European Journal of Operational Research 303 (2022) 581–592

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Integrated bin packing and lot-sizing problem considering the configuration-dependent bin packing process

Xinye Hao[a], Li Zheng[a], Na Li[b], Canrong Zhang[c,*]

[a] Department of Industrial Engineering, Tsinghua University, Beijing 100084, China
[b] Department of Industrial Engineering & Management, Shanghai Jiao Tong University, Shanghai 200240, China
[c] Logistics Engineering and Simulation Laboratory, Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China

ARTICLE INFO

Article history:
Received 16 September 2020
Accepted 8 March 2022
Available online 12 March 2022

Keywords:
Combinatorial optimization
Configuration-dependent bin packing process
Multi-level lot-sizing
Column generation
Bi-level dynamic programming

ABSTRACT

This paper addresses a practical production problem in an aviation manufacturing factory that focuses on the production of composite aeronautic products, where packing and assembling are two major operations conducted in the production process. We extract a novel integer programming model that integrates bin packing with the multi-level lot-sizing problem, which is characterized by re-configurable bins and configuration-dependent processing time. To obtain tighter lower bounds, we decompose the original model into a master problem and several sub-problems based on the Dantzig–Wolfe framework. To accelerate the column generation process, we develop a bi-level dynamic programming algorithm to solve the sub-problems exactly. A hybrid algorithm that combines the column generation and the fix-and-optimize heuristic is proposed to generate high-quality upper bounds. We first evaluate the quality of the lower bounds obtained from the linearly relaxed restricted master problem and then demonstrate the competitiveness of the proposed heuristic algorithm in terms of the solution quality and computational efficiency by comparing it with CPLEX, a commercial solver. In addition, we conduct sensitivity analysis of the bi-level dynamic programming to investigate the impact of the processing time required by the configuration-based bins. Moreover, a case study was conducted, which demonstrated the effect on cost reduction by shortening the length of unit time period in the planning horizon.

1. Introduction

The *bin packing problem (BPP)* and *lot-sizing problem (LSP)* are often encountered in the manufacturing industry, and most studies solved them separately as each problem is difficult to solve. In the BPP, a set of items must be packed into the minimum number of bins while ensuring that the total weight/length packed in any bins does not exceed the capacity (Delorme, Iori, & Martello, 2016). In the LSP, we must decide what and how much to produce over the planning horizon and particularly focus on the trade-off among different costs such as setup and holding costs (Melega, de Araujo, & Jans, 2018). As items packed into bins must undergo related operations which often involve the lot-sizing decisions, the integrated *bin packing and lot-sizing problem (BP-LSP)* has been gaining increasing interest in recent years (Melega et al., 2018), as the integration can generate better solutions in a broader spectrum than the sequential method.

In this paper, based on a practical production project on which we collaborate with an aviation manufacturing factory in China, we integrate the BPP and LSP with the aim of reducing the system-wide costs. More specifically, we focus particularly on the first stage, namely, curing, as it is a bottleneck in which items are required to be packed into different sizes of autoclaves, i.e. bins. On one hand, the autoclaves in the curing stage are re-configurable. Only when an autoclave is set up for a configuration can the items dedicated to this configuration be processed in it. On the other hand, the processing time of each autoclave is configuration-dependent. When an autoclave is set up for a particular configuration, the curing status must last for some time corresponding to this configuration. Therefore, the curing stage is actually a configuration-dependent bin packing process, differentiating it from previous research reported in the literature. In addition, we consider the subsequent assembling stage where the cured items are assembled into final products. For the two-stage production system, if we pack too many items into autoclaves in the first stage, the production costs will be reduced, whereas the holding costs of the cured items waiting for assembling in

* Corresponding author at: Logistics Engineering and Simulation Laboratory, Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China.

E-mail addresses: haoxinye_ie@163.com (X. Hao), lzheng@tsinghua.edu.cn (L. Zheng), na-li03@sjtu.edu.cn (N. Li), crzhang@sz.tsinghua.edu.cn (C. Zhang).

https://doi.org/10.1016/j.ejor.2022.03.012
0377-2217/

the second stage may increase, which indicates that bin packing decisions are coupled with lot-sizing decisions. In other words, the bin packing decision in our problem is assembly-oriented or internal demand-oriented, making the items that need to be packed unknown at the very beginning, while the traditional BPP solves a given set of items that must be packed.

We extract a novel *integrated model (IM)*, which is also called the “Kantorovich-based formulation” (De Carvalho, 2002) owing to the method used to formulate the bin packing decision. It integrates the BPP with a multi-level LSP while considering the configuration-dependent bin packing process. To tackle the computational difficulty, we decompose the model by machines into a *master problem (MP)* and several sub-problems based on the Dantzig–Wolfe framework and implement the *column generation (CG)* algorithm to solve the *linearly relaxed master problem (LMP)* to generate high-quality lower bounds of the IM. To accelerate the CG process, we prove that the sub-problem corresponding to the configuration-dependent bin packing process can be reformulated into a routing problem after solving a series of unbounded knapsack problems (UKPs), which enables us to develop a *bi-level dynamic programming (BLDP)* algorithm to solve the sub-problems exactly. More specifically, the first level of BLDP involves dynamic programming to solve a series of UKPs, and the second level solves a routing problem using another novel dynamic programming algorithm. To generate upper bounds, we combine the columns selected from the *linearly relaxed restricted master problem (LRMP)* with a *fix-and-optimize (FO)* heuristic, which is called the “*combined column generation and fix-and-optimize heuristic (CGFO)*”. Extensive experiments were conducted and the numerical results demonstrated the competitiveness of CGFO in terms of the solution quality and computational efficiency when compared with CPLEX 20.1.0, a commercial solver. A sensitivity analysis of BLDP demonstrated its stability. The tasks and contributions of our paper are summarized as follows:

- We establish a novel model, an IM, for the integrated bin packing and multi-level lot-sizing problem that is characterized by the configuration-dependent bin packing process. To the best of our knowledge, there is no reported study in the literature on a similar problem.
- We decompose the IM by machines based on the Dantzig–Wolfe framework to generate tighter lower bounds than the *linearly relaxed integrated model (LIM)*. We develop a high-efficiency exact algorithm, BLDP, which involves two levels of dynamic programming to solve the sub-problem corresponding to the configuration-dependent bin packing process, to accelerate the CG process dramatically.
- Based on the columns selected from the LRMP and the idea of FO, we develop a high-efficiency heuristic called CGFO to generate high-quality upper bounds.
- We conducted a case study to test the effect of changing the *unit time period (UTP)* in the planning horizon from the initial 8 to 4 h.

The remainder of this paper is organized as follows. In Section 2, we review the most relevant literature related to our study. In Section 3, we propose the IM. The Dantzig–Wolfe decomposition model, BLDP, and CGFO are proposed in Section 4. Computational experiments and a case study are described in Section 5. Conclusions and future research are summarized in Section 6.

2. Literature review

The BP-LSP has been receiving increasing interest in recent years owing to its theoretical complexity and practical significance in the manufacturing industry. In addition to the BPP, researchers also focus on to the *cutting stock problem (CSP)*, a variant of the BPP, which involves the cutting of large objects into small items to satisfy demand with the minimum number of deployed objects (Melega et al., 2018), and the CSP integrated with the LSP (CS-LSP). In essence, the CSP is a generalization of the BPP (Delorme et al., 2016); thus, in the remainder of this paper, we do not distinguish the BP-LSP from the CS-LSP, and use “integrated model” or “integrated problem” to represent the related research. Arbib & Marinelli (2005) established an integrated model for a gear belt production that combined a cutting process in stage one and an assembling process in stage two. Gramani, França, & Arenales (2009) and Gramani, França, & Arenales (2011) developed an integrated model for the furniture industry and demonstrated that it provides a plan with lower total costs. Vanzela, Melega, Rangel, & de Araujo (2017) studied an integrated problem for furniture production and indicated that the integrated approach performs well in terms of both the total costs of raw materials and inventory costs of pieces. Wu, Akartunalı, Jans, & Liang (2017) studied an integrated problem similar to the one proposed by Gramani et al. (2009) and modelled it based on the shortest path formulation. All the aforementioned studies drew the same conclusion, i.e. solving the integrated problem can generate a better solution in a broader spectrum.

In the integrated problem, the product structure is also frequently considered. In the related research, small items are packed together, or large objects are cut into pieces, and then the semi-finished products undergo some processes such as assembling to produce the final products while respecting the bill-of-material (BOM)-related constraints. In addition to the multi-level characteristics, researchers give special interest to the capacity of the involved cutting/packing processes, simply because the production resources are limited. Different types of capacity constraints are considered in the literature. For example, Arbib & Marinelli (2005) considered the capacity of the cutting process by limiting the total amount of materials to be cut. Similarly, in the research conducted by Gramani et al. (2009) based on furniture production, the saw machines were capacitated as the total area of the materials that could be cut in each period was limited. In addition, the capacity is also considered in the form of working time included in the planning horizon. For example, in the model proposed by Gramani et al. (2011), the capacity was limited by the total time. Alem & Morabito (2013) imposed limited regular time for the cutting process and required to determine the amount of overtime to satisfy the unfulfilled demand. In addition, different from other papers, this study considered the problem under uncertain parameters and used a stochastic programming approach. Pierini & Poldi (2021) modelled the capacity of machines that produce large paper reels in the form of the total weight of paper. In these papers related to capacitated cutting/packing processes, some researchers considered the setup operation. For example, Melega, de Araujo, & Morabito (2020) considered the sequence-dependent setup times and setup costs in the cutting process; similar research was conducted by Oliveira, Fiorotto, Song, & Jones (2021) and Pierini & Poldi (2021). Other studies did not consider these, e.g. de Lara Andrade, de Araujo, Cherri, & Lemos (2021). In some of these papers considering the setup operation, the summation of setup time and processing time, which are positively related to the quantity of products produced, cannot exceed the total available time of each machine in each period (or in other forms of capacity).

Our research integrated the BPP with the multi-level capacitated LSP. Our problem is special owing to the curing stage, in which the items are required to be packed into different lengths of autoclaves, i.e. bins, and then undergo the curing process. In this stage, each autoclave is re-configurable, as its hardware and software components can be reset (Khezri, Benderbal, & Benyoucef, 2021; Koren et al., 1999), based on which it can be set up for various configurations; subsequently, the items corresponding to their particular configurations can be processed in it. In addition, the processing time of each autoclave is configuration-dependent. Therefore, we define the curing stage as "the configuration-dependent bin packing process", and permit the curing process to last for some time.

With respect to the solution methods, to solve the integrated problem, researchers have developed various solution methods. Because the integrated problem is NP-hard, the reported studies focused primarily on heuristic methods. Poltroniere, Poldi, Toledo, & Arenales (2008) provided two heuristics, the first being a Lagrangian relaxation based heuristic, and the second one solves the CSP and LSP sequentially. The performances of the two heuristics were compared in terms of upper bounds and the number of iterations, while the performance with respect to convergence was not mentioned. Gramani et al. (2009) proposed a heuristic method based on Lagrangian relaxation to solve the integrated problem. Other studies selected to solve the linearly relaxed version of the MP (also called the "pattern-based formulation" in papers such as De Carvalho (2002)) decomposed from the integrated problem based on the Dantzig–Wolfe framework, and then developed heuristics using the fractional solution. Leao, Furlan, & Toledo (2017) studied the integrated problem in the paper industry and decomposed the integrated model by machines. They provided a rounding heuristic and an adaptive large neighbourhood search heuristic, both of which were based on CG and performed well in terms of computational efficiency and solution quality. Wu et al. (2017) proposed a hybridized CG and progressive selection heuristic to solve the integrated problem. Some similar heuristics are discussed in Nonås & Thorstenson (2008), Poltroniere, Araujo, & Poldi (2016), and Oliveira et al. (2021). In this paper, we decompose the IM by machines and generate tight lower bounds using CG. To generate high-quality upper bounds, we develop an efficient heuristic algorithm, CGFO, based on the columns selected from LRMP and the FO heuristic.

3. Mathematical formulation

This section consists of two parts. First, the problem is described based on the practice in an aviation manufacturing factory, and then the IM is presented. Note that, in this section, we first present an "original integrated model (OIM)", and then propose a new type of constraint to simplify the OIM into a "simplified integrated model (SIM)". In subsequent sections of the paper, "IM" refers to the SIM.

3.1. Problem description

In this paper, we discuss a practical production problem in an aviation manufacturing factory, which focuses on the production of composite aeronautic products in a multi-variety and small-batch mode. The entire process includes two main stages as shown in Fig. 1. In the first stage, items placed on their dedicated trays are sent into the industrial autoclaves with different lengths to undergo the curing process. During the process, only the items sharing the same curing configuration, such as the same temperature and pressure, can be processed in the same autoclave. That is, the items dedicated to different curing configurations cannot be mixed. After the curing process, which lasts for a time corresponding to this particular configuration, the trays are removed from the autoclave, and the cured items on trays are unloaded. In the second stage, the cured items are assembled into a composite end item according to the BOM requirement. Thereafter, the composite end item is delivered to customers.

Because the curing stage is actually a bin packing process, with the lot-sizing decisions to be made for each item, the problem is essentially an integrated bin packing and lot-sizing problem, which is characterized by the configuration-dependent bin packing process. Before formulating the integrated problem, the following assumptions related to the problem are made.

1) All the input parameters are known in advance. In other words, this is a deterministic problem.
2) Because the bottleneck appears at the curing stage, the assembling stage is assumed to be uncapacitated.
3) The external demands of the end items are permitted to be backordered, and the non-end items, i.e., the cured items, are not externally needed.
4) The curing process frequently lasts for some time; therefore, the lead time must be considered. We consider the lead time of the curing process as an integer multiple of the UTP. More specifically, we calculate lead time using $\lceil \frac{curingTime}{UTP} \rceil$. For example, when the UTP is set to 8 h, a configuration with curing time equal to 12 h has a lead time of $\lceil \frac{12}{8} \rceil = 2$.
5) The trays are assumed to have the same width and only differ in length to account for the different sizes of items. The trays are placed horizontally in a single row along the autoclaves, each of which has a rectangular area; therefore, the number of trays that can be accommodated in each autoclave depends on the length of the rectangular area and the accumulated length of the assigned trays.
6) The number of trays for each size required is sufficient.

3.2. Mathematical model

The notations of the IM (OIM and SIM) are given as follows:
Sets and indices:

- T Set of periods in the planning horizon, indexed by t and t'.
- I Set of all items, indexed by i and i'.
- End_p Set of end items, indexed by j, $End_p \subset I$.
- U Set of curing configurations, indexed by u.
- M Set of autoclaves, indexed by m.

Input parameters:

- lt_i Production lead time of cured item i.
- b_{iu} 1, if item i is cured under configuration u; 0, otherwise.
- l_u Production lead time of curing configuration u. If $b_{iu} = b_{i',u} = 1$, then $lt_i = lt_{i'} = l_u$.
- r_{ij} Quantity of cured item i required to be assembled into a unit of its immediate parent (end) item j according to the BOM requirement.
- d_{jt} External demand for end item j in period t.
- v_i Length of the tray compatible with the cured item i.
- q_m Length of the autoclave m that can be used to accommodate different trays.
- pc_{um} Production cost of the autoclave m under curing configuration u.
- hc_i Holding cost of a unit of item i for a period.
- bc_j Backorder cost of a unit of end item j for a period.

Decision variables:

- S^+_{it} Inventory level of item i at the end of period t.
- S^-_{jt} Backorder level of end item j at the end of period t.
- X_{imt} Quantity of item i placed into autoclave m in period t.
- Y_{umt} 1, if autoclave m is set up for configuration u in period t; 0, otherwise. Note that when the autoclave is set up for a particular configuration u, the production status should last for the entire lead time l_u.
- Z_{umt} 1, if autoclave m is used under configuration u in period t; 0, otherwise.
- A_{jt} Quantity of item j assembled in period t.

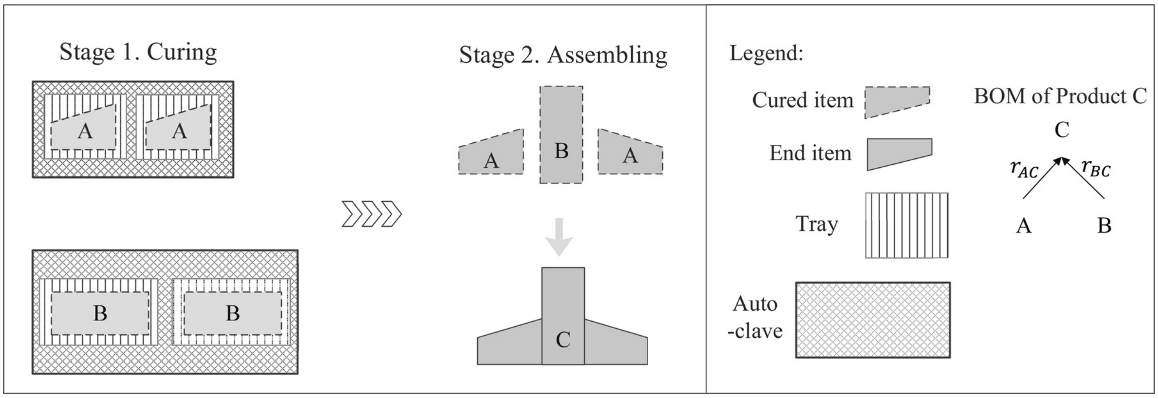

Fig. 1. Schematic production process of a composite item.

The OIM is expressed as follows.

$$[\textbf{OIM}] \quad \min \sum_{t\in T}\sum_{i\in I} hc_i \cdot S^+_{it} + \sum_{t\in T}\sum_{j\in End_p} bc_j \cdot S^-_{jt} + \sum_{t\in T}\sum_{m\in M}\sum_{u\in U} pc_{um}\cdot Y_{umt} \tag{1}$$

$$\textbf{s.t. } S^+_{j,t-1} - S^-_{j,t-1} + A_{jt} = d_{jt} + S^+_{jt} - S^-_{jt} \quad \forall j\in End_p; t\in T \tag{2}$$

$$S^+_{i,t-1} + \sum_{m\in M} X_{i,m,t-lt_i} = \sum_{j\in\left\{j:j\in End_p, r_{ij}>0\right\}} r_{ij}\cdot A_{jt} + S^+_{it} \quad \forall i\in I\setminus End_p; t\in T \tag{3}$$

$$\sum_{u\in U} Y_{umt} \le 1 \quad \forall m\in M; t\in T \tag{4}$$

$$\sum_{u\in U} Z_{umt} \le 1 \quad \forall m\in M; t\in T \tag{5}$$

$$\sum_{t'\in\{t':t\le t'\le t+l_u-1\}} Y_{u,m,t'} \le 1 \quad \forall u\in U; m\in M; t\in T \tag{6}$$

$$Y_{umt} \le Z_{u,m,t'} \quad \forall u\in U; m\in M; t\in T, t'\in\left\{t': t\le t'\le t+l_u-1\right\} \tag{7}$$

$$X_{imt} \le \left\lfloor \frac{q_m}{v_i}\right\rfloor \cdot \left(\sum_{u\in U} b_{iu}\cdot Y_{umt}\right) \quad \forall i\in I\setminus End_p; m\in M; t\in T \tag{8}$$

$$\left(\sum_{i\in I\setminus End_p} v_i\cdot X_{imt}\right) \le q_m \quad \forall m\in M; t\in T \tag{9}$$

$$S^+_{it}, S^-_{jt}, A_{jt}\in Z^+ \quad \forall i\in I; j\in End_p; t\in T \tag{10}$$

$$X_{imt}\in Z^+ \quad \forall i\in I\setminus End_p; m\in M; t\in T \tag{11}$$

$$Y_{umt}, Z_{umt}\in\{0,1\} \quad \forall u\in U; m\in M; t\in T \tag{12}$$

Objective function (1) minimizes the total costs over the planning horizon, including the production costs of autoclaves, holding costs, and backorder costs. Constraints (2) are the flow conservation constraints dedicated to the end items that are externally needed. Constraints (3) are the inventory balance equations dedicated to the cured items which only have dependent demands caused by the assembling process of their immediate parent (end) items. Constraints (4) indicate that an autoclave can select at most one curing configuration at the beginning of a time period. Similarly, an autoclave can be used under at most one configuration in each period, resulting in constraints (5). Constraints (6) indicate that for each curing configuration u, in any duration with length equal to l_u, the autoclave cannot be set up for the configuration more than once. Constraints (7) ensure that the autoclave is used under curing configuration u for a duration with length equal to l_u if it is set up for configuration u. Constraints (8) establish the connection between the curing configuration u and cured item i. In other words, only those items matching with the configuration of the autoclave can be cured in it. Constraints (9) are the spatial constraints, i.e. the accumulated length of trays placed into autoclave m cannot exceed its capacity q_m. Constraints (10) – (12) enforce the non-negativity and integrality requirements for decision variables.

3.3. Model simplification

In this part, we propose a new type of constraint which helps reduce the quantity of constraints in OIM. The details are summarized in the following proposition.

Proposition 1. *The following constraints (13) can replace constraints (4), (6), and (7) in the OIM.*

$$\sum_{t'\in\{t':\max\{t-l_u+1,1\}\le t'\le t\}} Y_{u,m,t'} = Z_{umt} \quad \forall u\in U; m\in M; t\in T \tag{13}$$

Proof. Constraints (13) indicate that, for each period t, whether an autoclave m is used under configuration u depends on whether the previous $l_u - 1$ periods and the present period have a setup for configuration u. We prove this proposition in three parts.

a) Summing constraints (13) over u, we obtain

$$\sum_{u\in U}\sum_{t'\in\{t':\max\{t-l_u+1,1\}\le t'\le t\}} Y_{u,m,t'} = \sum_{u\in U} Z_{umt} \quad \forall m\in M; t\in T$$

In line with constraints (5), we obtain

$$\sum_{u\in U} Y_{umt} \le \sum_{u\in U}\sum_{t'\in\{t':\max\{t-l_u+1,1\}\le t'\le t\}} Y_{u,m,t'} \le 1 \quad \forall m\in M; t\in T$$

Thus, constraints (4) are guaranteed.

b) As $Z_{umt}\le 1$, such that $\forall u\in U, m\in M, t\in T$, we obtain

$$\sum_{t'\in\{t':\max\{t-l_u+1,1\}\le t'\le t\}} Y_{u,m,t'} \le 1 \quad \forall u\in U; m\in M; t\in T$$

which are actually the same as constraints (6).

c) Constraints (13) can be divided into the following two parts:

$$\sum_{t'\in\{t':\max\{t-l_u+1,1\}\le t'\le t\}} Y_{u,m,t'} \le Z_{umt} \quad \forall u\in U;\, m\in M;\, t\in T \tag{14}$$

$$\sum_{t'\in\{t':\max\{t-l_u+1,1\}\le t'\le t\}} Y_{u,m,t'} \ge Z_{umt} \quad \forall u\in U;\, m\in M;\, t\in T \tag{15}$$

In line with constraints (14), we obtain

$$\max_{t'\in\{t':\max\{t-l_u+1,1\}\le t'\le t\}} Y_{u,m,t'} \le Z_{umt} \quad \forall u\in U;\, m\in M;\, t\in T$$

Thus, constraints (7) are guaranteed. □

Based on Proposition 1, we reformulate the OIM into the SIM, which has fewer constraints than OIM, as follows:

$$[\textbf{SIM}] \quad \min \quad \text{Objective} \quad (1)$$
$$\text{s.t.} \quad \text{Constraints} \quad (2), (3), (5), (8)-(13)$$

Note that, in the remainder of the paper, the IM refers to the SIM.

Owing to the poor quality of the lower bound obtained from the LIM, we instead decompose the IM based on the Dantzig–Wolfe framework in the following section to generate tighter lower bounds. Moreover, we propose CGFO, an efficient heuristic algorithm, to generate high-quality upper bounds.

4. Solution approach

In this section, we decompose the IM by machines (autoclaves) based on the Dantzig–Wolfe framework. We use CG to solve the LMP, which provides tighter lower bounds for the IM than the LIM. To initiate the computing, we use CPLEX to generate initial columns. To accelerate the convergence process of CG, we analyse the structure of the sub-problems and propose BLDP to solve it exactly. After sufficient columns are generated and added into LRMP, the FO heuristic is used to generate high-quality upper bounds, i.e. feasible solutions to the IM.

4.1. Dantzig–Wolfe decomposition

In this section, we decompose the IM by machines into one MP and $|M|$ sub-problems based on the Dantzig–Wolfe framework. First, we introduce additional notations in which the input parameters are denoted by lower-case letters, except for the sets, while the decision variables are denoted by upper-case letters.

Sets and indices:

Θ_m Set of feasible solutions (schedules) of autoclave m, indexed by n_m.

Input parameters:

$x_{i,n_m,t}$ Quantity of item i placed into autoclave m in period t under schedule n_m.

$y_{u,n_m,t}$ 1, if autoclave m is set up for configuration u in period t under schedule n_m; 0, otherwise.

Decision variables:

Q_{n_m} 1, if the schedule n_m of autoclave m is selected; 0, otherwise.

4.1.1. Master problem

Based on the defined notation shown above, we formulate the master problem of the IM as follows.

$$[\textbf{MP}] \quad \min \sum_{t\in T}\sum_{i\in I} hc_i \cdot S^+_{it} + \sum_{t\in T}\sum_{j\in End_p} bc_j \cdot S^-_{jt}$$
$$+\sum_{t\in T}\sum_{m\in M}\sum_{n_m\in\Theta_m}\sum_{u\in U} pc_{um}\cdot y_{u,n_m,t}\cdot Q_{n_m} \tag{16}$$

$$\textbf{s.t.}\ S^+_{j,t-1} - S^-_{j,t-1} + A_{jt} = d_{jt} + S^+_{jt} - S^-_{jt} \quad \forall j\in End_p;\, t\in T \tag{17}$$

$$S^+_{i,t-1} + \sum_{m\in M}\sum_{n_m\in\Theta_m} x_{i,n_m,t-lt_i}\cdot Q_{n_m} = \sum_{j\in\{j:j\in End_p,r_{ij}>0\}} r_{ij}\cdot A_{jt} + S^+_{it} \quad \forall i\in I\backslash End_p;\, t\in T \tag{18}$$

$$\sum_{n_m\in\Theta_m} Q_{n_m} \le 1 \quad \forall m\in M \tag{19}$$

$$S^+_{it}, S^-_{jt}, A_{jt}\in Z^+ \quad \forall i\in I;\, j\in End_p;\, t\in T \tag{20}$$

$$Q_{n_m}\in\{0,1\} \quad \forall m\in M;\, n_m\in\Theta_m \tag{21}$$

Objective function (16) minimizes the total costs including the holding and backorder costs over the planning horizon, as well as the production costs of feasible plans for each autoclave. Constraints (17) and (18) are the flow conservation constraints which are equivalent to constraints (2) and (3), respectively, where the previous decision variable X_{imt} is replaced by $x_{i,n_m,t}$. Constraints (19) ensure that, for each autoclave, at most one feasible solution can be selected. Constraints (20) – (21) are the domain confinement of decision variables. Additionally, we introduce two sets of dual variables, α and β, respectively corresponding to constraints (18) and (19) in the LRMP. α and β are used in the formulation of the sub-problems presented in the next subsection.

4.1.2. Sub-problems

In this part, we discuss the sub-problem corresponding to each autoclave m. We omit the index m here as they only differ in the dual price β_m and autoclave capacity q_m, both of which are constants. The formulation of sub-problem is expressed as follows:

$$[\textbf{SP}] \quad \min \sum_{t\in T}\sum_{u\in U} pc_u\cdot Y_{ut} - \sum_{t\in T}\sum_{i\in I\backslash End_p} \alpha_{it}\cdot X_{i,t-lt_i} - \beta \tag{22}$$

$$\textbf{s.t.}\ \sum_{u\in U} Z_{ut}\le 1 \quad \forall t\in T \tag{23}$$

$$\sum_{t'\in\{t':\max\{t-l_u+1,1\}\le t'\le t\}} Y_{u,t'} = Z_{ut} \quad \forall u\in U;\, t\in T \tag{24}$$

$$X_{it} \le \left\lfloor \frac{q}{v_i}\right\rfloor \cdot \left(\sum_{u\in U} b_{iu}\cdot Y_{ut}\right) \quad \forall i\in I\backslash End_p;\, t\in T \tag{25}$$

$$\left(\sum_{i\in I\backslash End_p} v_i\cdot X_{it}\right) \le q \quad \forall t\in T \tag{26}$$

$$X_{it}\in Z^+ \quad \forall i\in I\backslash End_p;\, t\in T \tag{27}$$

$$Y_{ut}, Z_{ut}\in\{0,1\} \quad \forall u\in U;\, t\in T \tag{28}$$

Objective function (22), whose optimal value is the reduced cost, consists of the production costs and objectives related to the dual variables from the LRMP. Constraints (23) – (28) are equivalent to constraints (5), (13), (8), (9), (11), and (12) in the IM except that index m is omitted from the subscripts.

4.2. Analysis of the sub-problem

In this part, we analyse the structure of the sub-problem and propose BLDP to solve it exactly. Note that, in the sub-problem, we need to settle the setup plan, with each setup decision corresponding to a bin packing plan. First, we propose the following lemma.

Lemma 1. *In the optimal solution of a sub-problem, if $Y_{ut} = 1$ for a certain configuration $u \in U$ in period $t \in T$, then in this particular period, the corresponding UKP must be solved to optimality. (Let $\delta_{it} = \alpha_{i,t+lt_i}$, and then omit the indices m and t in decision variable X and δ.)*

$$[\mathbf{UKP}] \quad \min\ pc_u - \sum_{i\in\{i:i\in I\setminus End_p, b_{iu}=1\}} \delta_i \cdot X_i \tag{29}$$

$$\mathbf{s.t.} \quad \sum_{i\in\{i:i\in I\setminus End_p, b_{iu}=1\}} v_i \cdot X_i \le q \tag{30}$$

$$X_i \in Z^+ \quad \forall i \in \{i : i \in I \setminus End_p, b_{iu} = 1\} \tag{31}$$

Proof. When $Y_{ut} = 1$, we must decide what items and how many of them that match this configuration should be processed, with the aim of minimizing function (29). As pc_u is actually a constant with no impact on the calculating process, we can omit it, and the remaining part is actually an UKP. Subsequently, we prove this lemma by contradiction. If $Y_{ut} = 1$ and its corresponding UKP is not solved to optimality, there must exist other better solutions resulting in a lower objective value of the sub-problem. □

Based on Lemma 1, we conclude that in a period t when an autoclave is set up, if we solve U UKPs to generate all possible bin packing plans, then there must be one plan selected in the optimal solution of the sub-problem. We denote the optimal value of the UKP under each configuration u in each period t as η_{ut}, which is used to reformulate the sub-problem. The *reformulated sub-problem (RSP)* is shown as follows:

$$[\mathbf{RSP}] \quad \min \sum_{t\in T}\sum_{u\in U} \eta_{ut} \cdot Y_{ut} - \beta \tag{32}$$

$$\mathbf{s.t.} \sum_{u\in U} Z_{ut} \le 1 \quad \forall t \in T \tag{33}$$

$$\sum_{t'\in\{t':\max\{t-l_u+1,1\}\le t'\le t\}} Y_{u,t'} = Z_{ut} \quad \forall u \in U; t \in T \tag{34}$$

$$Y_{ut}, Z_{ut} \in \{0, 1\} \quad \forall u \in U; t \in T \tag{35}$$

After denoting each period as a node, the multi-period curing process can be represented as a path from the period when a setup decision is made to the period when the corresponding curing process is terminated, and then we can consider the RSP as a routing problem. We solve the RSP using dynamic programming. The basic idea of BLDP is shown in Fig. 2 and is explained as follows. First, for each pair of u and t, we calculate η_{ut}. If $\eta_{ut} < 0$, we input it into the routing graph. In some period, if more than one configuration whose $\eta < 0$ and l_u are the same, that is, the starting and ending points are the same, then we select the configuration with the minimum η to construct the path. If a path is constructed, we label it as $\rho_{ut} = 1$. After constructing the routing graph, we invoke a dynamic programming algorithm to solve the RSP, i.e. the routing problem. The computational complexity of BLDP is summarized in the following proposition.

Proposition 2. *The sub-problem can be reformulated as a routing problem after solving $|T| \cdot |U|$ UKPs and then solved in pseudo-polynomial time, with the computational complexity equal to $\mathcal{O}(|T| \cdot |U| \cdot q \cdot |I \setminus End_p|)$.*

Proof. We prove this proposition in two parts, respectively corresponding to the two levels of BLDP.

In Level 1 of BLDP, for each pair of u and t, we calculate the corresponding UKP and obtain η_{ut}. We select to solve the UKP using the *efficient dynamic programming for the unbounded knapsack problem (EDUK)* proposed in Andonov, Poirriez, & Rajopadhye (2000) with the computational complexity of $\mathcal{O}(q \cdot |I \setminus End_p|)$. The recursive function is shown in (36). Combined with the procedure conducted for each period and setting, the computational complexity of Level 1 is $\mathcal{O}(|T| \cdot |U| \cdot q \cdot |I \setminus End_p|)$.

$$\eta_{ut}(q) = \min_{i\in\{i:i\in I\setminus End_p, b_{iu}=1\}} \{\eta_{ut}(q), \eta_{ut}(q - v_i) - \delta_{it}\} + pc_u \tag{36}$$

In Level 2, we reformulate the sub-problem after using η_{ut} as input parameters, and then solve the corresponding RSP using dynamic programming. We represent the optimal objective value from period 1 to t as $\varphi(t)$, which is initially set to $-\beta$. We conduct the recursion from period 1 to $|T|$ with the computational complexity of $\mathcal{O}(|T| \cdot |U|)$ based on the recursive function (37):

$$\varphi(t) = \min_{u\in\{u:u\in U, \rho_{ut}=1\}} \left\{\varphi(t), \varphi(t - l_u) + \eta_{u,t-l_u}(q)\right\} \tag{37}$$

Based on the above analysis, the computational complexity of BLDP is $\mathcal{O}(|T| \cdot |U| \cdot q \cdot |I \setminus End_p| + |T| \cdot |U|)$, i.e. $\mathcal{O}(|T| \cdot |U| \cdot q \cdot |I \setminus End_p|)$. □

The framework of BLDP is shown in Algorithm 1 .

Algorithm 1: Bi-level Dynamic Programming.

Input: $\boldsymbol{\alpha}, \boldsymbol{\beta} \leftarrow$ dual variables
•Step 1: Index construction
$\delta_i = \alpha_{i,t+lt_i}$
•Step 2: Calculate UKPs
For each pair of u and t, considering the area of autoclave, i.e. q, solve the UKP formulated as (29) – (31) to obtain η_{ut} based on the recursive function (36).
• Step 3: Construct the routing graph
Take η_{ut}, which is negative, as the weight of the corresponding path. If more than one $\eta_{ut} < 0$ exists in the same path, select the minimum one. In addition, if no $\eta_{ut} < 0$ exists between two adjacent periods, construct a path with $\eta = 0$ to ensure that the RSP can be solved.
• Step 4: Solve the RSP
Conduct a forward recursion based on the recursive function (37).
• Step 5: Results recovery
Backtrack along the path recorded to recover the optimal solution of the sub-problem.

4.3. Initial columns generation & bounding techniques

The initial columns for the LRMP can be generated in several ways, such as implementing heuristic algorithms, or applying a commercial solver directly. Similar to Wu, Xiao, Zhang, He, & Liang (2018), we generate initial columns using CPLEX. After the initial columns are added into the LRMP, CG begins to iterate.

A valid lower bound is available to the IM throughout the CG process (Degraeve & Jans, 2007; Xie, Wu, & Zhang, 2019). More specifically, let us denote the optimal objective value of LRMP at iteration ϵ as θ^{ϵ}_{LRMP}, and let θ^{ϵ}_{m} be the reduced cost of the column generated at iteration ϵ for autoclave m; thus, a valid lower bound for IM is $LB^{\epsilon} = \theta^{\epsilon}_{LRMP} + \sum_{m\in M} \min(\theta^{\epsilon}_{m}, 0)$. After the termination of the CG process, i.e. when sufficient columns generated and added into the LRMP, we generate upper bounds, i.e. feasible solutions to the IM based on the heuristic CGFO. The basic idea of the CGFO is

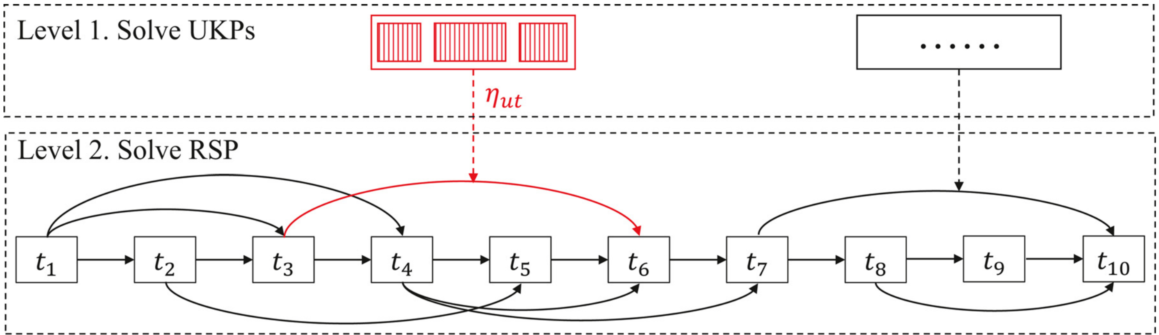

Fig. 2. Framework of bi-level dynamic programming.

to select columns from the LRMP and let $\rho_{ut}^m = 1$ if $y_{u,n_m,t} = 1$ exists in the selected columns. Subsequently, we establish and solve an IM whose decision variables Y are defined by (38).

$$Y_{umt} \in \{0, 1\} \quad \forall u \in U; m \in M; t \in T; \rho_{ut}^m = 1 \tag{38}$$

In other words, we establish an IM whose decision variables $Y_{umt} (\forall m \in M)$ are initiated according to $(u, t) \in \left\{(u, t) | u \in U, t \in T, \rho_{ut}^m = 1\right\}$. Similar to the FO heuristic, it reduces the solution space based on information given by the columns selected from the LRMP. Based on this idea, we propose the following two variants of the CGFO according to how many columns are selected from the LRMP, i.e. how many decision variables Y_{umt} are initiated:

- CGFO-I: Selecting columns whose $Q_{n_m} > 0$ in the LRMP.
- CGFO-II: Selecting columns whose $Q_{n_m} \geq 0.10$ in the LRMP.

Theoretically, CGFO-I selects more or an equal amount of columns compared with CGFO-II, resulting in a setup set not smaller than that constructed using CGFO-II. As a result, CGFO-I has a solution space not smaller than that of CGFO-II; therefore, its optimal solution will not be worse than that of CGFO-II. However, CGFO-I may need more computation time owing to the larger solution space. CGFO-II may generate high-quality solutions in a relatively short time. This trade-off was verified through the computational experiments.

5. Computational experiments

In this section, we first provide some rules based on which the experiment data were generated. Subsequently, we evaluate the quality of lower bounds provided by the LRMP and the computational efficiency of CG. In the third part, computational experiments related to the CGFO are described and the numerical results demonstrate the strong competitiveness of the CGFO in terms of computational efficiency and solution quality when compared with CPLEX. Sensitivity analysis of BLDP on the processing time is conducted. Moreover, in the final part, we describe a case study based on the real practice of the collaborative factory.

5.1. Data generation

Based on the practice in the factory we collaborate with, we generated two sets of instances, i.e. Set-A and Set-B, each of which consists of 20 instances. The parameters of instances were divided into four categories: parameters related to instance scale, parameters related to process, parameters related to costs, and parameters related to external demands. The parameters related to instance scale are summarized in Table 1.

The other parameters were set as follows.

Parameters related to process:

Table 1
Parameters related to instance scale.

Instance Set	$\lvert T\rvert$	$\lvert U\rvert$	$\lvert End_p\rvert$	$\lvert I \setminus End_p\rvert$	$\lvert M\rvert$
A	15	5	5	25	5
B	25	10	7	35	7

r_{ij} Based on the actual BOM structures, they were randomly selected from the set $\{0, 2, 4\}$.

q_m Based on the lengths of autoclaves in the factory, the length of each autoclave was generated using a uniform distribution $U(5, 10)$.

v_i Based on the lengths of trays collected from the factory, we generated v_i by $U(1, 2)$ with $Prob(0.8)$ and $U(0.5, 1)$ with $Prob(0.2)$.

lt_i, l_u Lead time of a cured item or a configuration was selected from the set $\{1, 2, 3\}$. Note that the lead time of a cured item should be consistent with l_u with which it matches.

Parameters related to costs:

pc_{um} Production cost in one period ranges in $c^* = U(100, 150)$. As the production cost for the autoclave is related to its area, it is reasonable that $pc_{um} = \frac{c^* \cdot l_u \cdot q_m}{\min_{m \in M} q_m}$.

hc_i For each cured item i, $hc_i = \frac{c^* \cdot lt_i}{10 \cdot |T|}$. For each end item j, $hc_j = 1.2 \cdot \sum_{i \in \{i: I \in I \setminus End_p, r_{ij} > 0\}} r_{ij} \cdot hc_i$.

bc_j For each end item j, the backorder cost is set to $bc_j = 10 \cdot hc_j$.

Parameters related to external demands:

d_{jt} For each end item j, external demands are generated in set $\{1, 2\}$. We generate external demands twice for each end item as we consider a dynamic lot-sizing problem where the external demands for each end item may arrive in more than one period.

In our computational experiments, all the algorithms were implemented in Java and the computations were conducted on a laptop with an Intel Core i7-8750H 2.20 GHz processor and 16GB RAM. All the linear and integer programs were solved using CPLEX 20.1.0.

5.2. Performance analysis of CG

In this part, we describe the performance of CG, including the quality of lower bounds provided by the LRMP and the computational efficiency of CG. We first used CPLEX to generate initial columns. Whenever any one of the following two criteria was satisfied, the computation process of CPLEX terminated: (i) a time limit of 90 and 180 s for instances in Set-A and Set-B, respectively; (ii) a relative gap of 20.00%. After the initial columns were added

Table 2
Performance of CG.

Instance No.	Set-A				Set-B			
	LB^{LIM}	LB^{LRMP}	CPU Time (s)	$Gap^{LRMP-LIM}$	LB^{LIM}	LB^{LRMP}	CPU Time (s)	$Gap^{LRMP-LIM}$
1	3814	8646	35	55.89%	15,972	28,541	198	44.04%
2	5101	16,857	93	69.74%	5758	11,898	201	51.60%
3	3484	10,465	94	66.71%	10,602	19,892	205	46.70%
4	6108	14,409	93	57.61%	7357	11,098	202	33.71%
5	7285	23,517	106	69.02%	16,097	38,891	205	58.61%
6	7152	23,674	94	69.79%	9309	21,772	199	57.24%
7	4799	11,851	78	59.51%	6713	13,899	212	51.70%
8	6931	12,579	94	44.90%	11,907	26,095	215	54.37%
9	4656	12,047	94	61.35%	10,987	23,886	196	54.00%
10	7675	28,878	95	73.42%	10,205	20,579	202	50.41%
11	6199	21,165	95	70.71%	6682	10,749	214	37.83%
12	6373	19,416	94	67.18%	8905	16,381	211	45.64%
13	4388	12,626	101	65.25%	11,364	22,343	208	49.14%
14	7175	15,513	97	53.75%	9647	19,591	209	50.76%
15	3627	9755	90	62.82%	5895	12,700	216	53.59%
16	5197	15,983	104	67.48%	9638	19,516	215	50.62%
17	5433	14,172	131	61.67%	13,447	23,501	205	42.78%
18	4437	10,721	97	58.61%	6978	14,152	236	50.69%
19	8963	29,816	97	69.94%	5976	11,498	253	48.02%
20	3414	7107	71	51.97%	6044	11,650	226	48.12%
AVG	5611	15,960	93	64.85%	9474	18,932	211	49.96%

into the LRMP, CG began to iterate. During the iteration, whenever any one of the following two criteria was satisfied, the computation terminated: (i) the summation of the objective values of all the sub-problems was larger than -0.1; (ii) CG iterated 500 times. After CG terminated, we obtained a valid lower bound for the IM, which was represented by LB^{LRMP}. In addition, we solved the LIM, which also provided a lower bound for the IM and denoted its value as LB^{LIM}. We calculated the relative gap between LB^{LRMP} and LB^{LIM} using $Gap^{LRMP-LIM} = \frac{LB^{LRMP}-LB^{LIM}}{\max\{LB^{LRMP}, LB^{LIM}\}} \cdot 100\%$.

Based on the results in Table 2, CG generated high-quality lower bounds for IM within minutes. More specifically, on average, for instances in Set-A and Set-B, the average relative gaps between LB^{LRMP} and LB^{LIM} were 64.85% and 49.96%, respectively, which indicated that the fractional solutions of LRMP provided a high-quality baseline for upper bounds search. In the next section, we describe the test for the quality of upper bounds generated using the CGFO based on the columns selected from the LRMP and FO heuristic.

5.3. Performance analysis of the CGFO

To test the performance of the CGFO, including CGFO-I and CGFO-II, we set the termination criteria as follows. Whenever any one of the following two criteria was satisfied, the FO computation terminated: (i) Time limit. For these two types of instances, the computation time limits for CGFO, including the CG process and subsequent FO heuristic, were set to 900 and 1800 s, respectively; (ii) the relative gap of CPLEX while conducting FO was less than 5.00%.

For comparison, the IM was solved using CPLEX under the setting of their respective time limit and a relative gap of 5.00% for instances in the two sets. We represent this method as IM-CPLEX.

With respect to the performance of the algorithm, we focued on the quality of feasible solutions and the computational efficiency. We considered the following indicators: (i) the total computation time; (ii) the relative gap between the upper bounds generated using the CGFO and IM-CPLEX, which was calculated using $UBGap^{CGFO-IM} = \frac{UB^{CGFO}-UB^{IM}}{\min\{UB^{CGFO}, UB^{IM}\}} \cdot 100\%$, where UB^{CGFO} and UB^{IM} represent the upper bounds generated using the CGFO and IM-CPLEX, respectively. The results are summarized in Tables 3, 4, and 5. We conducted the analysis in the following aspects:

- **Comparison between CGFO and IM-CPLEX.** We summarize the upper bounds quality improvement, i.e. the opposite of $UBGap^{CGFO-IM}$, and computation time reduction in Table 3. Detailed results for each instance are provided in Tables 4 and 5. For the instances in the two sets, on average, both CGFO-I and CGFO-II generated upper bounds with a quality higher than that of IM-CPLEX in less computation time, which demonstrates CGFO performs better than IM-CPLEX. For example, CGFO-II generated upper bounds with the average quality higher than IM-CPLEX by 3.34% for the instances in Set-B, with the computation time reduced by 43.96%. In addition, CGFO performed better in terms of convergence than IM-CPLEX. Specifically, the average relative gaps between UB^{CGFO} and LB^{LRMP}, which were calculated using $Gap^{CGFO} = \frac{UB^{CGFO}-LB^{LRMP}}{UB^{CGFO}} \cdot 100\%$, for instances in Set-A and Set-B, were both less than 10.00%, whereas the average relative gaps were 13.65% and 18.64% when using IM-CPLEX.
- **Comparison between CGFO-I and CGFO-II.** On average, for instances in Set-A, the computation time of CGFO-II was less than that of CGFO-I by 51.07%, but the upper bound quality declined by 0.18%. For the instances in Set-B, CGFO-II reduced the computation time of CGFO-I by 34.11%, and, on average, CGFO-II generated upper bounds with a higher quality than CGFO-I by 0.38%. That is because for some hard instances, e.g. instances 5, 7, 8, and 9, whose upper bounds generated using CGFO-I were poor owing to the large scale, CGFO-II reduced more solution space and searches for better upper bounds in the limited computation time, although using up the given computation time.

In conclusion, on average, CGFO outperforms CPLEX in terms of the quality of upper bounds, i.e. feasible solutions, and the computational efficiency. CGFO-II has advantages over CGFO-I with respect to the computational efficiency and it can generate high-quality upper bounds, which are comparable to those generated by CGFO-I, or even better when the computation time is relatively limited.

5.4. Sensitivity analysis on the processing time

In this part, we conduct sensitivity analysis on the curing time when using BLDP to solve the sub-problems. Based on the

Table 3
Results Comparison: improvement on upper bounds & computation time reduction.

Instance Set	Improvement on upper bounds		Computation time reduction	
	CGFO-I	CGFO-II	CGFO-I	CGFO-II
Set-A	2.25%	2.07%	36.74%	69.04%
Set-B	2.96%	3.34%	14.55%	43.96%

Table 4
Performance of upper bounds: Set-A.

Instance No.	IM-CPLEX		CGFO-I			CGFO-II		
	UB^{IM}	CPU Time (s)	UB^{CGFO}	$UBGap^{CGFO-IM}$	CPU Time (s)	UB^{CGFO}	$UBGap^{CGFO-IM}$	CPU Time (s)
1	8955	901	9154	2.22%	73	9272	3.54%	309
2	19,130	900	18,722	-2.18%	334	18,975	-0.82%	148
3	12,075	900	11,296	-6.90%	901	11,367	-6.23%	382
4	15,393	900	15,282	-0.72%	888	15,623	1.49%	171
5	27,054	900	26,623	-1.62%	900	26,269	-2.99%	334
6	27,856	900	27,135	-2.65%	144	27,031	-3.05%	101
7	12,816	900	12,779	-0.28%	165	12,837	0.16%	384
8	14,578	900	13,452	-8.37%	632	13,858	-5.19%	126
9	13,121	900	13,211	0.69%	206	13,337	1.64%	95
10	34,981	900	33,370	-4.83%	901	32,947	-6.17%	529
11	23,900	900	24,222	1.35%	902	24,152	1.05%	295
12	20,509	900	20,816	1.50%	702	20,874	1.78%	127
13	15,133	900	14,977	-1.04%	901	14,501	-4.36%	817
14	16,206	900	16,111	-0.59%	131	16,097	-0.67%	108
15	11,126	902	10,813	-2.89%	822	10,850	-2.54%	365
16	17,163	900	17,012	-0.89%	775	17,109	-0.32%	504
17	16,741	900	16,237	-3.11%	131	16,241	-3.08%	131
18	13,227	900	12,112	-9.20%	900	12,425	-6.45%	314
19	33,203	900	31,802	-4.41%	900	31,983	-3.81%	257
20	7401	901	7521	1.61%	84	7521	1.61%	76
AVG	18,028	900	17,632	-2.25%	570	17,663	-2.07%	279

Table 5
Performance of upper bounds: Set-B.

Instance No.	IM-CPLEX		CGFO-I			CGFO-II		
	UB^{IM}	CPU Time (s)	UB^{CGFO}	$UBGap^{CGFO-IM}$	CPU Time (s)	UB^{CGFO}	$UBGap^{CGFO-IM}$	CPU Time (s)
1	31,079	1801	30,057	-3.40%	1667	30,455	-2.05%	680
2	12,562	1805	12,677	0.91%	550	12,694	1.05%	515
3	21,793	1805	21,331	-2.17%	1525	21,535	-1.20%	477
4	12,208	1801	12,123	-0.70%	403	12,373	1.35%	230
5	47,220	1802	45,779	-3.15%	1801	43,394	-8.82%	1802
6	25,652	1801	23,158	-10.77%	1802	25,627	-0.10%	462
7	15,105	1801	14,789	-2.14%	1710	15,062	-0.28%	785
8	33,496	1801	32,911	-1.78%	1801	30,752	-8.92%	1805
9	32,159	1803	30,107	-6.81%	1802	28,906	-11.25%	1802
10	24,413	1800	24,224	-0.78%	1804	23,600	-3.44%	1803
11	11,926	1801	11,914	-0.10%	817	11,993	0.57%	244
12	17,805	1801	17,184	-3.62%	1665	17,657	-0.84%	608
13	25,920	1800	23,809	-8.87%	1805	23,762	-9.08%	1698
14	21,179	1801	21,411	1.10%	1802	21,375	0.93%	871
15	14,114	1801	13,878	-1.70%	1802	14,340	1.61%	1103
16	21,805	1801	20,895	-4.36%	1802	21,030	-3.68%	1677
17	25,184	1801	25,428	0.97%	1803	25,617	1.72%	1294
18	15,800	1802	15,385	-2.70%	1801	15,265	-3.50%	1688
19	12,550	1810	12,659	0.87%	832	12,546	-0.04%	273
20	12,753	1802	12,526	-1.81%	1803	12,688	-0.52%	476
AVG	21,736	1802	21,112	-2.96%	1540	21,034	-3.34%	1015

configuration of Set-B, we generated three sets of instances with $|T|$ set to 15, 20, and 25, respectively. In each set of instances, we set the lead time lt_i to five levels, in the range from 1 to 5; thus, we conducted 15 groups of experiments. We generated 100 instances for each group. The values of dual variables were generated randomly. More specifically, α was generated in $U(1000, 5000)$ and β was set to 0 as it is a constant that has no impact on the calculating process. We recorded the total computation time for each group. The results are shown in Fig. 3, based on which we make the following several observations:

- **Computational efficiency.** Generally, BLDP solved the sub-problems in each group in less than 0.10 s, while CPLEX needed more than 10 s. This indicated that BLDP has a noticeable advantage over CPLEX in terms of computational efficiency when solving sub-problems.
- **Sensitivity analysis of BLDP.** In each type of instance, the computation time for BLDP decreased slightly when the lead time increased. This was because when $|T|$ settled and the lead time increased, the distance between each pair of nodes linked by paths in the routing graph was larger; therefore, we

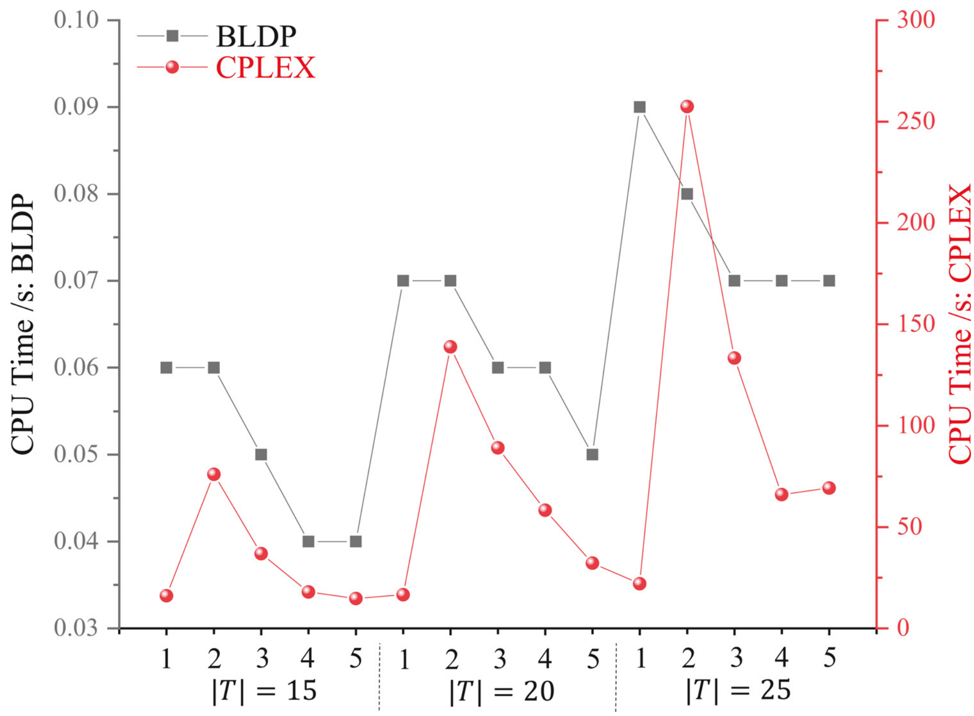

Fig. 3. Sensitivity analysis.

Table 6
Results Comparison: Cases UTP-8 and UTP-4.

Instance No	UTP	Production Cost	Holding Cost	Backorder Cost	Total Cost	CPU Time (s)
1	8 h	8500	222	4383	13,106	797
	4 h	9512	218	3175	12,904	246
2	8 h	9326	350	14,147	23,823	1802
	4 h	11,233	438	8816	20,487	1802
3	8 h	9135	362	3014	12,511	1803
	4 h	9245	286	3014	12,545	1802
4	8 h	13,289	508	7944	21,741	1800
	4 h	14,914	541	4992	20,447	1802
5	8 h	13,672	655	5904	20,231	648
	4 h	14,520	590	4896	20,005	1805
6	8 h	11,845	322	2264	14,431	313
	4 h	11,234	356	2744	14,334	393
7	8 h	9105	379	1557	11,042	367
	4 h	8837	321	1557	10,715	503
8	8 h	10,146	443	7388	17,977	1802
	4 h	12,818	621	3498	16,937	1802
9	8 h	8220	448	7020	15,688	549
	4 h	8646	274	6801	15,721	1479
10	8 h	10,998	284	10,062	21,344	1802
	4 h	14,305	452	4580	19,337	1801
AVG	8 h	10,424	397	6368	17,189	1168
	4 h	11,526	410	4407	16,343	1343

constructed a routing graph with fewer paths and conducted fewer recursive options in Level 2 of BLDP.

- **Sensitivity analysis of CPLEX.** In each type of instance, as the lead time increased, the computation time for CPLEX initially increased and then decreased. This was because when the lead time was set to 1, the production statuses of each autoclave between adjacent periods were decoupled, making the sub-problem relatively easier to solve. When all the lead times were set to 1, the decision variables Z could be removed from the model. When the lead time was larger than 1, higher computational complexity occurred owing to the linkage of the production statuses between adjacent periods for each autoclave; thus, CPLEX needs significantly more time to solve the sub-problem. However, with the increase of lead time, it may reduce larger solution space while conducting each branching operation especially when $Y_{ut} = 1$, resulting in the decrease of computation time.

In conclusion, BLDP performs quite well in terms of stability and computational efficiency.

5.5. Case study

In this section, we investigate a case that is encountered in the factory we collaborate with, i.e. the decrease in the cost when shortening the length of the UTP in the planning horizon. Currently, the factory is running under the setting of three shifts a day and creates a rolling production plan every five days. In this scenario, we would like to test what would occur if we set the UTP to 4 h (hereafter we call it "Case UTP-4") rather than 8 h ("Case UTP-8"). Theoretically, the total costs will be reduced as the solution space is augmented. However, the extent of the reduction in costs is an unanswered question and very appealing to the managers. In addition, the increase in computation time must be tested. Among the cured items we collected, the curing process required 2, 6, and 12 h. In Case UTP-8, the corresponding lead times were 1, 1, and 2 period(s), whereas in Case UTP-4, the lead times were 1, 2, and 3 period(s). We generated 10 instances based on the configuration of Set-B except where $|T|$ was set to 15, corresponding to Case UTP-8. We then transformed these instances into Case UTP-4 according to the following rules: (i) $|T| = 30$; (ii) the external demand was set to $d_{j,2t}, \forall t \in T, j \in End_p$; (iii) the holding (backorder) cost for an item in each period was set to $\frac{hc_i}{2}\left(\frac{bc_j}{2}\right), \forall i \in I, j \in End_p$. The termination criteria were the same as those for the instances in Set-B. The results related to costs and computation time are shown in Table 6.

The results in Table 6 indicate that, compared with Case UTP-4, the average total costs of instances in Case UTP-8 decreased by 4.92%. The backorder cost decreased by 30.79%, and the production cost for autoclaves increased by 10.58%. This was because when shortening the UTP, the curing process could be managed more accurately. The unproductive time of autoclaves decreased (e.g. for the item with curing time equal to 2 h, we had 6 h of idle time in Case UTP-8, whereas we could run the autoclave twice in Case UTP-4 with a total wasted time equal to 4 h), and it provided more flexibility and room to match capacity with demands. Moreover, the average computation time increased by 14.99%, as the solution space was augmented. However, the average computation time was still significantly less than that of IM-CPLEX. In conclusion, it is worth running the factory with UTP changed to 4 h. This will reduce the total costs with a slight increase on the computation time.

6. Conclusion and future research

This paper studies a practical problem from the aviation industry, which is defined as an integrated bin packing and lot-sizing problem and is characterized by the re-configuration of bins. More specifically, in addition to the typical characteristics of the BPP and LSP, this paper considers the configuration-dependent processing time in the bin packing process, making the problem very complicated and difficult to solve. We establish an IM and decompose it into one MP and $|M|$ sub-problems based on the Dantzig–Wolfe framework. CG is used to solve the LMP to generate tighter lower bounds compared with the LIM. A high-efficiency dynamic programming algorithm (BLDP) is proposed to solve the sub-problems exactly. A heuristic algorithm called CGFO and its variants, i.e. CGFO-I and CGFO-II, are proposed based on the columns selected from the LRMP and FO heuristic. The quality of lower bounds provided by the LRMP and computational efficiency of CG are evaluated. CGFO-I and CGFO-II are tested on the instances generated from real data and perform better than IM-CPLEX in terms of the quality of upper bounds and computational efficiency. For example, CGFO-II, which is more efficient than CGFO-I, can improve the quality of upper bounds by 3.34% while reducing the computation time by 43.96% when compared with IM-CPLEX for instances in Set-B. Moreover, we conducted a case study based on the real practice, which indicated that when shortening each period in the planning horizon to 4 h rather than 8 h, the idle time of autoclaves will be reduced, resulting in a decrease in the total costs by 4.92% on average.

For further study, two aspects are worth investigating. In this study, the trays were placed horizontally along the autoclaves in a single row. It is necessary to consider placing them in multiple rows in the autoclaves with a larger width. In addition, the configuration-dependent processing time required in the bin packing process is given and certain in this research. A stochastic version deserves further research.

Acknowledgments

This work was supported by the National Key R&D Program of China under grant No. 2018AAA0101705, and Shenzhen Municipal Science and Technology Innovation Committee under grant No. WDZC20200821140447001.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ejor.2022.03.012.

References

Alem, D., & Morabito, R. (2013). Risk-averse two-stage stochastic programs in furniture plants. *OR Spectrum, 35*(4), 773–806.
Andonov, R., Poirriez, V., & Rajopadhye, S. (2000). Unbounded knapsack problem: Dynamic programming revisited. *European Journal of Operational Research, 123*(2), 394–407.
Arbib, C., & Marinelli, F. (2005). Integrating process optimization and inventory planning in cutting-stock with skiving option: An optimization model and its application. *European Journal of Operational Research, 163*(3), 617–630.
De Carvalho, J. V. (2002). Lp models for bin packing and cutting stock problems. *European Journal of Operational Research, 141*(2), 253–273.
de Lara Andrade, P. R., de Araujo, S. A., Cherri, A. C., & Lemos, F. K. (2021). The integrated lot sizing and cutting stock problem in an automotive spring factory. *Applied Mathematical Modelling, 91*, 1023–1036.
Degraeve, Z., & Jans, R. (2007). A new Dantzig-Wolfe reformulation and branch-and-price algorithm for the capacitated lot-sizing problem with setup times. *Operations research, 55*(5), 909–920.
Delorme, M., Iori, M., & Martello, S. (2016). Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research, 255*(1), 1–20.
Gramani, M., França, P., & Arenales, M. (2009). A lagrangian relaxation approach to a coupled lot-sizing and cutting stock problem. *International Journal of Production Economics, 119*(2), 219–227.
Gramani, M. C. N., França, P. M., & Arenales, M. N. (2011). A linear optimization approach to the combined production planning model. *Journal of the Franklin Institute, 348*(7), 1523–1536.
Khezri, A., Benderbal, H. H., & Benyoucef, L. (2021). Towards a sustainable reconfigurable manufacturing system (SRMS): Multi-objective based approaches for process plan generation problem. *International Journal of Production Research, 59*(15), 4533–4558.
Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., & Van Brussel, H. (1999). Reconfigurable manufacturing systems. *CIRP Annals, 48*(2), 527–540.
Leao, A. A., Furlan, M. M., & Toledo, F. M. (2017). Decomposition methods for the lot-sizing and cutting-stock problems in paper industries. *Applied Mathematical Modelling, 48*, 250–268.
Melega, G. M., de Araujo, S. A., & Jans, R. (2018). Classification and literature review of integrated lot-sizing and cutting stock problems. *European Journal of Operational Research, 271*(1), 1–19.
Melega, G. M., de Araujo, S. A., & Morabito, R. (2020). Mathematical model and solution approaches for integrated lot-sizing, scheduling and cutting stock problems. *Annals of Operations Research, 295*(2), 695–736.
Nonås, S. L., & Thorstenson, A. (2008). Solving a combined cutting-stock and lot-sizing problem with a column generating procedure. *Computers & Operations Research, 35*(10), 3371–3392.
Oliveira, W. A., Fiorotto, D. J., Song, X., & Jones, D. F. (2021). An extended goal programming model for the multiobjective integrated lot-sizing and cutting stock problem. *European Journal of Operational Research, 295*(3), 996–1007.
Pierini, L. M., & Poldi, K. C. (2021). An analysis of the integrated lot-sizing and cutting-stock problem formulation. *Applied Mathematical Modelling, 99*, 155–165.
Poltroniere, S. C., Araujo, S. A., & Poldi, K. C. (2016). Optimization of an integrated lot sizing and cutting stock problem in the paper industry. *TEMA (São Carlos), 17*(3), 305–320.
Poltroniere, S. C., Poldi, K. C., Toledo, F. M. B., & Arenales, M. N. (2008). A coupling cutting stock-lot sizing problem in the paper industry. *Annals of Operations Research, 157*(1), 91–104.

Vanzela, M., Melega, G. M., Rangel, S., & de Araujo, S. A. (2017). The integrated lot sizing and cutting stock problem with saw cycle constraints applied to furniture production. *Computers & Operations Research, 79*, 148–160.

Wu, T., Akartunalı, K., Jans, R., & Liang, Z. (2017). Progressive selection method for the coupled lot-sizing and cutting-stock problem. *INFORMS Journal on Computing, 29*(3), 523–543.

Wu, T., Xiao, F., Zhang, C., He, Y., & Liang, Z. (2018). The green capacitated multi-item lot sizing problem with parallel machines. *Computers & Operations Research, 98*, 149–164.

Xie, F., Wu, T., & Zhang, C. (2019). A branch-and-price algorithm for the integrated berth allocation and quay crane assignment problem. *Transportation Science, 53*(5), 1427–1454.