INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

IFORS

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 28 (2021) 2191–2217
DOI: 10.1111/itor.12645

Multi-level, multi-stage lot-sizing and scheduling in the flexible flow shop with demand information updating

Hakeem-Ur-Rehman, Guohua Wan* and Yang Zhan

Antai College of Economics and Management, Shanghai Jiao Tong University, 1954 Huashan Road, Shanghai 200030, China

E-mail: hakeem@mail.sjtu.edu.cn [Hakeem-Ur-Rehman]; ghwan@sjtu.edu.cn [Wan]; zhanyangjy@sjtu.edu.cn [Zhan]

Received 28 February 2018; received in revised form 12 February 2019; accepted 12 February 2019

Abstract

Motivated by a real-world problem in an automobile manufacturing firm, we study the multi-level, multi-stage lot-sizing and scheduling problem with demand information updating. The objective is to determine the quantities and sequences of production on different production stages to minimize the total costs of production and inventory. We employ the martingale model for forecast evolution to model the evolving demand over time and build a mixed-integer programing (MIP) model for the problem using a hybrid period approach (i.e., combining micro- and macro-periods). To solve this NP-hard problem, we propose three heuristic algorithms based on the idea of "relax-and-fix" within the rolling horizon framework. We conduct computational experiments to test the performance of the heuristic algorithms as well as compare the performance of the proposed heuristics with the benchmark procedure (truncated procedure for MIP) for both small- and large-scale problem instances. The computational results show that Heuristic 1 performs the best among the three heuristic algorithms.

Keywords: lot-sizing and scheduling (LSS); flexible flow shop; demand information updating; relax-and-fix; rolling horizon

1. Introduction

In a manufacturing system, one of the most important problems in production planning and control is lot-sizing and scheduling (LSS), that is, determining the production quantities, machine assignment (in case of multiple machines), and production sequence to fulfill the demand with minimal costs of production and inventory. LSS is important due to its profound impact on the productivity of the manufacturing system. However, in most studies on LSS, the three LSS decisions (quantity, assignment, and sequence) are made separately, thus lead to suboptimal solutions (Dauzère-Pérès and Lasserre, 2002; Quadt and Kuhn, 2005). To obtain global optimal solutions, it is necessary to

*Corresponding author.

make integrated LSS decisions (i.e., simultaneously considering the production quantities, machine assignment, and production sequence) (Stadtler, 2005).

Integrating the LSS problem is necessary when setup times and/or costs are sequence dependent during production (Fandel and Stammen-Hegene, 2006). Furthermore, in a multi-stage production setting, different product combinations may use the machines at different times with different capacity requirements due to different configuration of the bill-of-materials (BOM), thus lead to shifting bottlenecks in the production shop. Therefore, it is more required to simultaneously consider the three LSS decisions (Seeanner et al., 2013).

In the literature, the LSS problem with deterministic demands over the planning horizon receives considerable attention (see, e.g., Seeanner and Meyr, 2013; Almeder et al., 2015). However, to the best of our knowledge, the integrated LSS problem with demand information updating has yet been studied and it is our focus in this study.

For production planning and scheduling problems, demand forecasting is the key input driving the planning and scheduling decisions. On the one hand, the errors of forecasting become larger as the planning horizon (h) increases (i.e., $\sigma_{t,t} \leq \sigma_{t,t+1} \leq \sigma_{t,t+2} \leq \cdots \leq \sigma_{t,t+h}$); on the other hand, the errors of forecasting decrease over time as more information becomes available (i.e., $\sigma_{t-h,t} \geq \sigma_{t-h+1,t} \geq \sigma_{t-h+2,t} \geq \cdots \geq \sigma_{t,t}$). Motivated by a real-world problem from a supplier of internal parts for automobile manufacturing, we consider a multi-level, multi-stage LSS problem with demand information updating. Here, the supplier produces quite a few types of products in its flexible flow shop, that is, a flow shop with one or more parallel machines in each stages connected by finite interstage buffers. A flexible flow shop is able to produce many types of products while keeping the product processing route flexible (Quadt and Kuhn, 2005).

To deal with this problem, although a general MIP formulation with commercial optimization packages (such as CPLEX and Gurobi) can be used to obtain solutions, normally it can only solve small-sized problems. A strong MIP formulation, together with efficient algorithm, is needed to solve large-scale LSS problems, in particular, when there exist sequence-dependent setup times. In the literature, there are two approaches, namely, discrete and continuous-time approaches, to modeling and solve the problem (e.g., Camargo et al., 2012). In OR community, researchers largely use the discrete time approach, where the finite planning horizon is divided into a number of small (e.g., one hour or one day) or large (e.g., one week or one month) periods, with the aim at finding the optimal tradeoff between inventory holding cost and the production cost (see Copil et al., 2017 for a compressive review). The process systems engineering community mainly adopts the continuous-time approach (Méndez et al., 2006). Hybrid models that combine these two approaches also exist (e.g., Günther, 2014). The key point is the choice of appropriate time structure in conjunction with the characteristics of the production system. The purpose of this study is to investigate a two-level hierarchical production planning problem (Stadtler, 2005), with the following tasks:

(i) computing the product demand over the planning horizon with information updating;
(ii) developing a mathematical model for the multi-level, multi-stage simultaneous LSS problem;
(iii) designing and implementing effective heuristic algorithms for solving the problem and conducting computational experiments to test the performance of the algorithms.

The remainder of the paper is organized as follows. Section 2 briefly describes the related studies in the literature. Section 3 is devoted to description of the LSS problem with demand information

Table 1
Classification of SLSS models

Time structure	Number of setups allowed	SLSS model	Reference
Small time bucket (STB) (microperiod)	At most one per period	Continuous setup lot-sizing problem (CSLP)	Karmarkar and Schrage (1985)
		Discrete lot-sizing and scheduling problem (DLSP)	Fleischmann (1990)
		Proportional lot-sizing and scheduling problem (PLSP)	Drexl and Haase (1995)
Large time bucket (LTB) (macroperiod)	Any number of setups per period	Capacitated lot-sizing problem with sequence-dependent setup costs (CLSD)	Haase (1996)
		General lot-sizing and scheduling problem (GLSP)	Fleischmann and Meyr (1997)

updating as well as to formulating the model for the problem. In Section 4, we elaborate the ideas of solution algorithms and propose three heuristic algorithms. In Section 5, we present the results of the computational experiments. We conclude in Section 6 with discussions of the future research directions.

2. Literature review

The literature review is organized into two streams, that is, the literature on demand forecasting with information updating, and the literature on LSS problem and the solution approaches.

The literature on demand forecasting with information updating can be classified into three categories: (1) time series models, (2) Quasi-Markovian or Martingale models, and (3) Bayesian statistics models (Sethi et al., 2005). Milner and Kouvelis (2005) argue that the behavior of these three types of models are different with respect to the flexibility lot-sizing (i.e., production quantity) and scheduling (i.e., timing and sequencing). All the three approaches require both forms of flexibility to significantly decrease the total cost of the LSS problem with evolving demands. In this paper, we use a simple yet deceptively powerful probabilistic model called the martingale model for forecast evolution (MMFE) to model the evolution of demand forecast over time. MMFE employs a decomposition technique to quantify the uncertainty of each time period within the planning horizon proposed by Graves et al. (1986) and Heath and Jackson (1994) independently. It can be viewed as an extension of Hausman (1969) model. The advantage of MMFE is that it incorporates the exogenous factors to determine the evolving demand and is proven to be a good method for effectively reducing the forecast errors. Many researchers use MMFE to analyze the production and inventory systems with demand information updating (see, e.g., Iida and Zipkin, 2006; Altug and Muharremoglu, 2011; Norouzi and Uzsoy, 2014).

The LSS problems has received extensive attention in the last two decades, starting with the ground-breaking work of Karmarkar and Schrage (1985). Since then, single-level single-stage LSS problems are the main focus of research. Table 1 is a summary of single-level lot-sizing and scheduling

(SLSS) models with respect to the representation of the time structure and a maximum number of setups allowed per period.

The advantage of the small time bucket (STB) models is that they automatically determine the sequence of lots due to the natural order of the microperiods. However, the length of microperiods must be chosen carefully as setup time is usually limited to its span. If the length of microperiod is too short, large setup times cannot be modeled. This shortcoming of the STB models is avoided by large time bucket (LTB) models as they use a small number of long macroperiods, which allow several production lots in a macroperiod. The LTB models typically employ two ways to determine the production sequence. In the first approach, the long macroperiods are further decomposed into a number of microperiods and allow only one setup per microperiod (e.g., Fleischmann and Meyr, 1997), whereas in the second approach, to determine the sequence in each macroperiod, additional constraints are added in the model as those in traveling salesman problem to eliminate the subtours (e.g., Haase, 1996). Wolsey (2002) presents a classification of different STB and LTB models and illustrates that the linear program relaxation of STB models produces weak lower bounds, but can be improved with strong valid inequalities and customized reformulation of the problem. In addition, LTB models may deliver much better lower bounds (Almeder and Almada-Lobo, 2011).

For comprehensive literature reviews on single-level LSS problems, the readers are referred to Copil et al. (2017), Drexl and Kimms (1997), Quadt and Kuhn (2008) and Zhu and Wilhelm (2006). In recent years, scholars have been studying the multi-level and the multi-stage LSS problems, natural extensions of the single-level problems. In the following, we mainly focus on the literature on the multi-stage, multi-level integrated LSS problem.

Haase (1994) and Kimms (1996) present the multi-stage STB models, where inventory balancing constraints are used to synchronize the different production stages and setup times are neglected while lead times of at least one microperiod are made compulsory. The variant of the LSS problem that allows for proportional processing times and "zero lead times" is proposed by Stadtler (2011) to solve the real-world problem arising from pharmaceutical industry. Chang et al. (2004) adopt a multi-level BOM PLSP on multiple machines based on the formulation of Kimms (1999) and introduce the lot-size upper bounds as well as capacity allocation rules for the products. Kaczmarczyk (2013) proposes a new MIP model for the LSS problem with problem characteristics such as identical parallel machines and allowed overlapping of set-up times on two consecutive periods. For material requirements planning, Clark (2003) studies a capacity feasible master production schedule problem using model for multi-stage and multi-level BOM. For a similar problem as in Clark (2003), Almeder et al. (2015) study two alternative models. They claim that the classical formulations for the multi-level capacitated lot-sizing problem often improperly capture resource requirements and precedence relationships if the order lead times are taken into consideration, because either it delivers infeasible production plans or plans with high inventory holding cost. In order to deal with this issue, they propose two MIP models, where in the first model, batch production is considered, and in the second model, lot-streaming is allowed. Fandel and Stammen-Hegene (2006) tried to combine the advantages of multi-level proportional LSS problem and continuous LSS problem, and developed a model to accurately calculate the inventory holding cost that leads to a nonlinear formulation. This model allows multi-product, multi-level integrated LSS in a job shop. Mohammadi (2010) and Mohammadi and Jafari (2011) propose models for hybrid flow shop based on Fandel and Stammen-Hegene (2006). They also divide each microperiod into three parts (i.e., setup, production, and idle times) where each macroperiod contains a predefined number of microperiods, making the problem relatively easier to solve. Meyr (2004a) extends the model of Fleischmann and Meyr (1997) to the case of multi-level and multi-stage. To enhance this model, Seeanner and Meyr (2013) introduce the constraints for synchronizing the multiple production stages at machine level and provide heuristics to solve the problem. Lang (2010) presents an extension of Meyr (2004a) with flexible production sequence and product substitution.

The existing solution methods in the literature for solving the integrated LSS problem, normally formulated as a MIP model, vary from exact optimization methods to heuristic approximation methods. The former relays on strong formulations of the problems, which help to solve them fast and efficiently. In principle, these methods are able to provide exact solutions (e.g., Araujo and Clark, 2013; Belo-Filho et al., 2014). The latter is more suitable in practice because it quickly discovers the near-optimal solutions to the problem. Many researchers use MIP-based heuristics such as rolling horizon (RH), relax-and-fix, fix-and-optimize to solve the problem, and some researchers combine these heuristics with meta-heuristics such as variable neighborhood search and simulated annealing (e.g., Sahling et al., 2009; Lang and Shen, 2011; Ramezanian and Saidi-Mehrabad, 2013; Seeanner et al., 2013; Xiao et al., 2013; Chen, 2015; Armellini et al., 2020). Araujo and Clark (2013) provide a facility location reformulation and strengthen it with the newly proposed constraints to obtain the better solution for the LSS problem. The proposed formulation is solved on a RH basis using the "relax-and-fix" method. Ferreira et al. (2009) consider a two-stage integrated LSS problem arising from soft drink manufacturing and present several heuristics such as "relax-and-fix" to obtain the feasible solution to the problem. Based on the iterative procedure, Mercé and Fontan (2003) propose two MIP-based heuristics within the RH framework to solve the capacitated lot-sizing problems, where the finite planning horizon is divided into three parts (i.e., beginning, central, and ending) at each step of the process. In the beginning and ending parts, selected freezing and simplifying strategies are applied, respectively, and the whole model is solved in the central part. Beraldi et al. (2008) study the LSS problem with sequence-dependent setup costs on identical parallel machines. They provide "rolling-horizon-" and "relax-and-fix-"based heuristics to efficiently solve the problem.

Belo-Filho et al. (2014) propose two MIP models for a capacitated lot-sizing problem with backlogging, setup carryover, and crossover. The first is similar to existing models in the literature while the second uses the idea of disaggregation of the setup variable. The results of their computational experiments show the superiority of the proposed models over the existing state-of-art formulation. Similar to Belo-Filho et al. (2014), Ramya et al. (2016); Fiorotto et al. (2017) also studied the capacitated lot-sizing problem with characteristics of setup carryover, crossover, and splitting. While considering the problem characteristics, they also propose MIP formulations and present their analyses. Considering problem characteristics such as nontriangle and sequence-dependent setup time and cost that can carry over, and the overlap on different production line settings, Mahdieh et al. (2018) proposed an MIP model for the capacitated LSS problem. They also conduct computational experiments to show the flexibility of the models. Moreover, recently, a few researchers study the integrated LSS problem with stochastic demand via robust optimization and/or stochastic programming models (e.g., Hu and Hu, 2016; Alem et al., 2018; Curcio et al., 2018). Ramezanian and Saidi-Mehrabad (2013) study the stochastic LSS problem in a capacitated multi-stage production system and propose an MIP model using chance-constrained programing theory.

To the best of our knowledge, the existing literature on integrated LSS problem is normally concerned with deterministic or stochastic demands. Different from these study, we consider an integrated LSS problem with *dynamic demand information updating*.

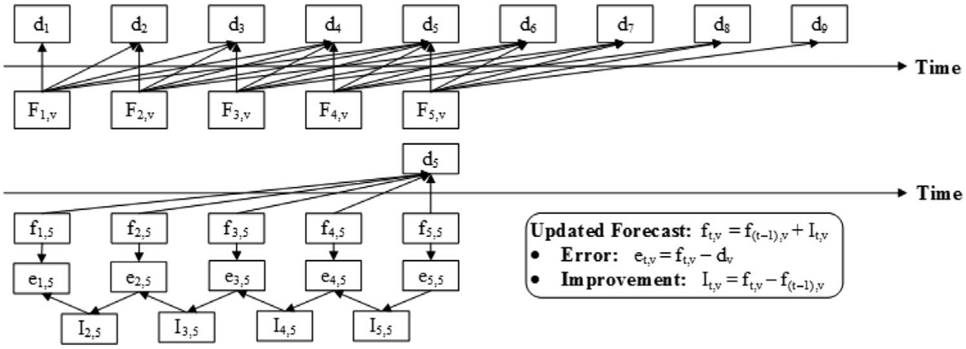

Fig. 1. Schematic evolution of uncertain demand forecast.

3. Problem description and mathematical model

3.1. Problem description

We consider a multi-stage integrated LSS problem in a flexible flow shop with demand information updating, where the demand is forecasted using historical data but updated with more available information over time, and backorders are not allowed. The objective of the problem is to generate a production schedule to fulfill the demand while minimize the total costs of production, setup, and inventory. In the problem, the demand forecasts are updated and the production plan is generated on RH basis. Obviously, this problem is NP-hard since $FFs||C_{max}$ is a special case of this problem.

3.2. Model for demand forecasting

The demand for a product evolves over time due to a number of cumulative factors. Such factors may include competitive offerings, advertising and marketing efforts, changes in the business settings, and demand noises. Suppose the demand e_v is normally distributed with mean 0 and variance σ^2 and arrives as time progresses, $v = 1, 2, \ldots, H$. Then, $d(H) = \overline{d}_0 + \sum_{v=1}^{H} e_v$ is the demand in period H, where $\overline{d}_0$ is the initial demand rate. Using currently available information, the demand forecasts can be obtained for the succeeding time periods. The forecasts are then dynamically updated as time progresses and more demand information becomes available. Such a dynamic forecast updating process can be modeled using MMFE. The model has two parameters, that is, the initial forecast vector and the variance–covariance (VCV) matrix. The mathematical description of MMFE can be sketched as follows.

Suppose $f_{t,v}$ represents the demand forecast for the period v made from the period t (i.e., $t \leq v$). A vector ($F_{t,v} = [f_{t,t}, f_{t,(t+1)}, f_{t,(t+2)}, \ldots, f_{t,(t+H)}]$, where H represents the forecast horizon) is the result of multiple forecasts for period v made from period t. Suppose in the subsequent periods $t+1, t+2, \ldots,$ the forecasts $F_{(t+1),v}, F_{(t+2),v}, \ldots,$ are obtained. As an example, an RH for the demand forecasts with $H = 4$ is shown in Fig. 1. In the figure, $f_{t,v}$ is continuously improved by $I_{t,v}$ (i.e., new information helps improving the demand forecast) as t increases from $t - H$ to v (depicted in the lower part of the figure).

In this paper, we assume that the demand for each end item (i.e., final product j) follows the AR(1) process. The autoregressive time series model with one period lag AR(1) can be formulated as follows:

$$d_t = \mu(1 - \rho) + \rho d_{(t-1)} + e_t, \tag{1}$$

where d_t is the demand of a product in period t, $e_t \sim N(0, \sigma^2)$ is the white-noise of the process, and $0 \leq \rho \leq 1$. In inventory management literature, the pattern of lag-1 autoregressive time series is commonly used (see, e.g., Lee et al., 2000; Miyaoka and Hausman, 2004; Ali et al., 2012). The range of AR(1) demand process can be captured by varying the value of ρ. For example, $\rho = 0$ indicates the case of independent and identically distributed demands. The value of $\rho > 0$ indicates the intertemporal relationship between periods $(t - 1)$ and t demands.

We apply the MMFE framework on AR(1) process to capture the advanced demand information (i.e., the forecast updates) (see Appendix). The updated demands over the time periods are generated using the above process.

3.3. Multi-level, multi-stage simultaneous lot-sizing and scheduling problem

The multi-stage integrated LSS problem in flexible flow shop can be found in many discrete as well as continuous manufacturing systems, such as an automobile, steel-making, electronics, and textile. We use a hybrid (i.e., combining micro and macro) period-based mathematical model, called multi-level multi-stage simultaneous lot-sizing and scheduling problem (MMSLSP) for modeling the problem. To minimize the total costs of LSS with updated information about demand, the following decisions must be made: (i) each product's updated demand in macroperiod, (ii) the lot size for each product, and (iii) the sequence of the lots.

The main assumptions of the model are as follows. Types of products are produced through sequentially arranged production stages in a capacitated flexible flow shop. Each production stage may contain heterogeneous (i.e., nonidentical, with respect to production rates, setup time, etc.) parallel machines that are capable to produce all the types of products; these machines are available all the time during the planning horizon. Lot-splitting is allowed at any production stage. In other words, at any given production stage, more than one machine can be used, simultaneously, to produce each lot of the products. The forecasted independent demand for the final products (processed by the last production stage) is evolved over the planning horizon in which the demand must be satisfied in the end of each macroperiod. The required quantity of the predecessor materials/components must be ready before the production of the successor components/products. Backorders are not allowed. A sequence-dependent setup time and setup cost incur if a machine switches from one product to another, and the time and cost vary among the machines. Setup state is carried over on each machine in all the production stages among adjacent periods. Single product production is possible over multiple periods (i.e., it is possible to have a setup from a product to itself, so a single setup for the state conversion allows production lots to span over multiple periods). Moreover, the production state after setup is not lost even crossing over the idle time.

The indices, parameters, and data and variables used in the model are summarized as follows.

Notations	Definition
Indices	
i, i', j, j'	Product types (=1, 2, . . . , N)
l	Production stage (=1, 2, . . . , L)
m, k	Parallel machines (=1, 2, . . . , K)
u, v, t	Macroperiods (=1, 2, . . . , T)
s	Microperiods (=1, 2, . . . , S)
Index sets	
S_t	Set of microperiods that are part of macroperiod t
N_t	Set of products that can be produced on production stage l
K_t	Set of all machines belonging to production stage l
Λ	Set of all final products (i.e., end products)
$\Gamma(j)$	Set of immediate successors of product j based on BOM
$\Omega(j)$	Set of immediate predecessors of product j based on BOM
Parameters and data	
a_{klj}	Production time required on machine k at stage l to produce one unit of product j
c^p_{kljs}	Unit production cost of product j on machine k at stage l in microperiod s
h_{jt}	Per unit per macroperiod t holding cost of product j
s_{klij}	Setup cost of a changeover from product i to product j on machine k at stage l
st_{klij}	Setup time of a changeover from product i to product j on machine k at stage l
c^b_{kl}	Standby (idle) costs on machine k at stage l
M_{klj}	Minimum lot-size of product j on machine k at stage l
p_{ji}	(Gozinto factor) Number of units of product j produced are required to produce one unit of product i
d_{jt}	Updated demand for each end-product in macroperiod t (Qty)
C_{klt}	Production capacity on machine k at stage l in macroperiod t (time)
$bigM$	A big number ($bigM \to \infty$)
Variables	
$I_{jt} \geq 0$	Inventory of product j at the end of macroperiod t
$st^b_{kls} \geq 0$	Starting portation of setup time for changeover in microperiod s on machine k at stage l
$st^e_{kls} \geq 0$	Ending portation of setup time for changeover in microperiod s on machine k at stage l
$sb_{kls} \geq 0$	Standby (idle) time on machine k at stage l in microperiod s
$Q_{kljs} \geq 0$	Total quantity of product j produced in microperiod s on machine k at stage l
$y_{kljs} \in \{0, 1\}$	$y_{kljs} = 1$ if machine k at stage l is setup for product j in microperiod s (0 otherwise)
$z_{klijs} \geq 0$	$z_{klijs} = 1$ if a changeover from product i to product j takes place on machine k at stage l during microperiod s (0 otherwise)

Before presenting the MMSLSP model, we first discuss the time structure as well as indices, parameters, and data needed to formulate the model. The finite planning horizon H is divided into the equal length of $t = 1, 2, \ldots, T$ macroperiods. Moreover, each macroperiod is divided further into predefined number of microperiods with variable length at all production stages for each machine. The production of a single item allows in each microperiod using a proposed time structure (see Fig. 2) for all machines at all production stages.

Fig. 2. A proposed possible microperiod structure.

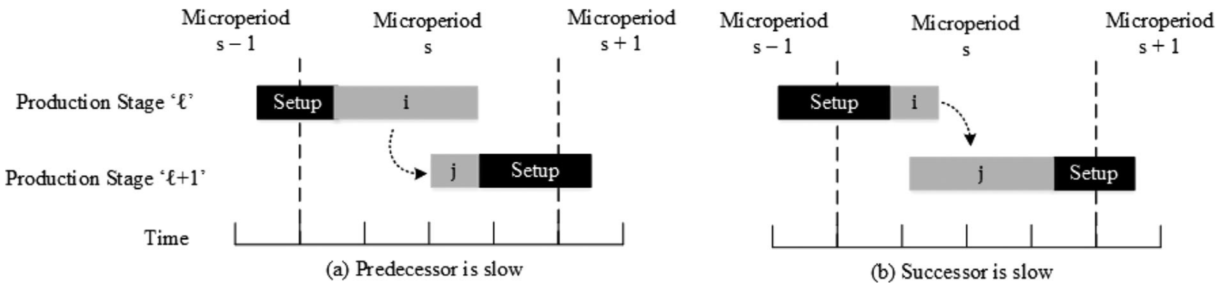

Fig. 3. The idea of setup splitting for production stages synchronization.

The length of each microperiod represents the total time of setup, production, and machine idle. The idea of setup splitting is applied in order to achieve high capacity utilization. Setup splitting means a share of the setup time can be implemented at the end of the sth period while the remaining portion at the beginning of the $(s+1)$th period (see, e.g., Haase, 1994; Drexl and Haase, 1995; Seeanner and Meyr, 2013). Figure 3 illustrates the idea of setup splitting for synchronizing the multi-stage production process.

By considering a finite planning horizon H, the physical products $i, j = 1, 2, \ldots, N$ has to be produced on machines $k = 1, 2, \ldots, K$ at production stage l. The coefficients a_{klj} represents the time required to produce one unit of product j on machine k at stage l, and c^p_{kljs} represents the per unit production cost of product j on machine k at stage l in microperiod s. Changeovers between products (i.e., i to j) on machine k at stage l incur the setup times and setup costs represented by st_{klij} and s_{klij}, respectively. For the production of the same product, holding machine k at stage l in a reserve mode again incur time-dependent standby (idle) cost c^b_{kl}. As the LBM model consists of non-overlapping macroperiods $t = 1, 2, \ldots, T$ work with a finite planning horizon H. Updated demand d_{jt}, inventory holding costs h_{jt}, and production capacity C_{klt} are represented per macroperiod t. Moreover, for each microperiod shown in Fig. 2, decision variables st^b_{klijs} and st^e_{klijs} represent the part of the setup time at the beginning and end of microperiod s on machine k at stage l due to a changeover between products (i.e., i to j), respectively. The value of Q_{kljs} represents the quantity of product j produced on machine k at stage l in microperiod s, which is used to compute the production time $a_{klj}Q_{kljs}$. Finally, sb_{kls} denotes the remaining part of microperiod s on machine k at production stage l.

Using the above notations, the proposed MMSLSP model can be described in Section 3.3.1.

3.3.1. MMSLSP model

The objective function (2) is to minimize, for each machine and microperiod, the total production cost, inventory holding cost, sequence-dependent setup cost, and idle-time-dependent cost. Note that the total production cost $\sum_{jt} c^p_j d_{jt}$ can be omitted from the model if production costs c^p_{kljs} are equal for all machines in respective microperiods ($c^p_{kljs} = c^p_j \quad \forall k, l, j, s \in S_t$), because it will be irrelevant for optimization.

$$Min \sum_{k \in K_l, l, j, s} c^p_{kljs} Q_{kljs} + \sum_{j,t} h_{jt} I_{jt} + \sum_{k \in K_l, l, i, j, s} s_{klij} z_{klijs} + \sum_{k \in K_l, l, s} c^b_{kls} sb_{kls}. \tag{2}$$

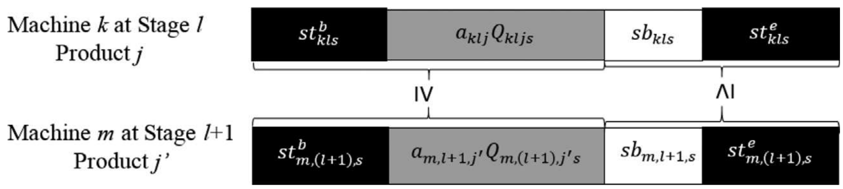

Fig. 4. The relationship between the time required by the successor product j on machine m at stage $l+1$ and time required by the predecessor product j machine k at stage l in sth microperiod.

In the multi-stage integrated LSS problem, the production stages must be synchronized at both macro- and microperiod levels to obtain a feasible production schedule. This issue is addressed through constraints (3) and (4) at the aggregate level with respect to product quantity and constraints (6) and (7) at microlevel for obtaining the appropriate length of microperiods. The standard lot-sizing and inventory balancing constraint (3) ensures the updated demands for final products (processed by the final production stage) that are met through current macroperiod total production and/or available initial inventory in this period without backorders allowed. The total production of each product in a macroperiod is obtained by adding together the production in all the microperiods belonging to that macroperiod ($s \in S_t$). For the material and work-in-process (WIP) inventory, balancing constraint (4) ensures that the total inflows must be equal to total outflows of a product for each macroperiod t.

$$I_{j,(t-1)} + \sum_{k \in K_L, s \in S_t} Q_{kLjs} = I_{it} + d_{jt} \quad \forall j \in \Lambda, t, j \in N_L, \tag{3}$$

$$I_{j,(t-1)} + \sum_{k \in K_L, s \in S_t} Q_{kLjs} = I_{it} + \sum_{m \in K_{l+1}} \sum_{i \in \Gamma(j), s \in S_t} p_{ji} Q_{m,(l+1),is} \quad \forall j \in N_l, t, l = 1, 2, \ldots, (L-1). \tag{4}$$

Constraint (5) enforces the regular production capacity restriction of machine k at production stage l and the sum of the setups and idle times may result in reducing the overall production capacity.

$$\sum_{j \in N_l, s \in S_t} a_{klj} Q_{kljs} + \sum_{(i,j) \in N_l, s \in S_t} st_{klij} z_{klijs} + \sum_{s \in S_t} sb_{kls} = C_{klt} \quad \forall k \in K_l, l, t. \tag{5}$$

The synchronization among multiple production stages at machine level with respect to the length of microperiod are ensured through constraints (6) and (7). Both constraints ensure that production of predecessor product j on machine k at stage l must be completed before the start of immediate successor product j on machine m at stage $l+1$ in each microperiod s. As shown in Fig. 4, constraint (6) makes sure that the total setup and idle times on production stage l are at most as long as total setup and idle time on subsequent production stage $l+1$. Constraint (7) enforces that the total setup and production times on production stage l are at least as long as the total setup and production times on production stage $l+1$. Moreover, both constraints are only active when the predecessor product j and successor product j are setups at any machine(s) on production stage l and $l+1$, respectively.

$$BigM(y_{kljs}-1)+\left(sb_{kls}+st^{e}_{kls}\right)\leq BigM\left(1-y_{m,(l+1),j's}\right)+\left(sb_{m,(l+1)s}+st^{e}_{m,(l+1),s}\right)$$
$$\forall j\in N_l,\ j\in\Omega(j'),\ j'\in N_{l+1},\ j'\in\Gamma(j),\ k\in K_l,\ m\in K_{(l+1)},\ s,\ l=1\ldots L-1, \tag{6}$$

$$BigM(1-y_{kljs})+\left(st^{b}_{kls}+a_{klj}Q_{kljs}\right)\geq BigM\left(y_{m,(l+1),j's}-1\right)$$
$$+\left(st^{b}_{m,(l+1),s}+a_{m,(l+1),j'}Q_{m,(l+1),j',s}\right)$$
$$\forall j\in N_l,\ j\in\Omega(j'),\ j'\in N_{l+1},\ j'\in\Gamma(j),\ k\in K_l,\ m\in K_{(l+1)},\ s,\ l=1\ldots L-1. \tag{7}$$

With constraint (8), the upper bound C_{klt}/a_{klj} on the production quantity is applied if the machine is setup for that product. In other words, production of the jth product can start only when the machine is setup for it. Furthermore, if the setup state variable is equal to 1 ($y_{kljs}=1$), then the length of microperiod s depends on capacity consumption of the production quantity Q_{kljs}, and zero-length microperiods are allowed. A minimum lot-size is needed (i.e., lower bound of the production quantity) to avoid setup state changes without product changeover through constraint (9) because the triangle inequality $st_{kliz}+st_{klzj}\geq st_{klij}$ of the setup times (or costs) do not always satisfy. Note that non-zero final inventories are possible due to minimum lot-sizes.

$$Q_{kljs}\leq\frac{C_{klt}}{a_{klj}}(y_{kljs})\qquad\forall l,k\in K_l,\ j\in N_l,\ s, \tag{8}$$

$$Q_{kljs}\leq M_{klj}(y_{kljs}-y_{klj,(s-1)})\qquad\forall l,k\in K_l,\ j\in N_l,\ s. \tag{9}$$

At all the production stages for microperiod s, each machine only setup for one product (i.e., a single setup state per microperiod) as indicated in constraint (10). Constraint (11) ensures that the changeover indicator variable z_{klijs} is equal to 1 only once in each microperiod.

$$\sum_{j\in N_l}y_{kljs}=1\qquad\forall l,k\in K_l,\ s, \tag{10}$$

$$\sum_{(i,j)\in N_l}z_{klijs}=1\qquad\forall l,k\in K_l,\ s\geq 2. \tag{11}$$

The idea of setup splitting is applied through constraint (12), ensuring that setup time fraction st^{b}_{kls}, st^{e}_{kls} are considered in an aggregate manner.

$$st^{e}_{kl,(s-1)}+st^{b}_{kls}=\sum_{(i,j)\in N_l}st_{klij}z_{klijs}\qquad\forall l,k\in K_l,\ s\geq 2. \tag{12}$$

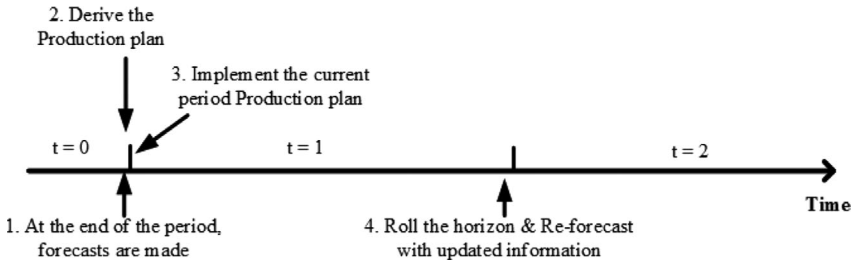

Fig. 5. The production planning sequence.

The link between machine setup state indicator variable y_{kljs} and product changeover indicator variable z_{klijs} is represented by constraint (13), which will perfectly trace the product i to product j changeovers during the planning horizon.

$$z_{klijs} \geq y_{kli,(s-1)} + y_{kljs} - 1 \qquad \forall l, k \in K_l, (i, j) \in N_l, s \geq 2. \tag{13}$$

4. Rolling horizon heuristics: Ideas and description

The integrated LSS problems are dynamic in nature, thus RH approach is usually used to solve such problems with an iterative procedure, where demand forecasts are dynamically updated over the planning horizon as time progresses and more information becomes available. On the other hand, if product demand is perfectly known or planning horizon is fixed, the RH approach is more suitable (Dellaert and Jeunet, 2003; Mohammadi et al., 2010a, 2010b). Figure 5 shows the general ideas for developing the production plan under RH approach.

Using the RH approach for solving the integrated LSS problem, only part of the planning horizon is represented as detailed schedules while the rest of the planning horizon is represented in an aggregate manner. Rolling horizon heuristics (RHHs) decompose the large-sized problem into a number of smaller subhorizon problems and then iteratively solve the subhorizon problems to determine the decision variables and carry over any unmet demand to the following subhorizon (Clark and Clark, 2000; Mercé and Fontan, 2003; Mohammadi et al., 2010a; Ramezanian et al., 2013). In principle, this approach significantly reduces the computational complexity and produces the feasible planning and scheduling solutions by substituting the binary constraints and variables with continuous constraints and variables for later periods of the planning horizon.

In the framework provided by Mercé and Fontan (2003), the RHH divides the finite planning horizon into three parts (initial, central and final) at each step of the iterative procedure. At an iteration t:

- FIX: The initial part consists of $(t-1)$ first periods. In this part, the decision variables are assigned the computed values from the previous iteration of the algorithm, either partially or completely, with respect to the chosen freezing strategy.
- OPTIMIZE: The second (central part) only contains tth period. The entire problem is considered in this section and the complete tth period model is solved (i.e., second part SS_t model).

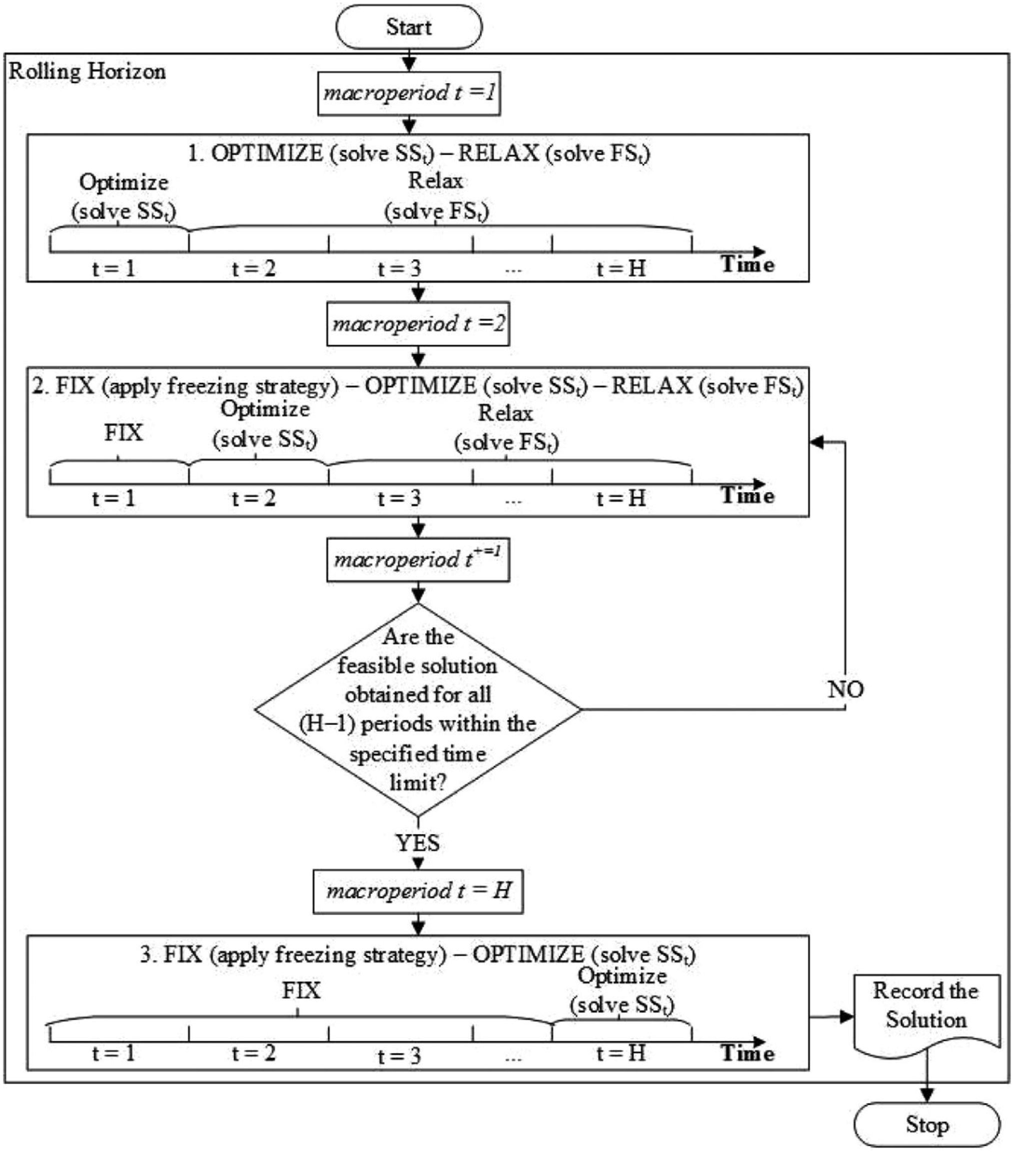

Fig. 6. The rolling horizon heuristic.

- RELAX: The final part (ending part) includes rest of periods, from the period $(t + 1)$ to H. In this section, preferred simplifying strategy is applied and the relaxed model is solved (i.e., final section FS_t model).

This iterative procedure terminates when the Hth iteration (i.e., last iteration) is completed. Figure 6 shows the iterative procedure of the RHH in which defined fix, optimize, and relax strategies are applied within the RH framework.

4.1. Heuristic algorithms

The computational time increases exponentially as the number of binary variables increases in the MIP model. Hence, faster solution approaches are needed to make appropriate decisions. Three heuristic algorithms are presented in this paper based on previously discussed iterative procedure.

Algorithm 1. Rolling horizon heuristic (RHH)

$t = 0$
while $(t = t + 1) \leq H$ **do**
  Solve the defined SS_t and FS_t mathematical models for the problem
  Freeze all the values of y_{kljs} obtained from SS_t
  if $t < H$ **then**
    Transfer the deviations from demand in SS_t to SS_{t+1}
  end if
end while

Algorithm 2. Rolling horizon heuristic (RHH)

$t = 0$
while $(t = t + 1) \leq H$ **do**
  Solve the defined SS_t and FS_t mathematical models for the problem
  Freeze all the values of y_{kljs} and Q_{kljs} obtained from SS_t
  if $t < H$ **then**
    Transfer the deviations from demand in SS_t to SS_{t+1}
  end if
end while

Algorithm 3. Rolling horizon heuristic (RHH)

$t = 0$
while $(t = t + 1) \leq H$ **do**
  Solve the defined SS_t and FS_t mathematical models for the problem
  Freeze all the values of y_{kljs} and z_{klijs} obtained from SS_t
  if $t < H$ **then**
    Transfer the deviations from demand in SS_t to SS_{t+1}
  end if
end while

The three algorithms are different regarding the freezing strategy chosen. All three algorithms use the "relax-and-fix" approach within the framework of RH (see Fig. 6) where the values of the variables is fixed with the chosen freezing strategy in the first $(t - 1)$ periods while the whole model is solved only for tth period. The binary and integer variables condition is relaxed for the period $(t + 1)$ to H (i.e., simplifying strategy).

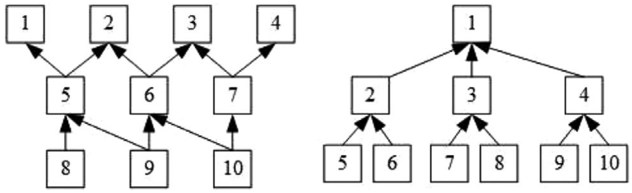

Fig. 7. General and assembly product structure problem Class B.

5. Computational experiments

The objective of this section is to test and compare the performance of three proposed heuristic algorithms (based on the relax-and-fix method with RH framework) through computational experiments on a set of benchmark problem instances. The three heuristics are implemented in IBM ILOG optimization programming language (OPL). Truncated MIP (TM) procedure is used as a benchmark to test the performance of the proposed heuristics. In TM procedure, the standard MIP solver (e.g., CPLEX, Gurobi) is run for a fixed amount of time and the best solution so far is recorded. The time limit is set to be 3600 seconds and the experiments are run on a 64-bit Window 7 PC with 2.3 GHz (two processors) Intel(R) Xeon(R) CPU E5-2650 v3 and 8 GB RAM.

To measure the performance of the heuristics, we calculate the gap (%) using the following formula:

$$Gap(\%) = \frac{[V(Heuristic) - V(Truncated\ MIP)]}{V(Truncated\ MIP)} \times 100, \tag{14}$$

where V (Heuristic) and V (Truncated MIP) represent the total cost.

5.1. The test instances

In order to compare the performance of the three heuristics with the Truncated MIP method, we use the instances of problem Classes B and D, with noncyclic resource allocation, developed and derived by Tempelmeier and Derstroff (1996). Class B contains small instances (10 items, three production stages, a planning horizon of four periods) while Class D includes large instances (40 items, six production stages, 16 periods). Later, Almeder et al. (2015) also used these instances for evaluating the performance of two proposed formulations. The complexity of these problems structure are illustrated in Figs 7 and 8. For more details of these problem instances, please see Tempelmeier and Derstroff (1996).

Note that in their test instances, the authors do not consider demand with information updating. However, we derive the demand forecast updates using MMFE from AR(1) model (as discussed in Section 3.2, and for derivation, see Appendix). To access the problem data, please see Rehman et al. (2019).

The problem Class B contains 180 small-sized test instances, while the problem Class D is made up of 180 large-sized test instances. The generated problem Classes B and D have the following properties:

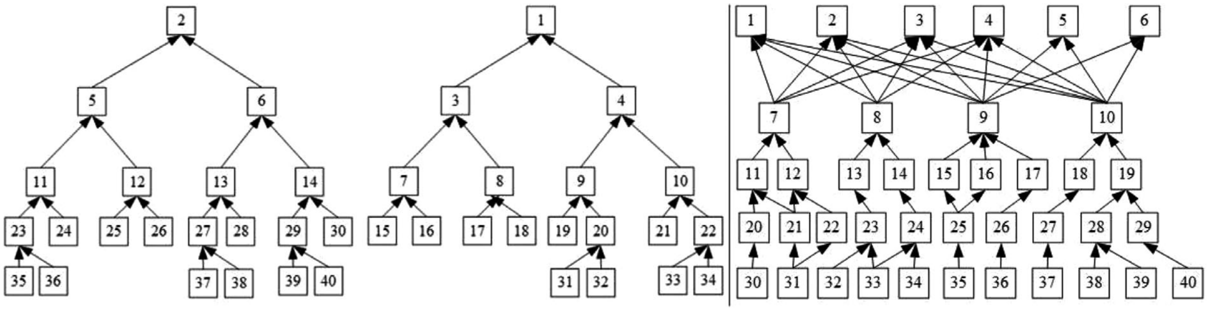

Fig. 8. Assembly and general product structure problem Class D.

- two product and operation structure (assembly (A) and general (G) product structure, and non-cyclic operation structure);
- five capacity profiles (i.e., C1, C2, ..., C5) with varying capacity utilization;
- two setup time profiles (i.e., S1, S2);
- nine time series of end products demand with updated information (with autocorrelation $\rho =$ 0.20, 0.60, 0.90 and coefficient of variation = 0.1, 0.4, 0.7).

For example, in the considered problem instances, when the demand of the end product(s) assumed that autocorrelation (ρ) is 0.90, the total forecast error variability over four time periods can be calculated as follows: $= Var(e_{(1,4)}) + Var(e_{(2,4)}) + Var(e_{(3,4)}) + Var(e_{(4,4)}) = \rho^6\sigma^2 + \rho^4\sigma^2 + \rho^2\sigma^2 + \sigma^2 = 0.5314\sigma^2 + 0.6561\sigma^2 + 0.8100\sigma^2 + \sigma^2 = 2.9975\sigma^2$.

Hence, the forecast error variability removed under updated information by advancing from time period t to $t+1$ is 17.72%, t to $t+2$ is 39.62%, t to $t+3$ is 66.64%, and t to $t+4$ is 100%.

Note that for both problem classes, we assume time between orders is equal to 1.

5.2. The results

As mentioned before, the motivation of this research is a practical problem faced by a supplier of automotive manufacturing firms. It is well known that the BOM in automobile manufacturing strictly follows the assembly processes (i.e., convergent) (Kreipl and Pinedo, 2004; Meyr, 2004b; Boysen et al., 2009). In the experiments, we evaluate the performance of the heuristics for both problem Classes B and D with assembly and general product structures, respectively. The purpose is to generalize the use of the heuristics in practices.

The standard MIP solver (i.e., CPLEX 12.6) is able to generate optimal solutions for problem Class B but unable to obtain the optimal solutions for problem Class D within the specified time limit (i.e., truncated after 3600 seconds) due to the complexity of those instances. In other words, CPLEX 12.6 produces the solutions with gap (%) 0 (i.e., an optimal solution) compared to the best integer solution for all the test instances from problem Class B. Here, the gap is defined as: $Integer\ gap(\%) = \frac{(best\ integer - best\ node)*objsen}{(abs(best\ integer)+e^{-10})} * 100$. For the test instances from problem Class D, the gap (%) varies from 0.09 to 6.18 and 8.65 to 38.68 compared to the best integer solution with assembly and general product structure, respectively.

As mentioned earlier, CPLEX 12.6 can provide optimal solutions for all the test instances from the problem Class B, but not for problem Class D (i.e., Truncated MIP solution). Therefore, to keep consistency for testing the performance of the proposed heuristics for problem Classes B and D, the gap is computed using Equation (14). Tables 2–5 compare the average and standard deviation of the computational times, the total costs as well as the gap (%) (i.e., the percentage difference of the total costs obtained from the heuristics versus the Truncated MIP model) for both assembly and general product structures in problem Classes B and D. All three proposed heuristics (i.e., Algorithms 1–3) are simple but effective as they can provide optimal solution for problem Class B with assembly and general product structure (see Tables 2 and 3). However, for general product structure, the performance of Heuristic 2 is not as good as for assembly product structure but the results of one-way analysis of variance (ANOVA) (see Table 6) test, for both the product structures, indicates that the average total cost produced by the four considered methods (i.e., Truncated MIP, Heuristic 1, Heuristic 2, and Heuristic 3) are equal.

The results (i.e., average and standard deviation of the computational times, the total costs as well as the gap (%)) for problem Class D using Truncated MIP and three heuristics, in assembly and general product structure, are summarized in Table 4 and 5, respectively. It can be observed from Table 4 that, for assembly product structure, Heuristics 1 and 3 produce the feasible solutions with a smaller gap (%) (i.e., on average, less than 4%) within six minutes (CPU time) compared to Heuristic 2, indicating the superiority of the Heuristics 1 and 3 over Heuristic 2 in this case. Figure 9 also shows that Heuristics 1 and 3 outperform Heuristic 2 and the total cost produced by Heuristics 1 and 3 is almost equal to that of Truncated MIP. In other words, the graph with respect to total cost indicates that the average total cost produced by Heuristics 1 and 3 and Truncated MIP is almost equal, but the average total cost obtained from Heuristic 2 is much higher compared to that of Heuristics 1 and 3 and Truncated MIP. Figure 9 also shows that Heuristic 2 is unable to produce a feasible solution for five instances out of 90.

Furthermore, one-way ANOVA tests (see Table 7) show that Truncated MIP and Heuristics 1–3 are not significantly equal.

The Tukey's honest significant difference (HSD) test for multiple comparisons (see Table 8) is applied to check which method(s) are significantly different or equal. The results show that Truncated MIP and Heuristics 1 and 3 are producing equal, on average, total cost; however, they are significantly better than Heuristic 2.

Table 5 shows the results for general product structure from problem Class D, indicating that Heuristics 1 and 3 perform better than Heuristic 2. It can be observed from Table 5 that Heuristics 1 and 3 produce the feasible solutions with, on average, less than 3% within seven minutes (CPU time) compared to Heuristic 2, indicating the superiority of the Heuristics 1 and 3 over Heuristic 2. Figure 10 also indicates that performance of Heuristics 1 and 3 compared to Heuristic 2 is quite well as both of them provide smaller average total cost, which is much closer to the Truncated MIP solution value. It also shows that Heuristics 2 and 3 was unable to provide the feasible solution 79 and 2 times, respectively.

Furthermore, one-way ANOVA test results (F value $= 0.011$, df $= 2$, p-value $= 0.989$) shows that Truncated MIP and Heuristics 1 and 3 are equal, showing that these three methods are producing same average total cost. Due to a smaller number of solution values produced by Heuristic 2, we do not consider it in this ANOVA test.

Table 2
MMSLSP model (problem set (B): assembly product structure): comparisons of the heuristics with respect to average and standard deviation of the CPU solution times, total costs, and gap (%)

Test instances setting	Truncated MIP	Heuristic 1			Heuristic 2			Heuristic 3		
	Total cost	CPU time (seconds)	Total cost	Gap (%)	CPU time (seconds)	Total cost	Gap (%)	CPU time (seconds)	Total cost	Gap (%)
S1C1	76,614.4	1.00	76,813.1	1.16	0.85	76,612.8	0.03	1.26	76,612.75	0.03
	(129,662.2)	(0.45)	(129,563.4)	(2.12)	(0.15)	(129,648.8)	(0.07)	(0.63)	(129,648.83)	(0.07)
S1C2	72,661.54	0.84	72,859.73	1.20	0.80	72,659.31	0.00	1.17	72,659.31	0.00
	(125,412.62)	(0.09)	(125,325.3)	(2.26)	(0.08)	(125,409.28)	(0.00)	(0.59)	(125,409.28)	(0.00)
S1C3	67,347.19	0.80	67,665.48	1.29	0.78	67,346.08	0.00	0.91	67,346.1	0.00
	(117,988.32)	(0.14)	(117,899.03)	(2.43)	(0.07)	(117,985.27)	(0.00)	(0.13)	(117,985.3)	(0.00)
S1C4	68,944.93	0.82	69,261.67	1.36	0.74	68,943.27	0.00	0.94	68,943.3	0.00
	(118,803.36)	(0.10)	(118,715.74)	(2.20)	(0.08)	(118,798.81)	(0.00)	(0.16)	(118,798.8)	(0.00)
S1C5	75,363.53	1.09	75,561.65	1.21	0.86	75,361.3	0.00	0.92	75,361.31	0.00
	(129,303.91)	(0.35)	(129,211.87)	(2.29)	(0.38)	(129,297.9)	(0.00)	(0.15)	(129,297.89)	(0.00)
S2C1	75,996.64	0.85	76,207.00	1.16	0.70	76,004.97	0.03	0.93	76,006.64	0.03
	(128,825.86)	(0.11)	(128,753.68)	(2.11)	(0.05)	(128,838.21)	(0.07)	(0.10)	(128,838.57)	(0.07)
S2C2	72,019.15	0.75	72,226.71	1.20	0.68	72,026.38	0.01	0.89	72,026.38	0.01
	(124,378.49)	(0.06)	(124,308.36)	(2.23)	(0.05)	(124,391.86)	(0.03)	(0.12)	(124,391.86)	(0.03)
S2C3	66,868.30	0.78	67,239.44	1.55	0.69	66,871.64	0.00	0.90	66,872.19	0.00
	(116,700.71)	(0.09)	(116,623.57)	(2.41)	(0.05)	(116,709.84)	(0.00)	(0.11)	(116,709.82)	(0.00)
S2C4	68,417.39	0.86	68,791.85	1.44	0.68	68,423.50	0.01	0.88	68,424.62	0.01
	(117,448.76)	(0.16)	(117,378.88)	(2.20)	(0.06)	(117,461.87)	(0.03)	(0.12)	(117,462.00)	(0.03)
S2C5	74,803.65	0.78	75,009.49	1.22	0.67	74,809.20	0.01	1.07	74,809.20	0.01
	(128,501.72)	(0.08)	(128,425.04)	(2.26)	(0.08)	(128,510.56)	(0.03)	(0.47)	(128,510.56)	(0.03)

Table 3
MMSLSP model (problem set (B): general product structure): comparisons of the heuristics with respect to average and standard deviation of the CPU solution times, total costs, and gap (%)

Test instances setting	Truncated MIP	Heuristic 1			Heuristic 2			Heuristic 3		
	Total cost	CPU time (seconds)	Total cost	Gap (%)	CPU time (seconds)	Total cost	Gap (%)	CPU time (seconds)	Total cost	Gap (%)
S1C1	51,028.6	1.00	51,026.8	0.00	1.14	51,418.3	4.16	0.96	51,026.8	0.00
	(80,360.3)	(0.30)	(80,355.2)	(0.00)	(0.30)	(80,172.8)	(3.06)	(0.07)	(80,355.2)	(0.00)
S1C2	48,866.6	1.00	48,897.6	0.25	1.54	49,277.2	4.51	0.99	48,897.7	0.25
	(77,738.8)	(0.01)	(77,725.7)	(0.38)	(1.36)	(77,553.6)	(3.35)	(0.11)	(77,725.9)	(0.38)
S1C3	45,505.7	0.94	45,609.6	0.29	1.07	46,089.1	4.76	0.95	45,609.6	0.29
	(73,606.5)	(0.13)	(73,627.6)	(0.36)	(0.11)	(73,495.5)	(3.08)	(0.07)	(73,627.7)	(0.36)
S1C4	46,830.6	0.88	46,923.2	0.14	0.94	47,402.8	4.41	0.97	46,923.2	0.14
	(74,531.0)	(0.09)	(74,560.0)	(0.37)	(0.09)	(74,429.1)	(2.86)	(0.08)	(74,560.0)	(0.37)
S1C5	49,996.5	0.98	50,008.4	0.15	1.00	50,399.9	4.55	1.02	50,008.3	0.15
	(79,903.5)	(0.22)	(79,897.9)	(0.16)	(0.09)	(79,715.7)	(3.33)	(0.06)	(79,897.8)	(0.16)
S2C1	50,958.7	0.95	50,960.0	0.00	0.96	51,391.0	4.11	0.96	50,959.9	0.00
	(79,780.8)	(0.08)	(79,780.9)	(0.01)	(0.15)	(79,598.7)	(2.88)	(0.07)	(79,780.8)	(0.01)
S2C2	48,741.2	1.13	48,762.3	0.12	0.97	49,144.0	4.22	0.96	48,762.3	0.12
	(77,125.0)	(0.30)	(77,116.5)	(0.33)	(0.15)	(76,952.0)	(3.06)	(0.07)	(77,116.6)	(0.33)
S2C3	45,351.1	1.01	45,414.6	0.32	1.00	45,949.8	4.96	1.11	45,414.6	0.32
	(73,140.4)	(0.08)	(73,144.8)	(0.29)	(0.18)	(73,012.2)	(3.12)	(0.15)	(73,144.9)	(0.29)
S2C4	46,675.9	0.90	46,719.5	0.06	1.03	47,250.5	4.43	1.02	46,719.4	0.06
	(73,982.9)	(0.08)	(73,997.1)	(0.17)	(0.15)	(73,868.2)	(2.76)	(0.12)	(73,996.9)	(0.17)
S2C5	49,895.9	0.92	49,913.8	0.25	0.99	50,348.9	4.71	0.98	49,913.7	0.25
	(79,394.8)	(0.11)	(79,385.4)	(0.28)	(0.13)	(79,201.2)	(3.34)	(0.07)	(79,385.3)	(0.28)

Table 4

MMSLSP model (problem set (D): assembly product structure): comparisons of the heuristics with respect to mean and standard deviation of the CPU solution times, total costs, and gap (%)

Test instances setting	Truncated MIP CPU time (seconds)	Truncated MIP Total cost	Heuristic 1 CPU time (seconds)	Heuristic 1 Total cost	Heuristic 1 Gap (%)	Heuristic 2 CPU time (seconds)	Heuristic 2 Total cost	Heuristic 2 Gap (%)	Heuristic 3 CPU time (seconds)	Heuristic 3 Total cost	Heuristic 3 Gap (%)
S1C1	>3600	135,936.0	181.5	137,545.2	2.03	172.23	200,787.7	98.59	295.02	137,347.2	1.89
		(188,053.3)	(7.5)	(188,361.9)	(0.72)	(11.88)	(179,233.8)	(42.10)	(17.95)	(188,093.4)	(0.79)
S1C2	>3600	116,744.5	170.2	118,548.9	2.20	165.06	178,946.6	108.23	278.74	118,487.3	2.23
		(152,788.4)	(7.11)	(153,560.0)	(0.74)	(4.86)	(153,057.3)	(53.99)	(8.89)	(153,575.9)	(0.86)
S1C3	>3600	97,203.6	174.0	99,178.5	2.58	170.35	180,626.2	135.93	275.10	99,358.0	3.04
		(115,496.7)	(8.31)	(116,364.0)	(0.98)	(15.03)	(127,094.5)	(45.72)	(6.63)	(116,186.9)	(1.38)
S1C4	>3600	114,876.1	173.0	116,773.3	2.25	174.24	186,711.5	104.41	295.18	116,704.1	2.15
		(137,876.4)	(7.7)	(138,655.1)	(0.65)	(10.92)	(140,153.4)	(38.13)	(11.47)	(138,740.4)	(0.82)
S1C5	>3600	118,623.9	171.90	120,142.8	2.14	173.49	195,630.0	130.69	286.58	120,249.5	2.43
		(168,379.4)	(5.76)	(168,850.7)	(0.88)	(16.05)	(165,914.5)	(52.59)	(14.39)	(168,688.4)	(1.13)
S2C1	>3600	167,641.9	171.6	168,618.5	1.06	163.27	230,613.4	59.76	291.09	168,653.0	1.11
		(210,628.6)	(6.40)	(210,569.2)	(0.55)	(11.30)	(206,004.5)	(29.77)	(18.02)	(210,553.4)	(0.63)
S2C2	>3600	140,538.4	162.5	141,450.1	1.14	164.41	210,026.5	73.77	293.76	141,489.7	1.20
		(171,612.3)	(8.20)	(171,564.0)	(0.47)	(13.94)	(173,313.1)	(31.46)	(17.05)	(171,543.4)	(0.50)
S2C3	>3600	114,619.7	172.4	116,148.6	1.80	161.57	193,821.5	95.11	291.49	116,187.6	1.80
		(131,849.5)	(5.40)	(132,387.7)	(0.52)	(6.91)	(143,754.7)	(20.57)	(20.57)	(132,659.0)	(0.43)
S2C4	>3600	141,561.5	162.7	142,541.3	1.08	174.36	211,440.2	68.04	301.39	142,627.5	1.17
		(158,144.9)	(5.99)	(158,048.7)	(0.40)	(14.02)	(162,313.1)	(27.39)	(24.84)	(158,035.4)	(0.48)
S2C5	>3600	139,509.3	167.6	140,549.7	1.54	199.66	212,356.1	89.12	304.34	140,563.2	1.59
		(189,469.3)	(12.5)	(189,254.3)	(0.69)	(54.53)	(187,518.0)	(41.32)	(19.36)	(189,241.2)	(0.74)

Table 5
MMSLSP model (problem set (D): general product structure): comparisons of the heuristics with respect to mean and standard deviation of the CPU solution times, total costs, and gap (%)

Test instances setting	Truncated MIP		Heuristic 1			Heuristic 2			Heuristic 3		
	CPU time (seconds)	Total cost	CPU time (seconds)	Total cost	Gap (%)	CPU time (seconds)	Total cost	Gap (%)	CPU time (seconds)	Total cost	Gap (%)
S1C1	>3600	3,528,359.0 (3,798,442.6)	209.68 (7.24)	3,576,633.8 (3,815,642.2)	1.98 (0.69)		–		366.89 (16.96)	3,574,930.5 (3,811,880.2)	1.96 (0.72)
S1C2	>3600	3,411,851.8 (3,767,795.9)	215.3 (8.40)	3,460,266.3 (3,781,300.6)	2.13 (0.73)		–		366.91 (19.95)	3,460,755.3 (3,782,516.8)	2.17 (0.80)
S1C3	>3600	3,237,444.7 (3,725,965.5)	221.5 (7.10)	3,280,257.4 (3,746,613.2)	1.91 (0.61)	204.22 (7.58)	2,382,424.5 (72,796.0)	63.15 (1.10)	363.25 (10.51)	3,280,030.5 (3,743,824.3)	1.94 (0.65)
S1C4	>3600	3,442,778.3 (3,767,798.4)	222.4 (4.50)	3,491,125.8 (3,781,876.9)	2.11 (0.77)		–		359.61 (16.73)	3,491,306.5 (3,781,765.0)	2.12 (0.80)
S1C5	>3600	3,360,791.5 (3,770,765.5)	231.9 (9.60)	3,410,342.5 (3,783,992.5)	2.30 (0.90)	197.15 (2.16)	2,418,800.7 (88,689.2)	52.52 (1.06)	361.43 (19.14)	3,408,951.7 (3,783,123.2)	2.22 (0.82)
S2C1	>3600	3,586,337.2 (3,812,031.1)	204.9 (3.10)	3,622,305.6 (3,831,665.6)	1.29 (0.31)		–		361.93 (17.70)	3,622,913.1 (3,834,570.1)	1.27 (0.26)
S2C2	>3600	3,478,763.9 (3,789,971.4)	205.5 (4.80)	3,527,309.7 (3,802,625.9)	2.11 (0.78)		–		358.42 (16.97)	3,527,770.6 (3,804,996.3)	2.07 (0.71)
S2C3	>3600	3,323,816.0 (3,750,078.9)	216.1 (8.00)	3,371,252.3 (3,769,350.3)	2.09 (0.70)		–		366.32 (9.97)	3,369,551.4 (3,771,768.3)	1.96 (0.63)
S2C4	>3600	3,498,654.5 (3789958.9)	217.3 (7.90)	3,546,197.3 (3804854.3)	2.01 (0.73)		–		366.48 (32.93)	3,774,290.1 (3,974,327.9)	1.92 (0.71)
S2C5	>3600	3,445,099.8 (3,791,578.3)	216.6 (5.10)	3,492,312.7 (3,803,671.4)	2.10 (0.83)		–		411.49 (89.01)	3,720,922.9 (3,979,390.2)	1.93 (0.74)

"–": Algorithm is unable to obtain the feasible solution.

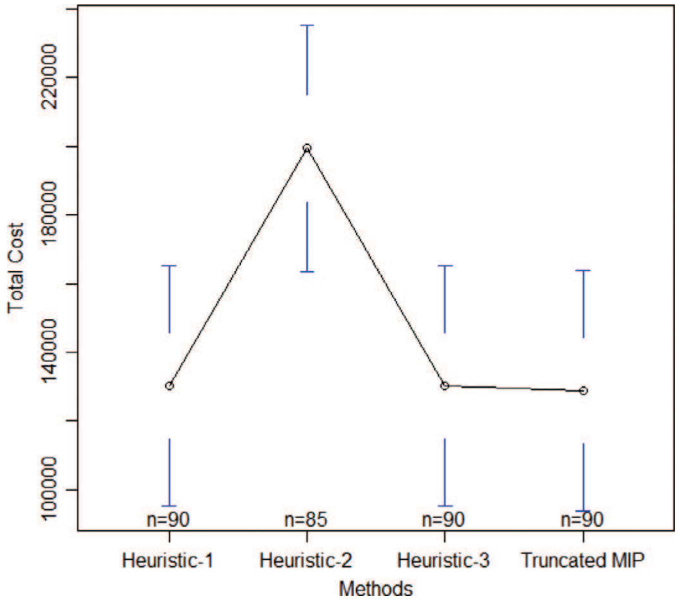

Fig. 9. Problem Class D—assembly: mean plot with 95% confidence interval.

Table 6
One -way ANOVA: problem set (B)

Product structure	df	Test static	*p*-Value
Assembly	(3, 356)	0.000	1.000
General	(3, 356)	0.001	1.000

Table 7
One-way ANOVA: problem set (D)

Product structure	df	Test static	*p*-value
Assembly	(3, 351)	3.767	0.011

Table 8
Tukey's HSD test: problem set (D): assembly product structure

Methods comparison	Adjusted *p*-value	Methods comparison	Adjusted *p*-value
Heuristic 2 – Heuristic 1	0.0321	Heuristic 3 – Heuristic 1	1.0000
Truncated MIP – Heuristic 1	0.9999	Heuristic 3 – Heuristic 2	0.0321
Truncated MIP – Heuristic 2	0.0274	Truncated MIP – Heuristic 3	0.9999

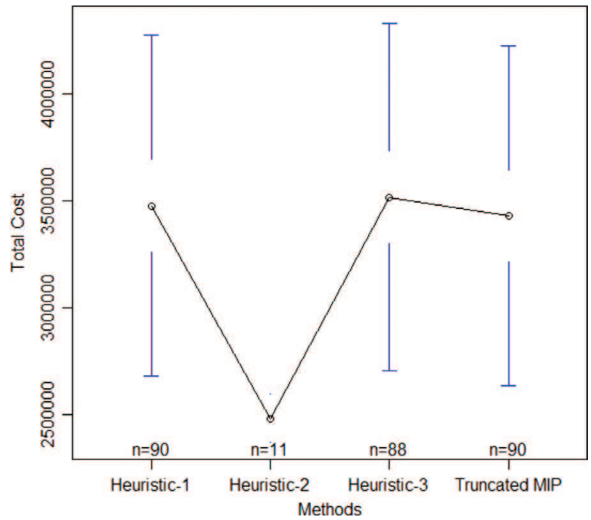

Fig. 10. Problem Class D—general: mean plot with 95% confidence interval.

6. Conclusions and future research

In this paper, we have studied the multi-product and multi-stage integrated LSS in a flexible flow shop with demand information updating. The evolving demand over time is one of the most important features in today's manufacturing industry. MMFE is employed to model such evolving demands. In order to tackle the LSS problem with the characteristics such as backorders being not allowed, limited capacity and sequence-dependent setup times and costs, an efficient MIP model is developed by combining the advantages of micro- and macroperiod models. Since the demand is evolved over time as new information arrives, the production plan must be revised and implemented on RH basis. Three RHH based on the idea of the "relax-and-fix" strategy are proposed. In the computational experiments, for the problem Class B, Heuristics 1 and 3 produce optimal solutions for both types of BOM structures (i.e., assembly and general structures), and both produce better results in terms of total costs; furthermore, the performance of Heuristic 2 is not as good in general as in assembly structure. For the problem Class D, only Heuristic 1 is able to produce feasible solutions for all the test instances. Thus, Heuristic 1 seems the better choice for both the problem classes. For future research, it is necessary to develop better mathematical models incorporating practical manufacturing features such as limited WIP inventory, keeping certain inventory level after the planning horizon (i.e., a certain level of safety stock), thus advanced solution approaches must be developed for solving such optimization problems.

Acknowledgments

The authors thank the editor and the anonymous referees for their valuable comments and suggestions on this paper. This work is partially supported by National Natural Science Foundation of China (71125003, 71421002) and Program of Shanghai Subject Chief Scientist (16XD1401700). The authors are listed alphabetically, and for correspondence, please contact Guohua Wan.

References

Alem, D., Curcio, E., Amorim, P., Almada-Lobo, B., 2018. A computational study of the general lot-sizing and scheduling model under demand uncertainty via robust and stochastic approaches. *Computers and Operations Research* 90, 125–141.

Ali, M.M., Boylan, J.E., Syntetos, A.A., 2012. Forecast errors and inventory performance under forecast information sharing. *International Journal of Forecasting* 28, 4, 830–841.

Almeder, C., Almada-Lobo, B., 2011. Synchronisation of scarce resources for a parallel machine lot-sizing problem. *International Journal of Production Research* 49, 24, 7315–7335.

Almeder, C., Klabjan, D., Traxler, R., Almada-Lobo, B., 2015. Lead time considerations for the multi-level capacitated lot-sizing problem. *European Journal of Operational Research* 241, 3, 727–738.

Altug, M.S., Muharremoglu, A., 2011. Inventory management with advance supply information. *International Journal of Production Economics* 129, 2, 302–313.

Araujo, S.A., Clark, A., 2013. A priori reformulations for joint rolling-horizon scheduling of materials processing and lot-sizing problem. *Computers and Industrial Engineering* 65, 4, 577–585.

Armellini, D., Borzone, P., Ceschia, S., Gaspero, L.D., Schaerf, A., 2020. Modeling and solving the steel-making and casting scheduling problem. *International Transactions in Operational Research* 27, 1, 57–90.

Belo-Filho, M.A., Toledo, F.M., Almada-Lobo, B., 2014. Models for capacitated lot-sizing problem with backlogging, setup carryover and crossover. *Journal of the Operational Research Society* 65, 11, 1735–1747.

Beraldi, P., Ghiani, G., Grieco, A., Guerriero, E., 2008. Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs. *Computers and Operations Research* 35, 11, 3644–3656.

Boysen, N., Fliedner, M., Scholl, A., 2009. Production planning of mixed-model assembly lines: overview and extensions. *Production Planning and Control: The Management of Operations* 20, 5, 455–471.

Camargo, V.C.B., Toledo, F.M.B., Almada-Lobo, B., 2012. Three time-based scale formulations for the two-stage lot sizing and scheduling in process industries. *Journal of the Operational Research Society* 63, 11, 1613–1630.

Chang, P.T., Chang, C.H., Chen, C.T., 2004. An extension of the multi-machine multi-level proportional lot sizing and scheduling model for product-life-cycle demand. *International Journal of Operations Research* 1, 1, 11–22.

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

Clark, A.R., 2003. Optimization approximations for capacity constrained material requirements planning. *International Journal of Production Economics* 84, 2, 115–131.

Clark, A.R., Clark, S.J., 2000. Rolling-horizon lot-sizing when set-up times are sequence-dependent. *International Journal of Production Research* 38, 10, 2287–2307.

Copil, K., Wörbelauer, M., Meyr, H., Tempelmeier, H., 2017. Simultaneous lotsizing and scheduling problems: a classification and review of models. *OR Spectrum* 39, 1, 1–64.

Curcio, E., Amorim, P., Zhang, Q., Almada-Lobo, B., 2018. Adaptation and approximate strategies for solving the lot-sizing and scheduling problem under multistage demand uncertainty. *International Journal of Production Economics* 202, 81–96.

Dauzère-Pérès, S., Lasserre, J.B., 2002. On the importance of sequencing decisions in production planning and scheduling. *International Transactions in Operational Research* 9, 6, 779–793.

Dellaert, N., Jeunet, J., 2003. Controlling multi-level production in a rolling-schedule environment. *International Journal of Production Economics* 85, 1, 113–121.

Drexl, A., Haase, K., 1995. Proportional lot-sizing and scheduling. *International Journal of Production Economics* 40, 73–87.

Drexl, A., Kimms, A., 1997. Lot sizing and scheduling: survey and extensions. *European Journal of Operational Research* 99, 2, 221–235.

Fandel, G., Stammen-Hegene, C., 2006. Simultaneous lot sizing and scheduling for multi-product multi-level production. *International Journal of Production Economics* 104, 2, 308–316.

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

Fiorotto, D.J., Jans, R., de Araujo, S.A., 2017. An analysis of formulations for the capacitated lot sizing problem with setup crossover. *Computers and Industrial Engineering* 106, 338–350.
Fleischmann, B., 1990. The discrete lot-sizing and scheduling problem. *European Journal of Operational Research* 44, 3, 337–348.
Fleischmann, B., Meyr, H., 1997. The general lot-sizing and scheduling problem. *OR Spektrum* 19, 1, 11–21.
Graves, S.C., Meal, H.C., Dasu, S., Qui, Y., 1986. *Two-Stage Production Planning in a Dynamic Environment*. Springer, Berlin, Germany.
Günther, H.O., 2014. The block planning approach for continuous time-based dynamic lot sizing and scheduling. *Business Research* 7, 1, 51–76.
Haase, K., 1994. *Lotsizing and Scheduling for Production Planning*. Springer-Verlag, Berlin, Germany.
Haase, K., 1996. Capacitated lot-sizing with sequence dependent setup costs. *OR Spektrum* 18, 1, 51–59.
Hausman, W.H., 1969. Sequential decision problems: a model to exploit existing forecasters. *Management Science* 16, 2, B93–B111.
Heath, D.C.D., Jackson, P.L.P., 1994. Modeling the evolution of demand forecasts with application to safety stock analysis in production/distribution systems. *IIE Transactions* 26, 3, 17–30.
Hu, Z., Hu, G., 2016. A two-stage stochastic programming model for lot-sizing and scheduling under uncertainty. *International Journal of Production Economics* 180, 198–207.
Iida, T., Zipkin, P.H., 2006. Approximate solutions of a dynamic forecast-inventory model. *Manufacturing & Service Operations Management* 8, 4, 407–425.
Kaczmarczyk, W., 2013. Modelling set-up times overlapping two periods in the proportional lot-sizing problem with identical parallel machines. *Decision Making in Manufacturing and Services* 7, 1–2, 43–50.
Karmarkar, U.S., Schrage, L., 1985. The deterministic dynamic product cycling problem. *Operations Research* 33, 2, 326–345.
Kimms, A., 1996. Multi-level, single-machine lot sizing and scheduling (with initial inventory). *European Journal of Operational Research* 89, 1, 86–99.
Kimms, A., 1999. A genetic algorithm for multi-level, multi-machine lot sizing and scheduling. *Computers and Operations Research* 26, 8, 829–848.
Kreipl, S., Pinedo, M., 2004. Planning and scheduling in supply chains: an overview of issues in practice. *Production and Operations Management* 13, 1, 77–92.
Lang, J.C., 2010. *Multi-Level Lot-Sizing Models with Flexible Bills-of-Materials*. Springer-Verlag, Berlin, Germany.
Lang, J.C., Shen, Z.J.M., 2011. Fix-and-optimize heuristics for capacitated lot-sizing with sequence-dependent setups and substitutions. *European Journal of Operational Research* 214, 3, 595–605.
Lee, H.L., So, K.C., Tang, C.S., 2000. The value of information sharing in a two-level supply chain. *Management Science* 46, 5, 626–643.
Mahdieh, M., Clark, A., Bijari, M., 2018. A novel flexible model for lot sizing and scheduling with non-triangular, period overlapping and carryover setups in different machine configurations. *Flexible Services and Manufacturing Journal* 30, 4, 884–923.
Méndez, C.A., Cerdá, J., Grossmann, I.E., Harjunkoski, I., Fahl, M., 2006. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering* 30, 6–7, 913–946.
Mercé, C., Fontan, G., 2003. MIP-based heuristics for capacitated lotsizing problems. *International Journal of Production Economics* 85, 1, 97–111.
Meyr, H., 2004a. *Simultane Losgrößen- und Reihenfolgeplanung bei mehrstufiger kontinuierlicher Fertigung*. Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL).
Meyr, H., 2004b. Supply chain planning in the German automotive industry. *OR Spectrum* 26, 4, 447–470.
Milner, J.M., Kouvelis, P., 2005. Order quantity and timing flexibility in supply chains: the role of demand characteristics. *Management Science* 51, 6, 970–985.
Miyaoka, J., Hausman, W., 2004. How a base stock policy using "stale" forecasts provides supply chain benefits. *Manufacturing & Service Operations Management* 6, 2, 149–162.
Mohammadi, M., 2010. Integrating lotsizing, loading, and scheduling decisions in flexible flow shops. *International Journal of Advanced Manufacturing Technology* 50, 9–12, 1165–1174.

Mohammadi, M., Fatemi Ghomi, S.M., Karimi, B., Torabi, S.A., 2010a. Rolling-horizon and fix-and-relax heuristics for the multi-product multi-level capacitated lotsizing problem with sequence-dependent setups. *Journal of Intelligent Manufacturing* 21, 4, 501–510.

Mohammadi, M., Fatemi Ghomi, S.M.T., Karimi, B., Torabi, S.A., 2010b. MIP-based heuristics for lotsizing in capacitated pure flow shop with sequence-dependent setups. *International Journal of Production Research* 48, 10, 2957–2973.

Mohammadi, M., Jafari, N., 2011. A new mathematical model for integrating lot sizing, loading, and scheduling decisions in flexible flow shops. *International Journal of Advanced Manufacturing Technology* 55, 5–8, 709–721.

Norouzi, A., Uzsoy, R., 2014. Modeling the evolution of dependency between demands, with application to inventory planning. *IIE Transactions* 46, 1, 55–66.

Quadt, D., Kuhn, H., 2005. Conceptual framework for lot-sizing and scheduling of flexible flow lines. *International Journal of Production Research* 43, 11, 2291–2308.

Quadt, D., Kuhn, H., 2008. Capacitated lot-sizing with extensions: a review. *4OR* 6, 1, 61–83.

Ramezanian, R., Saidi-Mehrabad, M., 2013. Hybrid simulated annealing and mip-based heuristics for stochastic lot-sizing and scheduling problem in capacitated multi-stage production system. *Applied Mathematical Modelling* 37, 7, 5134–5147.

Ramezanian, R., Saidi-Mehrabad, M., Fattahi, P., 2013. MIP formulation and heuristics for multi-stage capacitated lot-sizing and scheduling problem with availability constraints. *Journal of Manufacturing Systems* 32, 2, 392–401.

Ramya, R., Rajendran, C., Ziegler, H., 2016. Capacitated lot-sizing problem with production carry-over and set-up splitting: mathematical models. *International Journal of Production Research* 54, 8, 2332–2344.

Rehman, H.U., Wan, G., Zhan, Y., 2019. Multi-level multi-stage lot-sizing and scheduling in the flexible flow shop with demand information updating, Mendeley Data, v2. https://doi.org/10.17632/t5wd3p669p.2.

Sahling, F., Buschkuhl, L., Tempelmeier, H., Helber, S., 2009. Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. *Computers & Operations Research* 36, 2016, 2546–2553.

Seeanner, F., Almada-Lobo, B., Meyr, H., 2013. Combining the principles of variable neighborhood decomposition search and the fix-and-optimize heuristic to solve multi-level lot-sizing and scheduling problems. *Computers and Operations Research* 40, 1, 303–317.

Seeanner, F., Meyr, H., 2013. Multi-stage simultaneous lot-sizing and scheduling for flow line production. *OR Spectrum* 35, 1, 33–73.

Sethi, S.P., Yan, H., Zhang, H., 2005. *Inventory and Supply Chain Management with Forecast Updates*. Springer, Berlin, Germany.

Stadtler, H., 2005. Supply chain management and advanced planning: basics, overview and challenges. *European Journal of Operational Research* 163, 3, 575–588.

Stadtler, H., 2011. Multi-level single machine lot-sizing and scheduling with zero lead times. *European Journal of Operational Research* 209, 3, 241–252.

Tempelmeier, H., Derstroff, M., 1996. A Lagrangean-based heuristic for dynamic multilevel multiitem constrained lotsizing with setup times. *Management Science* 42, 5, 738–757.

Wolsey, L.A., 2002. Solving multi-item LSP with an MIP solver using classification and reformulation. *Management Science* 48, 12, 1587–1602.

Xiao, J., Zhang, C., Zheng, L., Gupta, J.N., 2013. MIP-based fix-and-optimise algorithms for the parallel machine capacitated lot-sizing and scheduling problem. *International Journal of Production Research* 51, 16, 5011–5028.

Zhu, X., Wilhelm, W.E., 2006. Scheduling and lot sizing with sequence-dependent setup: a literature review. *IIE Transactions* 38, 11, 987–1007.

Appendix

We assume the initial forecast values and obtained the variance–covariance (VCV) matrix of the forecast updates as follows:

When moving from tth to $(t+1)$th period: $d_{(t+1)} = \mu(1-\rho) + \rho d_t + e_{(t+1)} = d_{(t,t+1))} + e_{(t+1)}$; from $(t+1)$th to $(t+2)$th period: $d_{(t+2)} = \mu(1-\rho) + \rho d_(t+1) = \mu(1-\rho) + \rho(d_{t,(t+1)} + e_{(t+1)}) = d_{(t,t+2)} + \rho e_{(t+1)}$ and from $(t+2)$th to $(t+3)$th period: $d_{(t+3)} = \mu(1-\rho) + \rho d_(t+2) = \mu(1-\rho) + \rho d_{(t,t+2))} + \rho^2 e_{(t+1)} = d_{(t,t+3)} + \rho^2 e_{(t+1)}$.

From the above, we can observe a pattern:

$$d_{j,(t+v)} = f_{t,v} = \mu(1-\rho) + d_{j,(t,t+v-1)} + \rho^{(v-1)} e_{(t+1)} \quad \forall v \geq 1, t \leq v. \tag{A1}$$

The new information $I_{(t+v)}$ for each end product j is precisely $\rho^{(v-1)} e_{(t+1)}$, and it is used to update the demand forecast in all the future periods. The VCV matrix for MMFE is $\Sigma = (\sigma_{uv})$, where $\sigma_{uv} = E[e_{(t+1,t+u)}, e_{(t+1,t+v)}] = E[\rho^{(u+v)}(e_{(t+1)})^2] = \rho^{(u+v)}\sigma^2$, where $u, v = 0, 1, 2, \ldots, H$.

So, if the forecast horizon is H, then the VCV matrix is of $(H+1) \times (H+1)$.

$$\Sigma = \begin{bmatrix} \sigma^2 & \rho\sigma^2 & \rho^2\sigma^2 & \ldots & \rho^H\sigma^2 \\ \rho\sigma^2 & \rho^2\sigma^2 & \rho^3\sigma^2 & \ldots & \rho^{(H+1)}\sigma^2 \\ \rho^2\sigma^2 & \rho^3\sigma^2 & \rho^4\sigma^2 & \ldots & \rho^{(H+2)}\sigma^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^H\sigma^2 & \rho^{(H+1)}\sigma^2 & \rho^{(H+2)}\sigma^2 & \ldots & \rho^{(2H)}\sigma^2 \end{bmatrix}.$$