (MIP) model for the joint problem; furthermore the model is dynamic, while previous models that have combined the storage location assignment problem with inventory models are static; 3) a novel Lagrangian relax-and-fix heuristic approach and its variants for solving the integrated optimization of production planning and warehouse layout problem, 4) a real-world industrial production warehouse problem, which is solved with the proposed methods and the results obtained could be applied to coordinate the production and warehousing at other similar facilities.

The remainder of the paper is organized as follows. Section 2 provides literature review on the warehouse layout and the capacitated lot-sizing problem. A real-world problem and its background are presented in Section 3. A mixed integer linear programming model is presented in Section 4. Lagrangian decomposition and coordination is introduced in Section 5. A Lagrangian relax-and-fix heuristic approach is proposed in Section 6. The two variants of the Lagrangian relax-and-fix heuristic method are discussed in Section 7. The computational results for test problems and the preliminary numerical results from the proposed Lagrangian methods for the industrial problem are reported in Section 8. Section 9 concludes the paper with a discussion of future work.

2   Literature review

The storage location assignment/warehouse layout problem

Extensive literature has been conducted on warehouse design and operations, which examined the different operations, components and areas of improvement in warehousing. Here we review some existing work and literature related to warehouse layout problem and comparisons of different storage assignment policies. For a comprehensive review of research on warehouse operations, we refer readers to Gu *et al.*

(2007, 2010) and Francis *et al.* (1992). The warehouse layout problem is to determine the physical location in storage departments/zones for incoming items while taking into consideration the storage structure, the capacity and the storage/retrieve process and requirement. An efficient warehouse layout can reduce material handling cost and improve space utilization (De Koster *et al.* (2007), and Gu *et al.* (2007)).

According to the different rules used to assign items in a warehouse, a warehouse system can implement different layout policies, i.e., storage assignment policies. De Koster *et al.* (2007) describe the five storage assignment policies, including dedicated, random, class-based, turnover-based, and closest open location storage policies.

A considerable number of studies compared the storage assignment policies under various warehouse environments. Malmborg and Altassan (1998) examined a less-than-unit load warehousing system and developed models to compare dedicated storage using the cube per order index, and randomized storage using closest open location. Randomized storage only achieved a 65% space utilization compared with 100% space utilization achieved with dedicated storage. Zeng *et al.* (2002) examined a short term-and long-term plan to show different alternatives for reducing the time spent filling customer orders. For the short-term plan they used an activity-based storage philosophy. Petersen and Aase (2004) examined static random storage, volume-based storage, and class-based storage policies, and used a simulation model to compare the policies in a warehouse environment. It was shown that when the number of order increased, the percentage of savings for using each policy decreased. Muppani and Adil (2008) concluded that if classes were formed considering only handling costs, a dedicated storage assignment policy would produce the lowest costs, and if classes were formed based on space costs, then a completely randomized storage assignment policy yielded the lowest cost solution. However, they pointed out that when handling costs and space costs were considered together, a class-based storage assignment policy was optimal. Pohl *et al.* (2011) investigated the storage assignment in unit-load warehouses with non-traditional aisles, and concluded that warehouse design parameters that perform best under random storage also perform well under turnover-based storage. Çelk and Süral (2014) studied order picking performance under random and turnover-based storage policies in fishbone aisle warehouses. They provided simple heuristics for fishbone layouts and performed computational experiments in order to compare the performances of fishbone and traditional layouts under optimal routing. Guo *et al.* (2016) investigated the impact of required storage space on the warehouse performance with random, full turnover -based and class-based storage policies.

4

In the recent years, the class-based storage policy has attracted considerable attention. Larson *et al.* (1997) examined a class-based warehouse layout. The aim was to achieve effective use of floor space. The model developed by Muppani and Adil (2007) examined the implementation of a class-based storage assignment policy and its effect on storage space and material handling costs, and the model was solved using a branch and bound algorithm. Nishi and Konishi (2010) proposed beam search heuristics for optimization of floor-storage warehousing systems. Pan *et al.* (2014) proposed a model to estimate a travel time for a high-level picker-to-part system with class-based storage policies. Pan *et al.* (2015) studied the order batching problem in a pick-and-pass warehousing system.

Capacitated Lot-sizing Problem (CLSP)

The capacitated lot-sizing problem (CLSP) is one of the most important and difficult problems in production planning (Karimi *et al.* (2003)). The capacitated lot-sizing problem was considered to be NP hard by Florian and Klein (1971) and Bitran and Yanesse (1982). Many exact and heuristic solution methods have been developed to solve CLSP. Here we only provide a brief review of existing work on CLSP that takes into account inventory and warehouse capacity. For the comprehensive reviews of the models and algorithms on CLSP, the reader is referred to Karimi *et al.* (2003), and Robinson *et al.* (2009).

The CLSP has been considered with bounded inventories and direct application to the warehouse environment. Love (1972) was the first to examine a bounded inventory problem. Page and Paul (1976) considered the problem of maintaining inventory for multiple products when there is a restriction on the maximum inventory investment or on the maximum amount of warehouse space. Gutierrez *et al.* (2002) examined a relevant class of production inventory systems when the inventory levels were bounded. Liu and Tu (2008) examined a production planning problem where the inventory capacity was the limiting factor. They formulated the model and developed an algorithm that was considered to be O (T^2). The capacitated lot-sizing problem with bounded inventory has also been extended to include multiple products, e.g., Pochet and Wolsey (1991), Absi and Kedad-Sidhoum (2008), Nascimento *et al.* (2010). Hwang and Kang (2016) studied the lot-sizing problem with backlogging for stepwise transportation cost.

Warehouse capacity or layout can be a critical resource or issue of CLSP (Zhang *et al.* 2012). Chu and Chu (2008) examined the single item dynamic lot-sizing model with bounded inventory and outsourcing. The inventory was bounded by the storage capacity of the warehouse. Transchel and Minner (2009) analyzed the replenishment of multiple products to satisfy dynamic demand when the warehouse capacity or the available inventory budget was limited. A savings-based heuristic was suggested for the warehouse

scheduling problem, and three simple approaches to the replenishment of multiple products with dynamic demand and limited warehouse capacity were developed. Buschkühl *et al.* (2010) indicated that production planning and particularly lot-sizing is strongly relevant to the layout type and organizational structure, but no warehouse layout has been considered in the lot-sizing problem in their reviewed literature.

This literature review shows that previous research has been conducted for establishing the need to coordinate storage assignment with inventory control, and there are some studies combining storage assignment with static EOQ and replenishment policies. Extensive research has been devoted to the storage location assignment problem and the capacitated lot-sizing problem separately. However, to our knowledge, no work has been reported that combines the storage location assignment problem with the capacitated lot-sizing problem. The study on the integration of the two problems is motivated by a real-world case where there is a limited production warehouse space. In the next section, the real-world problem and background information are introduced.

3   The Problem and Background

In the real-world production and warehousing problem, the company aims to reorganize their warehouse and determine if they have enough space to store each product to meet customer demand, while keeping up with current company operations and allowing for future growth. Figure 1 depicts the flow of items from the production area to the output/loading point of the facility. The areas of interest associated with each section of the process have been highlighted in Figure 1.
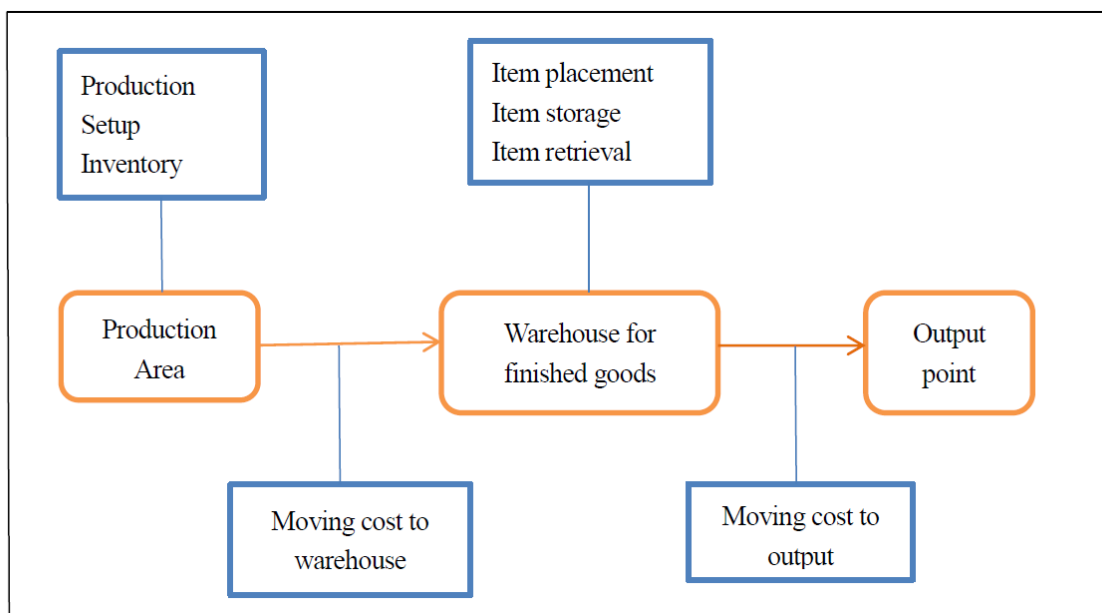

Figure 1. Item flow from production to warehousing to output and related costs

In the production area, there are eight production lines. The production planner decides on the lot size of all items at each period with capacity limits. The planning is concerned with production costs, setup costs, and inventory costs during the planning horizon. Setup costs are related to the amount of time an operator would spend preparing a machine for production. The operator is paid per hour and the salary is determined by the amount of time that the operator needs to perform the setup operation. This data is taken from the time study that was conducted, and the cost for each setup varies depending on the production line and the type of item that is going to be produced. Inventory is determined by the number of items that are in the warehouse at the end of the period. Demand is forecasted and known. However, the current production planning does not consider the availability of the warehouse space, which could lead to infeasible solutions that would exceed the warehouse capacity.

The transportation cost is incurred whenever a product is carried by a forklift truck from the production area to the storage locations. Each storage location has its own individual address that items could be delivered to, and the cost is proportional to the transportation distance in feet by the forklift operator from the production area to that specific storage location.

A dedicated storage assignment policy is used to organize the warehouse space, by assigning locations to each item. A dedicated storage assignment policy also has industrial relevance to the warehouse as it allows order pickers to become familiar with item locations, and it typically reduces the material handling costs. The optimal storage locations are determined based on cost: this cost is proportional to the transportation distance in feet by the forklift operator during the placement and retrieval of items in the warehouse. The cost is based on the wage of the operator, order processing, labeling, and shrink wrapping of the items.

The travel costs associated with the movement of product from the storage locations to the output point and from the production area to the storage locations are calculated based on operator travel. The only difference is the starting and ending points which alter the cost of travel. Therefore, in most cases the distance from the production area to the storage locations differs from the distance from the storage locations to the loading/output point.

At the output point, the demand is the order from the customers for each item and is considered as the pull mechanism in the item flow process. The forecasted demand, which is known, determines how many

7

items would flow to the loading point at each period to be shipped to customers. This is where the item's flow ended in the model.

The challenge faced by the company is the difficulty in finding available space during periods when the products are transferred from production area to the warehouse. Occasionally, because of misplaced items due to employee errors or lack of visibility, the operator cannot find the items to be picked up even though they are available, which leads to extra production. Several solution approaches are investigated, including reorganizing the warehouse and increasing visibility. More importantly, a proposal to combine the production planning with the storage location assignment is presented and leads to this research. It is noted that the storage location assignment can be adjusted annually, which is actually the practical way in the company. Thus the time horizon for both production planning and storage location assignment is one year. This paper develops an integrated strategy that combines a dedicated storage location assignment with the capacitated lot-sizing problem into a single mathematical model that minimize the total cost of travel, reserved storage space, handling, production, inventory holding, and setup costs. The formulation of the problem is developed in the next section.

4   Model

4.1. Model Assumptions

The basic assumptions of this model are as follows:

- Demand is forecasted and known, and shortages are not permitted.

- A dedicated warehouse layout policy is utilized.

- The time horizon for both production planning and storage location assignment is the same. For the real-world problem, the horizon is one year.

- All items are stored and moved on pallets. Pallets are considered to be the same size, weight, and geometric configuration and these factors had no effect on the storage and handling costs.

- A discrete number of warehouse storage locations are used.

- Items are delivered and retrieved using a single-command forklift truck.

- One unit of item accounted for a column of three pallets.

- There is one general production area that products come from, and has one output point.

- Costs associated with the placement and retrieval are directly proportional to the transportation distance.

8

- All costs of production, set up, and holding are known, and the value of variable unit space is set to 1.

- Setup costs are proportional to the time spent by the operator to prepare for each production.

4.2. Notations

The parameters of the proposed model include the number of items, storage locations, the planning horizon, forecasted demand, as well as the costs of production, setup, storage, handling, travel costs, and reservation cost. The notations for those parameters are defined as the following.

Indices:

i : the index of an item.

t : the index of a period within the planning horizon

l : the index of a storage location within the warehouse.

The parameters used in the model are:

R^l : Unit cost of reserving storage location l.

O^l : Unit cost of moving a column of any item from storage location l to the output point.

P^l : Unit cost of moving a column of any item from the production area to storage location l.

h_t^i : Unit inventory cost of holding item i at period t.

c_t^i : Variable unit production cost of item i at period t.

u_t^i : Unit setup cost of item i at period t.

d_t^i : The amount of required demand of item i in period t. This parameter denotes the demand of each
item, which varies from product to product and period to period.

v_t^i : The variable capacity of item i at period t.

f_t : A key resource constraint on production, for example, budget, production, labor, etc.

M : A big number, used for logic constraints.

The problem includes lot-sizing decision and storage location assignment. The following are the decision variables of the model:

x_t^i : The quantity of item i produced during period t.

s_t^i : The inventory level for item i at the end of period t.

y_t^i : 1, if item i is produced during period t; 0, otherwise.

w_t^{il} : 1, if item i is moved from the production area and placed in storage location l during period t; 0, otherwise.

q_t^{il}: 1, if item i is requested (by demand) from location l during period t; 0, otherwise.

9

n_t^{il} : 1, if item i is inventoried in location l during period t; 0, otherwise

z^{il} : 1, if location l is reserved for item i for the planning horizon; 0, otherwise.

Note: Each location can only hold one unit at a time.

4.3. Formulation

The integrated problem is formulated as the following mixed integer programming model.

4.3.1. Objective Function

$$Z = \sum_{i=1}^{I} \sum_{l=1}^{L} R^l z^{il} + \sum_{i=1}^{I} \sum_{l=1}^{L} \sum_{t=1}^{T} P^l w_t^{il} + \sum_{i=1}^{I} \sum_{l=1}^{L} \sum_{t=1}^{T} O^l q_t^{il} + \sum_{i=1}^{I} \sum_{t=1}^{T} \left(c_t^i x_t^i + u_t^i y_t^i + h_t^i s_t^i \right) \tag{1}$$

The objective function is divided into four terms: the first term is the cost of reserving locations for items, in a dedicated storage policy. The second term is the cost associated with the travel of products from the production area to assigned storage locations. The third term is the cost associated with the transportation of products from the storage locations to the output point, which occurs once a product is demanded during a given time period. The fourth term is all the costs associated with production planning, production, setup, and holding inventory. The objective is to find the optimal production planning and the best locations of the products in order to minimize the total costs.

4.3.2 Constraints

$$\sum_{i=1}^{I} z^{il} \leq 1, \quad \forall l \tag{2}$$

Constraints (2) limit the number of items that could be assigned to a reserved location to one. They ensure that a location can only be reserved by one item.

$$\sum_{i=1}^{I} q_t^{il} \leq 1, \quad \forall l, \forall t \tag{3}$$

Constraints in (3) ensure that at most one item can be requested at each storage location in each period.

$$\sum_{i=1}^{I} w_t^{il} \leq 1, \quad \forall l, \forall t \tag{4}$$

Constraints in (4) state at most one item can be moved from production area to a specific storage location l in period t.

$$\sum_{i=1}^{I} n_t^{il} \leq 1, \quad \forall l, \forall t \tag{5}$$

Constraints in (5) state that at most one item can be inventoried at each storage location in each period.

10

$$\sum_{l=1}^{L} q_t^{il} = d_t^i \quad \forall i, \forall t \tag{6}$$

Constraints (6) state that the number of products requested from storage locations during period t is equal to the demand of that product during period t.

$$\sum_{l=1}^{L} w_t^{il} = x_t^i \quad \forall i, \forall t \tag{7}$$

Constraints (7) ensure that then number of products moved from production area to the warehouse in a period is the same as what produced in the period.

$$\sum_{l=1}^{L} n_t^{il} = s_t^i \quad \forall i, \forall t \tag{8}$$

Constraints (8) state that all the locations that items remain in at the end of time period t are consistent with the inventory at the end of time period t.

$$q_t^{il} \le w_t^{il} + n_{t-1}^{il} \quad \forall i, \forall l, \forall t \tag{9}$$

Constraints in (9) state that items could only be retrieved from locations which they have previously been moved in from production area or inventoried.

$$w_t^{il} + n_{t-1}^{il} \le z^{il} \quad \forall i, \forall l, \forall t \tag{10}$$

Constraints in (10) state that when moving an item from production area to the warehouse, the item is only can be placed at an available storage location. Therefore, no item could be assigned to a location where an item has already been inventoried in the previous period.

$$n_t^{il} = w_t^{il} - q_t^{il} + n_{t-1}^{il} \quad \forall i, \forall l, \forall t \tag{11}$$

Constraints (11) are the flow balance of an item from the production area to a storage location, and from the storage location to the output point.

$$q_t^{il} \le z^{il} \quad \forall i, \forall l, \forall t \tag{12}$$

Constraints (12) ensure that items are only retrieved from locations that had been reserved for them.

$$w_t^{il} \le z^{il} \quad \forall i, \forall l, \forall t \tag{13}$$

Constraints (13) ensure that items are only placed in locations that had been reserved for them.

$$n_t^{il} \le z^{il} \quad \forall i, \forall l, \forall t \tag{14}$$

Constraints (14) ensure that items are only inventoried in locations that have been reserved for them.

$$\sum_{i=1}^{I}\sum_{l=1}^{L} v_t^i \cdot w_t^{il} \le f_t \quad \forall t \tag{15}$$

Constraints (15) are production capacity restrictions. The constraints ensure that the number of each item produced multiplied by the variable capacity of each item did not exceed any key resource during the planning horizon.

$$\sum_{l=1}^{L} w_t^{il} \le M y_t^i \quad \forall i, \forall t \tag{16}$$

Constraints (16) are set up constraints; the constraints put a limit on production during each period. The value of M is fixed in terms of both production capacity and item demands.

$$z^{il}, w_t^{il}, q_t^{il}, n_t^{il}, y_t^i \in \{0,1\} \quad \forall i, \forall l, \forall t \tag{17}$$

The model described by (1) to (17) is a mixed integer programming model. Similar to most capacitated lot-sizing problems, the problem for the real-world data is very difficult to solve.

A simpler model can be established by substituting $q_t^{il} = w_t^{il} + n_{t-1}^{il} - n_t^{il}$ into (9) leads $n_t^{il} \ge 0$ $\forall i, \forall l, \forall t$ which is clearly satisfied. Then, constraints (9) are eliminated. Constraints (6) can be rewritten as $\sum_{l=1}^{L} w_t^{il} + n_{t-1}^{il} - n_t^{il} = d_t^i$ $\forall i, \forall t$. Similarly, constraints (3) and (12) are satisfied by (2), (9) and (10). Constraints (17) are also satisfied since w_t^{il} and n_t^{il} are binary variables. Since $z^{il} \in \{0,1\} \forall i, \forall l$, constraints (13) and (14) are always satisfied by (10) since $n_t^{il} \ge 0$ $\forall i, \forall l, \forall t$. All together implies the following simplified model.

Objective function:

$$Z = \sum_{i=1}^{I}\sum_{l=1}^{L} R^l z^{il} + \sum_{i=1}^{I}\sum_{l=1}^{L}\sum_{t=1}^{T}\left((P^l + c_t^i)w_t^{il} + O^l\left(w_t^{il} + n_{t-1}^{il} - n_t^{il}\right) + h_t^i n_t^{il}\right) + \sum_{i=1}^{I}\sum_{t=1}^{T}\left(u_t^i y_t^i\right) \tag{18}$$

Subject to constraints (2), (4), (5), (6), (10), (11), (15), (16),

$$z^{il}, w_t^{il}, q_t^{il}, n_t^{il}, y_t^i \in \{0,1\} \quad \forall i, \forall l, \forall t \tag{19}$$

The approaches for solving the problem are discussed in the next section.

5   LAGRANGIAN DECOMPOSITION AND COORDINATION ALGORITHM

The Lagrangian decomposition and coordination method is one of the widely used approaches for capacitated lot-sizing problem (Buschkühl *et al.*, 2010), integrated optimization of production scheduling and distribution planning (Nishi *et al.*, 2007), supply chain coordination (Nishi et al., 2008), multi-product newsvendor problem with supply discount (Zhang, 2010), and so on. The key idea of this method is to relax the coupling constraints through

Lagrangian multipliers and to decompose the original problem into some simple subproblems. Given the Lagrangian multipliers, the relaxed problem provides a lower bound for the optimal primal objective value in a minimization problem. Generally, by means of updating the Lagrangian multipliers effectively, the lower bound can be improved gradually.

5.1. Lagrangian decomposition and coordination method

Applying a Lagrangian decomposition to the model, we choose to relax coupling constraints (2), (4), (5), and (15). This leads to a decomposition with as many independent subproblems as items.

Using non-negative Lagrange multipliers $\alpha^l, \beta_t^l, \pi_t^l$ and μ_t, we can generate the following relaxed problem (LRP). Let the set of items, the set of periods and the set of locations be I, T, L, respectively.

(LRP)

Minimize $L\left(z^{il}, w_t^{il}, q_t^{il}, n_t^{il}, y_t^i \right)$, with

$$
L\left(z^{il}, w_t^{il}, q_t^{il}, n_t^{il}, y_t^i \right) = \sum_{i=1}^{I} \sum_{l=1}^{L} R^l z^{il} + \sum_{i=1}^{I} \sum_{l=1}^{L} \sum_{t=1}^{T} \left((P^l + c_t^i) w_t^{il} + O^l \left(w_t^{il} + n_{t-1}^{il} - n_t^{il} \right) + h_t^i n_t^{il} \right)
$$

$$
+ \sum_{i=1}^{I} \sum_{t=1}^{T} \left(u_t^i y_t^i \right) + \sum_{l=1}^{L} \alpha^l \sum_{i=1}^{I} \left(z^{il} - 1 \right) + \sum_{l=1}^{L} \sum_{t=1}^{T} \beta_t^l \sum_{i=1}^{I} \left(w_t^{il} - 1 \right) + \sum_{l=1}^{L} \sum_{t=1}^{T} \pi_t^l \sum_{i=1}^{I} \left(n_t^{il} - 1 \right)
$$

$$
+ \sum_{t=1}^{T} \mu_t \sum_{i=1}^{I} \left(\sum_{l=1}^{L} v_t^i w_t^{il} - f_t \right) \tag{20}
$$

subject to constraints (6), (10), (11), (16), (19).

The problem (LRP) can be decomposed into independent subproblems, SP_1, $SP_2, \cdots, SP_{|I|}$, as follows.

(SP_i)

Minimize $L_i\left(z^{il}, w_t^{il}, q_t^{il}, n_t^{il}, y_t^i \right)$, with

$$
L_i = \sum_{l=1}^{L} R^l z^{il} + \sum_{l=1}^{L} \sum_{t=1}^{T} \left((P^l + c_t^i) w_t^{il} + O^l \left(w_t^{il} + n_{t-1}^{il} - n_t^{il} \right) + h_t^i n_t^{il} \right) + \sum_{t=1}^{T} \left(u_t^i y_t^i \right)
$$

$$
+ \sum_{l=1}^{L} \alpha^l z^{il} + \sum_{l=1}^{L} \sum_{t=1}^{T} \beta_t^l w_t^{il} + \sum_{l=1}^{L} \sum_{t=1}^{T} \pi_t^l n_t^{il} + \sum_{t=1}^{T} \mu_t v_t^i x_t^i \tag{21}
$$

subject to constraints (6), (10), (11), (16), (19).

13

Problem SP_i is equivalent to the single item production planning and layout problem. This problem is a mixed integer linear programming problem and can be solved optimally using standard optimization software.

5.2. Subgradient optimization

To improve the lower bound, we update the Lagrangian multipliers by solving the following Lagrangian dual problem.

(LDP)

$$\max_{\alpha^l, \beta^l_t, \pi^l_t, \mu_t} \min L\left(z^{il}, w^{il}_t, q^{il}_t, n^{il}_t, y^i_t\right)$$

subject to the constraints in problem LRP, where $\min L\left(z^{il}, w^{il}_t, q^{il}_t, n^{il}_t, y^i_t\right)$ is concave and piecewise linear continuous to $\alpha^l, \beta^l_t, \pi^l_t$ and μ_t. We use the subgradient method to optimize the dual problem by updating the multipliers at the kth iteration as follows. For example, the Lagrange multiplier $\alpha^{l,(k)}$ is updated by

$$\alpha^{l,(k+1)} = \alpha^{l,(k)} + \frac{\varphi\left(UB_{Heu} - L^{(k)}\right)\left(\sum_{i=1}^I z^{il} - 1\right)}{\sum_{l=1}^L \left(\sum_{i=1}^I z^{il} - 1\right)^2} \tag{22}$$

where $\alpha^{l,(k)}$ is the Lagrange multipliers at kth iteration. $UB_{Heu}, L^{(k)}$ are the value of upper bound and Lagrange function value at kth iteration respectively. φ is the step size which is determined by preliminary experimental results.

5.3. Construction of a feasible solution

The solution to the relaxed problem is usually infeasible to the original problem due to the relaxation of the coupling constraints. A feasible solution, i.e., upper bound, is constructed by the following heuristic algorithm. The heuristic algorithm successively checks the violation of constraints (2), (4), (5) and (15) and modifies the solution of the LRP problem. We also propose a new algorithm to obtain better upper bound in Section 6.

Heuristic algorithm

Step 1: Initialization.

Set the parameter $I_{plan} \leftarrow I, I_{error} \leftarrow \emptyset$.

Step 2: Remove those items with reserved space larger than needed: i.e., if there exists item i that satisfies the condition $\sum_{l \in L} z^{il} > \max_{t \in T} d^i_t$, item i is removed from the planning. $I_{plan} \leftarrow I_{plan} \setminus \{i\}$.

Step 3: Check if there are any items that do not satisfy the constraints (2). If item i does not satisfy (2),

then $I_{error} \leftarrow I_{error} \cup \{i\}$.

Step 4: For each item $i \in I_{error}$, if there exists an feasible storage location l_2, i.e., $\sum_{i \in I} z^{il_2} = 0$, then the item is reassigned to the location l_2; otherwise, the planning of item i is removed.

14

Update $I_{plan} \leftarrow I_{plan} \setminus \{i\}$. $I_{error} \leftarrow I_{error} \setminus \{i\}$. If $I_{error} = \phi$, go to Step5.

Step 5: Generate x_t^i, s_t^i, y_t^i for $i \in I_{plan}$ with fixed reserved location such that the demand for each item is

satisfied. Generate w_t^{il}, n_t^{il} for $i \in I_{plan}$ with fixed reserved location such that the constraints (4) - (14)

are satisfied.

Step 6: Check if there is any period t that does not satisfy the constraints (15). If the constraint is violated at period t,

then select the item i with $x_t^i > 0$ and set $w_{t+1}^{il} \leftarrow w_t^{il}$,

$n_{t+1}^{il} \leftarrow n_t^{il}$ for $i \in I_{plan}, l \in L$. Update $w_{t+1}^{il}, w_t^{il}, n_{t+1}^{il}, n_t^{il}, y_{t+1}^i, y_t^i$ for $i \in I_{plan}$.

If (15) is satisfied for all time horizon, go to Step 7.

Step 7: Calculation of upper bound. Obtain an upper bound (UB) for a feasible solution.

The violations of the constraints (2), (4) and (5) are checked at Step 3 and modified at Step 4. Constraint (15) is

checked and modified at Step 6.

5.4. Constraints for reducing the computation time

To improve the performance of the algorithm, the constraints for reducing the computation time are introduced

into the subproblem SP_i. From the flow conservation law, we derive the following constraints.

$$\sum_{l \in L} n_{t-1}^{il} \le d_t^i + \sum_{l \in L} n_t^{il} \ (i \in I, t \in T) \tag{23}$$

$$d_t^i \le \sum_{l \in L} w_t^{il} + \sum_{l \in L} n_{t-1}^{il} \ (i \in I, t \in T) \tag{24}$$

The maximum number of available locations for item i is calculated as $|L| - \sum_{i' \in I \setminus \{i\}} (\max_{t \in T} d_t^{i'})$. From the flow

conservation law and the location capacity, we derive the following constraints.

$$\sum_{l \in L} w_t^{il} + \sum_{l \in L} n_{t-1}^{il} \le |L| - \sum_{i' \in I \setminus \{i\}} (\max_{t \in T} d_t^{i'}) \ (i \in I, t \in T) \tag{25}$$

$$\sum_{l \in L} n_{t-1}^{il} \le |L| - \sum_{i' \in I \setminus \{i\}} (\max_{t \in T} d_t^{i'}) - d_{t-1}^i \ (i \in I, t \in T) \tag{26}$$

5.5. Overall optimization algorithm

The algorithm of the Lagrangian decomposition and coordination is denoted by LDC Algorithm.

LDC algorithm

Step 1: Initialization.

Set the parameters and Lagrange multipliers $\alpha^{l,(0)}, \beta_t^{l,(0)}, \pi_t^{l,(0)}, \mu_t^{(0)} \leftarrow 0$.

Step 2: Solving subproblem.

The subproblem SP_i is solved for each item $i \in I$. If the Lagrange multipliers are updated

once at Step 5, the Lagrange multipliers are updated by subgradient method (22) after solving

each subproblem.

Step 3: Construction of a feasible solution.

Generate a feasible solution from the solution of subproblems derived at Step 2 by heuristic

algorithm.

Step 4: Evaluation of convergence.

If the difference between the lower bound and upper bound are not updated for predefined time, end

the algorithm.

Step5: Update of Lagrange multipliers.

The Lagrange multipliers are updated by subgradient method (22) and return to Step 2.

6   LAGRANGIAN RELAX-AND-FIX HEURISTICS

The LDC algorithm presented in the last section does not generate a good feasible solution and upper bound. We improve the algorithm by proposing a fix heuristics based on COI rule, which is introduced later in this section. The idea of the fix heuristics is that the solutions of the subproblem of the item i described in Section 5.1 are fixed one by one (Ohga et al. 2013). The similar approach is applied to other capacitated lot sizing problems (Chen 2015). The main point is to add the constraints to reduce the feasible area so that better feasible solutions can be found.

6.1. Constraints to derive a feasible solution

We add the following constraints to SP_i. Let I_rest denote the set of items whose solutions are not fixed and $z_{(\mathrm{fix})}^{il}, w_{t,(\mathrm{fix})}^{il}, n_{t,(\mathrm{fix})}^{il}, y_{t,(\mathrm{fix})}^{i}$ denote fixed decision variables $z^{il}, w_t^{il}, n_t^{il}, y_t^{i}$.

$$z^{il} + \sum_{i \in I \setminus I_\mathrm{rest}} z_{(\mathrm{fix})}^{il} \leq 1, \ \ \forall i \in I_\mathrm{rest}, \forall l \tag{27}$$

$$w_t^{il} + \sum_{i \in I \setminus I_\mathrm{rest}} w_{t,(\mathrm{fix})}^{il} \leq 1, \ \ \forall i \in I_\mathrm{rest}, \forall l, \forall t \tag{28}$$

$$n_t^{il} + \sum_{i \in I \setminus I_\mathrm{rest}} n_{t,(\mathrm{fix})}^{il} \leq 1 , \ \ \forall i \in I_\mathrm{rest}, \forall l, \forall t \tag{29}$$

$$\sum_{l=1}^{L} v_t^i w_t^{il} + \sum_{i \in I \setminus I_\mathrm{rest}} \sum_{l \in L} v_t^i w_t^{il} \leq f_t , \ \ \forall i \in I_\mathrm{rest}, \forall t \tag{30}$$

We also add the following constraints to SP_i in order to make the solutions feasible regardless of the fixing order. The constraints which limit the available location for the rest item have to be considered when unfixed items influence its production planning and warehouse layout.

$$\sum_{l \in L} z^{il} + \sum_{i' \in I_{\text{rest}} \setminus \{i\}} \left(\max_{t \in T} d_t^{i'} \right) + \sum_{i \in I \setminus I_{\text{rest}}} \sum_{l \in L} z_{(fix)}^{il} \leq |L|, \ \forall i \in I_{\text{rest}} \tag{31}$$

6.2. Cube per-Order Index (COI) rule

The cube-per-order index (COI) rule is one of the classical and widely used dedicated layout policies (e.g., Hodgson and Lowe (1982), and Malmborg and Deutsch (1988)). Since the rule is used in our heuristic solution, a brief introduction and discussion are given as follows.

The COI rule of Heskett (1963) is defined as the ratio of the item's total space requirement to number of trips required based on the items demanded. The original heuristics consisted of locating the items with the lowest COI value closest to the input/output points, and putting items that combined a high turnover frequency with a low space requirement in the best storage locations in the warehouse. Items were then assigned to locations progressively farther away from the I/O point by increasing COI. Both Francis (1967) and Harmatuck (1976) proved the optimality of the COI rule for the assignment of products to storage locations to minimize travel distance during storage/retrieval. COI rule and variants are still widely employed for various warehouse layout problems, mostly combining with other approaches, which can be found in the recent literature on the topic, e.g, Malmborg and Krishnakumar (1989), Muppani and Adil (2008), and Zhang and Lai (2010).

We define COI value for item i as

$$COI_i = \frac{\sum_{l \in L} z^{il}}{\sum_{t \in T} d_t^i}. \tag{32}$$

The item with the lowest value of COI_i is selected and fixed in the Lagrangian relax-and-fix heuristics. It is expected that allocation based on the priority of COI rule leads to reduced total costs.

6.3. Overall optimization algorithm

The algorithm of the fix heuristics based on COI rule in combination with LDC algorithm is denoted by LDC-LF algorithm. The solution of the subproblem derived by the LDC algorithm is fixed according to the minimum value of COI value for item i.

LDC-LF algorithm

Step 1: Initialization.

Set the parameter, $I_{\text{rest}} \leftarrow I$.

Step 2: LDC.

Obtain the lower bound and the upper bound with LDC algorithm (Algorithm 1).

17

Step 3: Fixing the solution.

Select the item $i_{\min}$ that has the minimum COI_i of I_{rest}. Fix the solution of the $SP_{i_{\min}}$ with (27) -(31).

$I_{\text{rest}} \leftarrow I_{\text{rest}} \setminus \{i_{\min}\}$.

Step 4: Check if all items are fixed. If $I_{\text{rest}} \neq \emptyset$, go to Step 3.

Step 5: Evaluation of the upper bound.

The upper bound derived by the fixed solutions is compared with the one derived from LDC algorithm at

Step 2 and obtain a better upper bound.

7   A large scale neighborhood search based on decomposition structure

In Section 6, the better feasible solutions and the upper bound are obtained in a short time. In this section, we

further improve the LDC algorithm by using the neighbourhood search. The idea of the search is to locally optimize

subproblem, once a solution of the problem is available. This is based on the POPMUSIC (Taillard and Voß (2002),

Alvim *et al.* (2009), Ostertag *et al.* (2009)). These local optimizations are repeated until no improvements are found.

The main point is the definition of the subproblem. The definition of the subproblem is presented in the next section.

7.1. The definition of the subproblem

The original problem is decomposed into *I* subproblems by removing the coupling constraints (2), (4), (5) and

(15).   Several subproblems are chosen randomly. They are aggregated to build a subproblem by adding extended

(2), (4), (5) and (15). The formulation of the subproblem is the following when *r* parts are chosen. Let $J \in I$

$(r = |J|)$ denote the set of the selected items. Let $z_{(\text{pop})}^{il}, w_{t,(\text{pop})}^{il},\ n_{t,(\text{pop})}^{il}, y_{t,(\text{pop})}^{i}$ denote the variables of the

SP_i $(i \in J)$ and $z_{(\text{fixpop})}^{il}, w_{t,(\text{fixpop})}^{il}, n_{t,(\text{fixpop})}^{il}\ n_{t,(\text{fixpop})}^{il}, y_{t,(\text{fixpop})}^{i}$ denote the variables of the SP_i $(i \notin J)$.

The variables of the SP_i $(i \in J)$ are decision variables and the variables of the SP_i $(i \notin J)$ are constant in the

following formulation.

(ASP_r)

$$\min \quad \sum_{i \in J} \left\{ \sum_{l=1}^{L} R^l z^{il} + \sum_{l=1}^{L} \sum_{t=1}^{T} \left((P^l + c_t^i) w_t^{il} + O^l (w_t^{il} + n_{t-1}^{il} - n_t^{il}) + h_t^i n_t^{il} \right) + \sum_{t=1}^{T} (u_t^i y_t^i) \right\} \quad (33)$$

s.t $(6), (10), (11), (16), (19)$

$$\sum_{i \in J} z_{(\text{pop})}^{il} + \sum_{i \in I \setminus J} z_{(\text{fixpop})}^{il} \leq 1, \ \ \forall l \quad (34)$$

$$\sum_{i \in J} w_{t,(\text{pop})}^{il} + \sum_{i \in I \setminus J} w_{t,(\text{fixpop})}^{il} \leq 1, \ \ \forall l, \forall t \quad (36)$$

18

$$\sum_{i \in J} n_{t,(\text{pop})}^{il} + \sum_{i \in I \setminus J} n_{t,(\text{fixpop})}^{il} \le 1, \quad \forall l, \forall t \tag{37}$$

$$\sum_{i \in J} \sum_{l=1}^{L} v_t^i w_{t,(\text{fixpop})}^{il} + \sum_{i \in I \setminus J} \sum_{l=1}^{L} v_t^i w_{t,(\text{fixpop})}^{il} \le f_t, \quad \forall t \tag{38}$$

If the solution of ASP_r is updated, the current tentative solution is also updated.

7.2. Overall optimization algorithm

The algorithm of the large-scale neighborhood search based on POPMUSIC in combination with LDC algorithm is described as LDC-POP algorithm.

LDC-POP algorithm

Step 1: LDC.

Obtain the lower bound and the upper bound with LDC algorithm.

Step 2: Constructing the subproblem.

Select the *r* parts randomly and construct the subproblem ASP_r.

Step 3: Optimizing the subproblem and updating the solutions.

Optimize the subproblem ASP_r. If the solutions improve the objective of the original problem,

the solutions are updated.

Step 4: Evaluation of convergence.

If the number of the search iterations meets a predefined repeat count, give the upper bound as output.

Otherwise go to Step 2.

8   Numerical results

In our experiment, five test instances are solved with proposed model and approaches. As shown in Table 1, among those instances, four instances are developed to verify the proposed model and solution: two are of small size and another two are of medium size, which are similar to four of eight production lines. Fifth instance is from the original real-world case, which is a large-scale problem.

Table 1. Test instances

CASE	Items No	Storages No	Period No.
Small_1	5	10	5
Small_2	10	25	6
Medium_1	25	65	8
Medium_2	50	233	12
Large (industry case)	153	813	12

19

The size parameters for the small size instance 1 are $|I|$=5, $|L|$=10, and $|T|$=5. The main data of small-sized instance 1 is provided in Table 2. The other data are available upon request.   Due to the space limitation of the paper, we omit the detailed data of medium and large size instance. The data for all instances are available upon request.

Table 2. Data for the small instance 1

R^l:	4.24	u_t^i:	4.50~15.0
O^l:	3.51~5.51	d_t^i:	1~3
P^l:	8.14~9.06	v_t^i:	1
h_t^i:	2.94	f_t:	10
c_t^i:	5.82~6.73	M:	5

We first attempt to use GAMS/CPLEX to solve the five instances. Since the proposed model is a mixed integer programming, the branch and bound method by CPLEX is utilized to solve the problems. Table 3 shows computational results of the branch and bound method with CPLEX for small and medium instances, where S-, and M- represent the small and medium sizes of instances respectively. BB indicates the branch and bound method by CPLEX. CPLEX cannot obtain any feasible solution in 375 seconds for the second medium size instance but obtains a feasible solution with the gap of 0.60% after almost 1 hour. However, for the large scale-instance of the real-world case, the solver fails to gain any feasible solution due to the complexity and size of the model.

Table 3. Performance of a branch and bound method

CASE	LB [-]	UB [-]	GAP[%]	Time [s]
S-BB1	1130.31	1130.31	0.00	0.44
S-BB2	2711.43	2711.43	0.00	6.22
M-BB1	6784.46	6784.46	0.00	69.53
M-BB2	193510.3	194673.28	0.60	3593.13
Large (industry case)	-	-		*

*Process is terminated with memory over

We use the small and medium instances to verify the proposal of the model and solutions with practical situations. The solutions provide what to produce for each period, how many locations to assign to for each item, where the items are to be located and retrieved from, and how much of each item should be inventoried over a specified time horizon. Table 4 and Table 5 report the production planning and storage locations for small instance 1 respectively. From Table 5, we can see that item 1 uses location slots 1, 2, and 5 in all five months; item 3 uses location 10 in February and April. Due to space limit, the other solutions are not reported, but are available upon request.

Table 4. Production x(i,t): Results for production planning

	Jan	Feb	Mar	Apr	May
Item1	3	3	3	3	3
Item2	2	2	2	2	2
Item3	1	2	1	2	1
Item4	2	2	2	2	2
Item5	1	1	1	1	1

Table 5. Item Placement w(i,l,t): The storage locations during each period

	Jan	Feb	Mar	Apr	May
Item1.L1	1	1	1	1	1
Item1.L2	1	1	1	1	1
Item1.L5	1	1	1	1	1
Item2.L6	1	1	1	1	1
Item2.L7	1	1	1	1	1
Item3.L8	1	1	1	1	1
Item3.L10	-	1	-	1	-
Item4.L4	1	1	1	1	1
Item4.L9	1	1	1	1	1
Item5.L6	1	1	1	1	1

To solve the real-world case problem, the related data on demands and warehouse is collected. The parameter sets for this large instance are 153 items, 12 periods, and 813 storage locations. An attempt is made to solve the problem directly using GAMS/CPLEX but it fails to gain any feasible solution.

The computational experiments were executed to investigate the effectiveness of the proposed methods. The program code for the proposed method was coded by the Microsoft Visual C++ Express 2010 Express. SP_i for each item i was solved by the CPLEX12.6 (IBM ILOG). Although the Lagrangian heuristics developed aims at solving the large-scale instance, the five instance problems are solved with the heuristics to verify the algorithms. Our computational experiences show that the simplified model can save the computational times for both CPLEX solver and our heuristics. The following computational results are based on the simplified model.

Table 6 shows the computational results of the conventional method (LDC algorithm introduced in Section 5.5) and the proposed method (LDC-LF algorithm and LDC-POP algorithm proposed in Section 6.3 and 7.2

21

respectively). The convergence condition is set that the lower bound (LB) and the upper bound (UB) are not updated 3 times in small and medium cases. In large case, the LDC algorithm is stopped when the iteration is repeated 5 times. The GAP column reports the relative gap between upper bound and lower bound.

Table 6. Performance comparison between the conventional and the proposed method

CASE	LB[-]	UB[-]	GAP[%]	Iteration[-]	Total time[s]
S-LDC 1	1130.31	1130.31	0.00	40	11.67
S-LDC-LF 1	1130.31	1130.31	0.00	40	11.67
S-LDC-POP 1	1130.31	1130.31	0.00	5	11.99
S-LDC 2	2708.09	2738.13	1.39	50	45.49
S-LDC-LF 2	2708.09	2713.40	0.20	50	46.04
S-LDC-POP 2	2708.09	2711.43	0.12	100	54.40
M-LDC 1	6767.39	7006.63	3.78	30	102.06
M-LDC-LF 1	6767.39	6796.17	0.43	30	100.65
M-LDC-POP 1	6767.39	6784.71	0.26	400	153.30
M-LDC 2	192818.33	195199.68	1.24	10	243.36
M-LDC-LF 2	192818.33	195018.60	1.14	10	271.36
M-LDC-POP 2	192818.33	194290.36	0.76	400	375.42
L-LDC	421129.24	455088.85	8.06	5	1541.11
L-LDC-LF	421129.24	455088.85	6.79	5	1882.51
L-LDC-POP	421129.24	447625.31	6.29	5	1732.40

For the first small size instance, all developed algorithms can find an optimal solution, which is the same as the solution from CPLEX solver, while algorithms LDC-LF and LDC-POP obtain good solutions with the small gaps from 0.12% to 0.20% for the second small size instance. For the first medium size instance, the computation times for LDC-LF and LDC-POP are longer than CPLEX. The GAPs of the solutions for the two algorithms are 0.43% and 0.26%, respectively, which are very small. The second medium size instance is much larger than the first instance, and the computation times for LDC-LF and LDC-POP are much shorter than CPLEX (CPLEX cannot find any feasible solution in 375 seconds). The GAPs of the solutions for the two algorithms are 1.14% and 0.72%, respectively, which are small. For the large size case, the proposed algorithms can gain good feasible solutions while CPLEX fails to find any feasible solution due to limited memory. GAPs of the solutions from the three algorithms are from 6.29% to 8.06%, while the running times are around 30 minutes.

Among those algorithms, the performance LDC-LF and LDC-POP is better than that of LDC in all instances, which shows the effectiveness of COI rule for fixing the subproblem. Between LDC-LF and LDC-POP, the computation time for LDC-LF is shorter than that of LDC-POP except for the large size case, while the performance

of LDC-POP is better than LDC-LF, which illustrates the neighborhood search in subproblems can improve solution performance.

The above computational results illustrate that the proposed model and solution approaches are effective, especially for medium and large size instances. Among all algorithms developed, the improved Lagrangian algorithm LDC-POP with COI rule and neighborhood search is the best in terms of solution quality. the gap of 6.29% for the real-world large size problem shows the solution is very close to optimal. Considering that in practice the planning is done only one or two times for the whole year, the running time of 30 minutes is quite acceptable.

It is notable to mention that no feasible solution can be found if the warehouse layout problem and capacitated lot-sizing problem for the industry case are solved separately due to the space limit of the warehouse. The proposed integrated strategy provides an effective way to coordinate the production planning and warehouse management.

9   Conclusions

In this paper, we propose an integrated strategy to combine a storage location assignment problem with a capacitated lot-sizing planning, which is motivated by a real-world case. The joint multi-item storage location assignment capacitated lot-sizing problem determines when and how much to produce for each product, and simultaneously determines where to place the products in the warehouse based on a dedicated storage policy. A mixed integer linear programming model is developed with the objective of minimizing the total costs of production, setup, storage, handling, and reserving space. It also ensures the availability of warehouse space in the planning horizon. The model and solution approaches are verified though different size of instances.

The model with the data from the real-world case is large scale and the problem complexity is beyond the capacity of the current optimization solvers. To solve the large-scale real-world case, three Lagrangian heuristic approaches are developed. Different from the conventional Lagrangian heuristics, the Lagrangian relax-and-fix heuristic approach presents a new way to construct a feasible solution by using COI rule to fix the subproblem one by one. We test the proposed approach with small, middle and large-scale instances, where the largest one corresponds   the real-world case. Computational results show that the performance of the Lagrangian relax-and- fix heuristics with COI rule outperforms that of the conventional Lagrangian relaxation method. Further, an improved variant of the novel Lagrangian relax-and-fix heuristics is developed by introducing a neighborhood search method, and the computation results show that the heuristics can obtain a near optimal solution with a reasonable running time.

It would be worthwhile to investigate the way to reduce the convergence time of the proposed method or

develop other solution approaches for the problems. The current model is developed under the situation where the demands are given and deterministic. An extension to this research is to take into account uncertain demands. With uncertain demands, a stochastic model and new solution approaches can be studied.

Acknowledgments

References

Absi, N., and Kedad-Sidhoum, S. (2008) The multi-item capacitated lot-sizing problem with setup times and shortage costs. *European Journal of Operational Research*, **185**, 1351-1374.

Alvim, A. C. F., Taillard, E. D. (2009) Popmusic for the point feature label placement problem. *European Journal of Operational Research*, **192**, 396-413.

Bitran, G.R., and Yanasse, H.H. (1982) Computational complexity of the capacitated lot size problem. *Operations Research*, **28**, 1174-1186.

Buschkühl, L., Sahling, F., Helber, S., and Tempelmeier, H. (2010) Dynamic capacitated lot-sizing problems: a classification and review of solution approaches. *OR Spectrum*, **32**, 231-261.

Çelk, M., and Süral, H. (2014) Order picking under random and turnover-based storage policies in fishbone aisle warehouses. *IIE transactions*, **46**(3), 283 -300.

Chen, H. (2015) Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems. *Omega,* 56, 25-36.

Chu, F., and Chu, C. (2008) Single-item dynamic lot-sizing models with bounded inventory and outsourcing. *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, **38**, 70-77

De Koster, R., Le-Duc, T., and Roodbergen, K.J. (2007) Design and control of warehouse order picking: a literature review. *European Journal of Operational Research*, **182**, 481-501.

Florian, M., and Klein, M. (1971) Deterministic Production Planning with Concave Costs and Capacity Constraints. *Management Science*, **18**, 12-20.

Francis, R. L. (1967) On Some Problems of Rectangular Warehouse Design and Layout. *The Journal of Industrial Engineering*, **18**, 595-604.

Francis, R.L., McGinnis, L.F., and White, J.A (1992) *Facility Layout and Location: An analytical Approach*, Prentice-Hall Inc., Englewood Cliffs, NJ

Gu, J., Goetschalckx, M., and McGinnis L.F. (2007) Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, **177**, 1-21.

Gu, J., Goetschalckx, M., and McGinnis, L.F. (2010) Research on warehouse design and performance evaluation: a comprehensive review. *European Journal of Operational Research*, **203**(3), 539-549.

Guo, X., Yu, Y., De Koster, R.B.M. (2016) Impact of required storage space on storage policy performance in a unit-load warehouse. *International Journal of Production Research*, **54**, 2405-2418

Gutierrez, J., Sedeno-Noda, A., Colebrook, M., and Sicilia, J. (2002) A new characterization for the dynamic lot size problem with bounded inventory. *Computers & Operation Research*, **30**, 383-395.

Harmatuck, D.J. (1976) A comparison of two approaches to stock location. *Logistics and Transportation Review*, **12**, 282–285.

Hassini, E. (2008) Storage space allocation to maximize inter-replenishment times. *Computers and Operations Research*, **35**, 2162-2174.

Heskett, J. L. (1963) Cube-per-order index-a key to warehouse stock location. *Transportation and Distribution Management*, **3**, 27-31.

Hodgson, T.J., and Lowe, T.J. (1982) Production Lot-sizing with Material-Handling Cost Consideration. *AIIE Transaction*, **14**, 44-51.

Hwang, H., and Kang, J. (2016) Two-phase algorithm for the lot-sizing problem with backlogging for stepwise transportation cost without speculative motives. *Omega*, **59**, 238-250.

Jans, R., and Degraeve, Z. (2008) Modeling industrial lot-sizing problems: a review. *International Journal of Production Research*, **46**, 1619-1643.

Karimi, B., Fatemi Ghomi, S.M.T., J.M. Wilson, and J.M. (2003) The capacitated lot-sizing problem: a review of models and algorithms. *Omega*, **31**, 365-378.

Kulturel, S., Ozdemirel, N.E., Sepil, C., and Bozkurt, Z. (1999) Experimental investigation of shared storage assignment policies in automated storage/retrieval systems. *IIE Transactions*, **31**, 739-749.

Larson. T. N., March. H., and Kusiak. A. (1997) A heuristic approach to warehouse layout with class-based storage. *IIE Transactions*, **29**, 337-348.

Liu, X., and Tu, Y. (2008) Production planning with limited inventory capacity and allowed stockout. *Int. J. Production Economics*, **111**, 180-191.

Love, S.f. (1972) A facilities in series inventory model with nested schedules. *Management Science*, **18**, 327-338.

Malmborg, C.J., and Altassan, K.M. (1988) Analysis of storage assignment policies in less than unit load warehousing systems. *International Journal of Production Research*, **36**, 3459–3475.

Malmborg, C.J., and Deutsch, S.J. (1988) A stock location model for dual address order picking systems. *IIE Transactions*, **20**, 44-52.

Malmborg, C.J., and Krishnakumar, B. (1989) Optimal storage assignment in multiaddress warehousing systems. *IEEE Transactions on Systems, Man and Cybernetics*, **19**, 197-204.

Muppani, V.R., and Adil, G. K. (2007) A branch and bound algorithm for class based storage location assignment. *European Journal of Operational Research*, **189**, 492-507.

Muppani, V.R., and Adil, G.K. (2008) Efficient formation of storage classes for warehouse storage location assignment: A simulated annealing approach. *Omega*, **36**, 609-618.

Nascimento, M.C.V., Resende, M.G.C., and Toledo, F.M.B. (2010) Grasp heuristic wit path-relinking for the multi-plant capaciated lot-sizing problem. *Eur. J. Oper. Res.*, **200**, 1-17.

Nishi, T. and Konishi, M. (2010) An optimization model and its effective beam search heuristics for floor-storage warehousing systems. *Int. J. Prod. Res.*, **48**, 1947-1966.

Nishi, T., Konishi, M., and Ago, M. (2007) A distributed decision making system for integrated optimization of production scheduling and distribution for aluminum production line, *Comput. Chem. Eng.,* 31, 1205-1221.

Nishi, T., Shinozaki, R., and Konishi, M. (2008) An augmented Lagrangian approach for distributed supply chain planning for multiple companies. *IEEE Transactions on Automation Science and Engineering*, **5**(2), 259-274.

Ostertag, A., Doerner, K.F., Hartl, R.F., Taillard, E.D., and Waelti, P. (2009) Popmusic for a real-world large-scale vehicle routing problem with time windows. *Journal of the Operational Research Society*, 60, 934-943.

Ohga, K., Nishi, T., Zhang, G., and Turner, S. (2013) Lagrangian relax and fix heuristics for integrated production planning and warehouse layout problem. *Proceedings of the 2013 IEEE International Conference on Industrial Engineering and Engineering Management*, **464.**

Page, E., and Paul, R.J. (1976) Multi-product inventory situations with one restriction. *Operational Research Quarterly*, **27**, 815-834.

Pan, J.C.-H., Wu, M.-H., and Chang, W.-L. (2014) A travel time estimation model for a high-level picker-to-part system with class-based storage policies. *European Journal of Operational Research*, **237**, 1054-1066.

Pan, J.C.-H., Shih, P -H., and Wu, M.-H. (2015) Order batching in a pick-and-pass warehousing system with group genetic algorithm. *Omega,* **57**, 238-248.

Petersen, C.G., and Aase, G. (2004) A comparison of picking, storage, and routing policies in manual order picking. *International Journal of Production Economics*, **92**, 11-19

Pochet, Y., and Wolsey, L.A. (1991) Solving multi-item lot-sizing problems using strong cutting planes. *Management Science*, **37**, 53-67.

Pohl, L.M., Meller, R.D. and Gue, K.R. (2011) Turnover-based storage in new unit-load warehouse designs. *IIE Transactions*, **43**(10), 703–720.

Robinson, P., Narayanan, A., and Sahin F., (2009) Coordinated deterministic dynamic demand lot-sizing problem: A review of models and algorithms, *Omega* **37**, 3 -15.

Taillard, E.D., and Voß, S. (2002) Popmusic: partial optimization metaheuristic under special intensification conditions. Esseys and Surveys in Metaheuristics, Operations Research/Computer Science Interface Series, **15**, 613-629.

Transchel, S., and Minner, S. (2009) Dynamic pricing and replenishment in the warehouse scheduling problem - A common cycle approach. *International Journal of Production Economics*, **118**, 331-338.

Wilson, H.G. (1977) Order quantity, production popularity, and the location of stock in a warehouse. *AIIE Transaction*, **9**, 230-237.

Zeng, A., Mahan, M., and Fluet, N. (2002) Designing an efficient warehouse layout to facilitate the order-filling process: An industrial distributor's experience. *Production and Inventory Management Journal*, **43**, 83-88.

Zhang G., Turner, S., and Nishi, T. (2012) An Integrated Strategy for a Production Planning - Layout Problem, Proceeding of Japan Scheduling Symposium, 183-188.

Zhang, G., (2010) The multi-product newsboy problem with supplier quantity discounts and a budget constraint, *European Journal of Operational Research*, **206**, 350-360

Zhang, G., and Lai K.K. (2010) Tabu search approach for multi-level warehouse layout problem with adjacent constraints, *Engineering Optimization*, **42**, 775-790.

Highlights:

- *A new integrated strategy that combines storage location assignment with capacitated lot-sizing problem is presented.*

- *The problem is motivated by a real-world case, and the data for testing are from the case.*

- *A mixed integer programming model has been developed for the joint problem; furthermore the model is dynamic.*

- *A novel Lagrangian relax- and-fix heuristic approach and its variants are proposed to solve the large scale problem.*

- *The real world industrial production warehouse problem is presented and solved by the proposed methods.*