European Journal of Operational Research 300 (2022) 602–614

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Production, Manufacturing, Transportation and Logistics

Timed route approaches for large multi-product multi-step capacitated production planning problems

Sébastien Beraudy[a,*], Nabil Absi[a], Stéphane Dauzère-Pérès[a,b]

[a] Department of Manufacturing Sciences and Logistics, Mines Saint-Etienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, CMP, Gardanne F-13541, France
[b] Department of Accounting and Operations Management, I Norwegian Business School, Oslo 484, Norway

ARTICLE INFO

Article history:
Received 19 August 2020
Accepted 10 August 2021
Available online 16 August 2021

Keywords:
Manufacturing
Production planning
Multiple steps
Timed route
Column generation

ABSTRACT

In complex systems that can be found in semiconductor manufacturing, linear programming production planning models must consider many products with hundreds of production steps to be performed on hundreds of machines. To deal with this complexity and solve problems with flexible lead times in a reasonable CPU time, the new concept of timed route is introduced. In a timed route, each production step of a product is associated with a specific time period. A new formulation relying on timed routes is then proposed. Because the number of feasible timed routes can grow exponentially, a column generation approach is presented. Algorithms to generate relevant timed routes are given, and their complexity analyzed. Computational experiments on industrial data with different lead time profiles, fixed lead times and flexible lead times, show that computational times are very significantly reduced when using our approaches, by 92% on average and even divided by more than 1,000 in some cases. The advantages of timed routes are also discussed.

1. Introduction

In many industries, products have to go through a series of production steps (called *a route*) to be completed. When machines are partially flexible, i.e. can process steps of different products or different steps of the same product (reentrant flows), controlling the production flows is difficult. When the cycle times of the routes are long (several weeks), production planning is needed to determine when and how many products should be released, and how they should be processed in order to meet demands on time. This is particularly true in semiconductor manufacturing facilities. In the integrated circuit supply chain, raw wafers need between two and three months to be completed. There might be hundreds of products and each requires hundreds of production steps to be completed, which must be processed on hundreds of partially flexible machines. Machines are grouped in a limited number of workcenters, and products have to pass many times through the same workcenters. Hence, it is often critical for production planning models to consider internal production flows to deliver products on time with reasonable cycle times while satisfying the capacity of machines.

One of the classical ways of modeling congestion in front of machines is to consider both fixed lead times and limited capacity. The limited capacity models the amount of time a machine can be used in a period, while lead times model more exogenous parameters such as the waiting time and the scheduling of products on the machines, see Section 2.3. A fixed lead time is the number of periods that are needed to entirely process a given quantity in a production step in the route of a product. Several assumptions are considered in the literature: (1) The value of the lead time does not depend on the production quantity, (2) The capacity consumption is only considered in the last period of the lead time, and (3) The processing time of a production step is assumed to be smaller than one period. For example, let us consider the products in a production step with batching (e.g. heating in a oven) that will often wait for a complete batch before starting. Hence, the fixed lead time of the production step will be equal to one or more periods although its processing time is smaller than one period. Note that fixed lead times can be equal to zero, in particular for production steps that are short and need to be grouped with other production steps to exceed one period. In this paper, we also consider flexible lead times which, although they require much more complex constraints to be modeled, help to determine more efficient pro-

[*] Corresponding author.
E-mail addresses: sebastien.beraudy@emse.fr, absi@emse.fr, dauzere-peres@emse.fr (S. Beraudy), absi@emse.fr (N. Absi), dauzere-peres@emse.fr (S. Dauzère-Pérès).

https://doi.org/10.1016/j.ejor.2021.08.011
0377-2217/© 2022 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

duction plans, which better use the machine capacity by allowing some products to wait more than the fixed lead time in a step if necessary. However, the resulting linear program may require very long computational times, up to several hundred hours, as shown in the numerical results of Section 6. Another drawback of allowing flexible lead times is that, because lead times have no maximum limits, products may remain too long in the manufacturing system and have very long cycle times. To solve the problem in reasonable computational times, we propose the concept of timed routes to address the multi-product multi-step capacitated production planning problem. In the timed route of a product, each production step of the route of the product is assigned to a period in the planning horizon. A major advantage of timed routes is that the cycle time of products can be limited, by only considering timed routes with a maximum number of periods between the period assigned to the first step of the route and the period assigned to the last step of the route. Note that setup costs and times are not explicitly considered in this paper.

This paper is structured as follows. Section 2 proposes a literature review on related problems such as multi-level lot sizing, integrated lot sizing and scheduling and semiconductor manufacturing production planning, together with an overview on column generation approaches. Then, in Section 3, mathematical models for production planning with fixed lead times and flexible lead times are presented. In Section 4, a reformulation based on the new concept of timed routes is proposed. When modeling flexible lead times instead of fixed lead times, the number of timed routes becomes exponential. Hence, a column generation approach is presented in Section 5 to solve the problem with flexible lead times. Computational experiments on industrial data are conducted in Section 6, which shows the efficiency of the timed route reformulation. Conclusions are drawn and future research directions are provided in Section 7.

2. Literature review

In this section, lot-sizing problems that show similarities with the multi-product multi-step production planning problem are first discussed. However, one main difference is that setup costs and times are not considered in the multi-product multi-step production planning problem. Still, the classical formulation of the problem is really close to a multi-level dynamic lot-sizing problem. Due to the detailed production steps, it is also somehow related to integrated lot-sizing and scheduling problems, although it cannot be considered as an integrated problem. Because it considers multi-product multi-step production planning problems, a review on the semiconductor manufacturing literature on production planning is then given. Anticipating the need to deal with a large number of variables, a short summary on column generation approaches in production planning completes this section.

2.1. Multi-level dynamic lot-sizing

Dynamic lot-sizing problems are classical problems in production management, introduced by Wagner and Whitin (1958). Dynamic lot-sizing aims at determining the production quantities to start on a planning horizon discretized in periods to meet the demand while minimizing inventory, setup and other possible costs. A generalization of this family of problems is the multi-level capacitated lot-sizing problem (MLCLSP), proposed by Billington, McClain, and Thomas (1983). Multi-level lot sizing is generally separated in three branches depending on the Bill of Materials (BOM) structure (note that other specific BOMs may occur). If each product has at most one predecessor product and at most one successor product, it is called production in series. In an assembly structure, each product has at most one successor, whereas, each product has at most one predecessor in a divergent structure. Lead times can be used at every level, not only to model capacity (which is already considered in the capacitated case), but also to consider various exogenous phenomena. The MLCLSP was solved using various heuristics, from Lagrangian heuristics to metaheuristics. The reader is referred to Buschkühl, Sahling, Helber, and Tempelmeier (2010) for a literature review on dynamic capacitated lot-sizing problems.

The problem we want to address is in the family of serial multi-level capacitated lot-sizing problems with lead times without setup costs or times. Not only fixed lead times but also flexible lead times will be considered. Another characteristic of our problem is the shared capacity between levels.

2.2. Integrated lot sizing and scheduling

Because multi-step production planning seeks to grasp the full complexity of internal production flows rather than approximating the cycle times for each product, it can be compared to models that integrate two decision levels such as integrated lot-sizing and scheduling models. Although our multi-product multi-step production planning problems do not explicitly integrate sequencing decisions on machines, routing constraints are partly taken into account by considering lead times for the production steps. Lead times ensure that two consecutive steps in the route of a product are performed in the same period or in two different periods.

Integrated lot-sizing and scheduling problems have often been studied, according to the review of Copil, Wörbelauer, Meyr, and Tempelmeier (2017). While lot sizing aims at meeting the demand at the lowest cost, scheduling corresponds to assigning and scheduling products on machines while minimizing the makespan or other objective functions. Such a junction between operational and tactical problems can lead to high complexity. Solving the full integrated mathematical model with multiple machines and multiple steps is often unrealistic. For this reason, solution methods can be separated in three kinds: Heuristics which solve the full problem (e.g. Gómez Urrutia, Aggoune, & Dauzère-Pérès, 2014), hierarchical methods which solve the problems sequentially (e.g. Liberatore & Miller, 1985), and iterative methods (e.g. Dauzère-Pérès & Lasserre, 1994). In the literature, focus may vary between scheduling oriented modeling and lot sizing oriented modeling. Furthermore, various types of heuristics are proposed depending on the problem to solve.

In a sense, multi-product multi-step production planning problems can be seen as production planning problems that integrate intermediate production decisions. Contrary to the integrated lot-sizing and scheduling problems, the uniformity of the decision variables avoid adversarial decisions between the two levels.

2.3. Production planning in semiconductor manufacturing

As discussed in Section 1, our multi-product multi-step production planning problem is particularly relevant in semiconductor manufacturing. Hence, the literature on production planning in semiconductor manufacturing is the most related to our problem. Note that most research papers in this context only use linear programming. Integer variables are not considered due to the dimension and complexity of the industrial problem. A critical issue in production planning in semiconductor manufacturing is the modeling of the complex reentrant production flows and of the congestion on machines. Congestion is modeled in three main ways.

1. The first and most straightforward way to model congestion is to use fixed lead times, which are equivalent to the ones used

in multi-level lot sizing. Although lead times fail to seize the dynamic of congestion, they can model frequent delays happening due to fixed bottlenecks in the production step. This is for example the case in a workcenter that is central in the production flows and regularly overloaded, or in production steps that require auxiliary resources not always available in the period. The dynamics of the congestion, that can fluctuate due to the workloads in each workcenter, is taken into account by the capacity constraints that limit the quantities to be processed in one period in a workcenter. Lead times are usually determined based on historical data. Fixed Lead Times are easy to model and introduce low complexity but they have several drawbacks. In particular, the workload is not balanced on all periods of the lead time but is only counted in the last period. In addition, production flows are not flexible and the values of the lead times are critical. If the lead times are too short, production flows must be strongly reduced in order to satisfy capacity constraints. Nevertheless, fixed lead times are convenient and can be improved by using non-integer fixed lead times as shown in Kacar, Mönch, and Uzsoy (2016).

2. An important fact is that lead times are not exogenous parameters. In fact, they directly depend on the production flows of products which compete for the same resource in a period. In short, lead times depend on the production plan. To address this circularity between production planning and operational level execution, Hung and Leachman (1996) propose an iterative procedure using both linear programming and discrete event simulation. The linear programming model is used to find a production plan that takes into account the lead times given by the simulation model, while the simulation model takes as inputs the production plan and evaluate it. These two steps are repeated until convergence. However, as stated by Missbauer (2020), processes that iterate only on the lead times, do not meet the theoretical requirements to insure a convergence. Furthermore, the experiments of Bang and Kim (2010) show that iterative procedures are affected by the choice of the simulation model. In addition, a major drawback of iterative procedures is their computational burden.
3. The last main way to tackle congestion is the use of so-called Clearing Functions (CFs). Initially introduced by Graves (1986), Clearing Functions give the expected output of machines (or workcenters) as a function of the workload. In their current shape (since the paper of Asmundsson, Rardin, & Uzsoy, 2006), CFs are non-linear functions that are estimated using simulation or historical data. CF constraints are generally linearized and included in a single linear programming model. In a recent work, Albey, Bilge, and Uzsoy (2017) study a CF that can deal with multiple products and multiple stages. One of the main advantages of using CFs is the short computational times compared to using iterative procedures, because the burden is moved to the pre-processing phase (i.e. establishing CFs). But when the structure of the facility changes, e.g. new machines are added, CFs need to be re-evaluated.

We will not consider CFs in this paper, because they do not specify lead times and capture the delays in a very different way. We first consider fixed lead times which are widely used and easy to implement. Then we use a more flexible definition of lead times by considering additional constraints.

Within the semiconductor manufacturing literature, the classical formulation of a multi-step production planning problem can be tracked back to the step separated formulation of Leachman and Carmon (1992). This paper is also interesting because it might be the first one to discuss a route based formulation of the problem. Unfortunately, the authors discarded the idea due to the large number of decision variables required by the model.

2.4. Column generation for production planning

Column generation approaches are known to efficiently deal with mathematical programs with a large number of decision variables. Column generation has been successfully applied to various optimization problems such as vehicle routing problems (e.g. Azi, Gendreau, & Potvin, 2010), airplane crew scheduling problems (e.g. Gamache, Soumis, Marquis, & Desrosiers, 1999) or machine scheduling problems (e.g. Lopes & de Carvalho, 2007). Column generation was introduced by Dantzig and Wolfe (1960), and consists in separating the original problem into a master problem and a pricing problem that generates useful columns for the master problem. At first, a Restricted Master Problem (RPM) with a limited number of columns is solved. Then, using reduced costs, the pricing problem is solved to find one or several columns to add to the RPM. The process is iterated until no new column is found. To better understand column generation, the reader can refer to Barnhart, Johnson, Nemhauser, Savelsbergh, and Vance (1998) where different strategies of generation are discussed (in a branch and price framework) or the extensive tutorial of Desrosiers and Lübbecke (2005).

In production planning and lot sizing, the first work on column generation was published by Manne (1958), two years prior to the seminal paper of Dantzig and Wolfe (1960). Manne's paper is partially deficient and was corrected and implemented in Degraeve and Jans (2007). Column generation was applied to solve lot-sizing problems in several kinds of industries such as the tire industry (Jans & Degraeve, 2004), the paper industry (Bredström, Lundgren, Rönnqvist, Carlsson, & Mason, 2004) and the steel industry (Yi, Jia, Du, & Liu, 2019). The most commonly used column type is a production plan column which specifies the production periods. However, in terms of production planning without setup costs, the production periods are not critical. That is why the formulation proposed in our study is significantly different. As far as we know, column generation was never applied to solve a multi-product multi-step lot-sizing problem.

In semiconductor manufacturing, to the best of our knowledge, column generation was never used to solve production planning problems. Even in the entire semiconductor manufacturing literature, only four articles using column generation were spotted: On lot allocation to customer (Ng, Sun, & Fowler, 2010), on cutting wafers (Nisted, Pisinger, & Altman, 2011), on capacity expansion (Kim & Uzsoy, 2008) and on scheduling (Jampani & Mason, 2010).

3. Mathematical models with fixed and flexible lead times

In this section, a compact formulation based on the literature is presented for planning the production of P products over a discrete time horizon. The time horizon is decomposed into T periods (usually one period is one day), and demands D_{pt} are given per product p and period t. Each product p needs a sequence of steps $\mathcal{L}_p$ to be processed on K workcenters. Each workcenter k can process a finite set of steps $\mathcal{L}_p^k$ for each product p and has a finite capacity C_k.

The plan is determined by optimizing internal production flows. The goal is to optimize the quantities X_{plt} of product p to be processed, at step l and period t. The set of steps of product p and their resource consumption α_{pl} provide the timing of steps. In order to trace production flows, a variable W_{plt} that models the work in process (WIP) of product p, at step l and period t is introduced. A unitary work in process cost w_{pl} is associated with product p and step l.

The goal is to satisfy demands while minimizing inventory, backlogging and work in process costs. Note that the WIP cost is the intermediate inventory cost in a multi-level lot-sizing problem. We introduce a unitary inventory cost h_{pt} and a unitary backlogging cost b_{pt} for product p and period t. Let us also introduce two decision variables I_{pt} and B_{pt}, that respectively model the inventory and the backlog of product p at time period t. In this model, we assume that the transportation times and costs between two workcenters are negligible or constant. Products that complete a given production step are placed in a waiting queue for the next step (the queue is supposed to be uncapacitated).

Capacity congestion is first modeled with a fixed lead time LT_{pl} for product p at step l. Note that LT_{pl} can be larger than 1 but, as in the models of the literature, we assume that the capacity is consumed in the last period of the lead time, and that the processing time of a production step never exceeds one period (and that it does not overlap two different periods). More precisely, the capacity required to produce X_{plt} is consumed in period t when $LT_{pl} = 0$, in period $t+1$ when $LT_{pl} = 1$, in period $t+2$ when $LT_{pl} = 2$, etc. This means that products are waiting in period t if $LT_{pl} = 1$, in periods t and $t+1$ if $LT_{pl} = 2$, etc. Production capacity is consumed in the same way when flexible lead times are considered.

3.1. Model with fixed lead times

The parameters and decisions variables are summarized below.

- P: Number of products;
- K: Number of workcenters;
- $\mathcal{L}_p$: Sorted list of steps of product p;
- $\mathcal{L}_p^k$: Set of steps for product p processed in workcenter k;
- T: Number of periods in the planning horizon for production;
- α_{pl}: Unitary resource consumption of step l of product p;
- C_k: Daily available resource capacity of workcenter k;
- LT_{pl}: Lead time of step $l \in \mathcal{L}_{(p)}$ of product p;
- D_{pt}: Demand of product p at the end of period t;
- h_{pt}: Unitary inventory cost of product p at the end of period t;
- b_{pt}: Unitary backlogging cost of product p at the end of period t;
- w_{pl}: Unitary work in process cost of product p at step l;
- B_{p0}: Initial backlog of product p;
- I_{p0}: Initial inventory of product p;
- W_{pl0}: Initial work in process of product p at step l.

There are two types of variables: Variables related to the internal production flow ($X_{plt}/Y_{plt}/W_{plt}$), and variables related to the demand (I_{pt}/B_{pt}). Y_{pt}^{out} is a variable linking both sets of variables.

- X_{plt}: Quantity of product p to be released in period t at step $l \in \mathcal{L}_p$;
- $X_{pt}^{\text{in}} = X_{p1t}$: Quantity of product p released in period t;
- Y_{plt}: Quantity of product p completing step $l \in \mathcal{L}_p$ in period t;
- $Y_{pt}^{\text{out}} = Y_{p|\mathcal{L}_p|t}$: Output quantity of product p in period t;
- W_{plt}: Quantity in the Work in process (WIP) of product p, at step $l \in \mathcal{L}_p$ at the end of period t;
- I_{pt}: Inventory level of product p at the end of period t;
- B_{pt}: Backlogging level of product p at the end of period t.

The mathematical model with fixed lead times is written below.

$$\min \quad \sum_{p=1}^{P}\sum_{l\in\mathcal{L}_p}\sum_{t=1}^{T} w_{pl}W_{plt} + \sum_{p=1}^{P}\sum_{t=1}^{T}(h_{pt}I_{pt} + b_{pt}B_{pt}) \tag{1}$$

$$\text{s.t.} \quad Y_{plt} = X_{p(l+1)(t)} \qquad \forall p \in \{1,\dots,P\} \quad \forall l \in \mathcal{L}_p \quad \forall t \in \{1,\dots,T\} \tag{2}$$

$$W_{plt} = W_{pl(t-1)} + X_{plt} - Y_{plt} \qquad \forall p \in \{1,\dots,P\} \quad \forall l \in \mathcal{L}_p \quad \forall t \in \{1,\dots,T\} \tag{3}$$

$$X_{plt} = Y_{pl(t+LT_{pl})} \qquad \forall p \in \{1,\dots,P\} \quad \forall l \in \mathcal{L}_p \quad \forall t \in \{1,...,T-LT_{pl}\} \tag{4}$$

$$D_{pt} + B_{p(t-1)} = Y_{pt}^{\text{out}} + I_{p(t-1)} - I_{pt} + B_{pt} \qquad \forall p \in \{1,\dots,P\} \quad \forall t \in \{1,\dots,T\} \tag{5}$$

$$\sum_{p=1}^{P}\sum_{l\in\mathcal{L}_p^k}\alpha_{pl}Y_{plt} \le C_k \qquad \forall k \in \{1,\dots,K\} \quad \forall t \in \{1,\dots,T\} \tag{6}$$

$$X_{plt}, Y_{plt}, W_{plt}, I_{pt}, B_{pt} \ge 0 \qquad \forall p \in \{1,\dots,P\} \quad \forall l \in \mathcal{L}_p \quad \forall t \in \{1,\dots,T\} \tag{7}$$

The objective function (1) minimizes the total inventory, backlogging and work in process cost. Constraints (2)–(5) model flow conservation. Constraints (2) link the output of step l, Y_{plt}, to the input of the next step, X_{pl+1t}. Constraints (3) balance the work in process over the planning horizon for each step. Constraints (4) guarantee that the fixed lead time for each step of each product is satisfied. Constraints (5) are the flow conservation constraints for the final products, ensuring the satisfaction of demands through the inventory and the production at the current period or their backlogging to subsequent periods. The capacity constraints in each workcenter are modeled through Constraints (6). Constraints (7) ensure the non-negativity of decision variables. Note that, due to Constraints (2), (3) and (4), the decision variables X_{plt}, Y_{plt} and W_{plt} are correlated and could be replaced by a single family of variables. However, in this case, the flexible lead times constraints in the following section cannot be written.

3.2. Model with flexible lead times

Fixed lead times are certainly the most common and easy way to model lead times, but there are other ways to model lead times which allow more flexibility. One possibility is to fix minimum lead times, i.e. that, at each step, products have to wait at least a given minimum lead time but can wait more. To the best of our knowledge, only two papers (Chen, Sarin, & Peake, 2010; Hwang & Chang, 2003) used similar constraints called WIP penetration constraints. These constraints are expressed in order to limit the number of steps a product can perform in a single period. Used with the right parameters, WIP penetration constraints can model the minimum lead times discussed earlier, but can also model lead times on several consecutive steps. The first aim of WIP penetration constraints is to limit the flow of a product, by limiting the number of steps in a single period. In the following, these constraints are called ”flexible lead time constraints”. Let us introduce $o_{\max}(l)$ which represents the maximum number of steps after l (l included) which can be processed in the same period as l. If there is no such limit, $o_{\max}(l)$ is set to $+\infty$. Flexible lead time constraints are expressed by constraints (8). In the model with fixed lead times (1)-(7), Constraints (4) are replaced by Constraints (8).

$$Y_{pl+o_{\max}(l)t} \le \sum_{k=l}^{l+o_{\max}(l)} W_{pk(t-1)} \quad \forall t \in \{1,...,T\} \quad \forall p \in \{1,...,P\} \quad \forall l \in \mathcal{L}_p \qquad \text{s.t. } o_{\max}(l) \ne +\infty \tag{8}$$

Constraints (8) bind the output of step $l + o_{\max}(l)$ with the work in process of previous steps, i.e. products which have not yet completed step $l-1$ cannot be processed in step $l + o_{\max}(l)$.

If $o_{\max}(l) = 0$, Constraints (8) ensure that only products already in the WIP of step l can be produced, i.e., products will have to wait at least one period in the WIP of l, which is a relaxation of the fixed lead time when $LT = 1$. Note that flexible lead times, as shown later in this article, significantly increase the complexity of our problem and the computational time needed to solve it. However, flexible lead times allow internal production flows to be better modeled, and some of the issues related to the fixed lead times

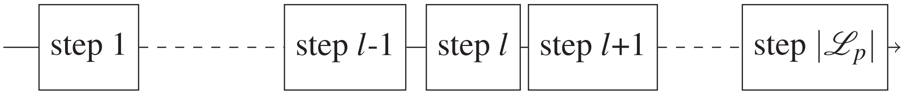

Fig. 1. A production route.

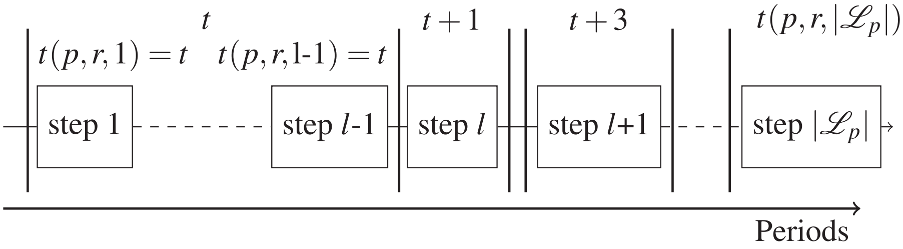

Fig. 2. A timed route.

to be fixed (e.g. the lack of decisions on the quantities to process in intermediate steps). In particular, the use of machine capacity can be smoothed. However, this smoothing could be at the cost of products waiting a long time in the same step, and could potentially lead to large cycle times.

4. A novel formulation using timed routes

In this section, a reformulation of the mathematical models in Section 3 is proposed. The new model is based on the new concept of ”timed route” which is formalized in Section 4.1. Timed routes allow production flows to be fully modeled. The mathematical model using timed routes is introduced in Section 4.2. In Section 4.3, a polynomial time algorithm to generate all possible timed routes with fixed lead times is presented.

4.1. Concept of timed route

A production route is the sequence of steps that a product needs to follow to be completed (see Fig. 1). A timed route is a production route for which a processing period is assigned to each step (see Fig. 2). More formally, in a timed route r, a period $t(p,r,l)$ is assigned to each step l in the route of product p. For example, in Fig. 2, the timed route starts at period t and is completed at period $t(p,r,|\mathcal{L}_p|)$. Note also that step $l+1$ is processed at period $t+3$ and has a lead time of 2 periods. Furthermore, the cycle time of a timed route r of product p is:

$$CT(p,r) = t(p,r,|\mathcal{L}_p|) - t(p,r,1) + 1$$

A timed route is a complete representation of one production flow, with the exact timing of each step. With timed routes, it is possible to exhaustively detail the productions flows, and to know exactly where and when capacity is consumed. The cycle time related to a timed route is explicit, contrary to the classical lead time formulations of Section 3 where, although the cycle time is also fixed, determining it means looking at the set of lead time constraints on the step of the route to extract the total cycle time. With the full view of possible production flows, inconsistent or useless timed routes can be discarded. The timed routes could be validated based on industrial knowledge. Moreover, new constraints on production flows could be introduced such as minimal and maximal cycle times.

4.2. Mathematical model

In the following, the timed route model is formalized. Let us denote $\mathcal{R}_p$ the set of timed routes of product p. With each timed route $r \in \mathcal{R}_p$, a WIP management unitary cost w_{pr} is associated. The WIP cost of a timed route is equivalent to the sum of the WIP costs of the different steps on the time horizon. Only the first step of each period (except for the first period) carries a WIP cost. This WIP cost can be counted several times if no step takes place in the subsequent periods. Let us write the total WIP cost of a given timed route r, $\sum_{l\in\mathcal{L}_p} b_l^{pr} w_{pl}$, where b_l^{pr} is the number of periods between the processing periods of step $l-1$ and step l in timed route r, i.e. $b_l^{pr} = t(p,r,l) - t(p,r,l-1)$. Note that waiting before the first step of a route is not allowed, i.e. $b_1^{pr} = 0$. Let a_{lt}^{pr} a binary parameter which is equal to 1 if, in timed route $r \in \mathcal{R}_p$ of product p, step l is processed in period t, and is equal to 0 otherwise. Z_{pr} is the decision variable that corresponds to the quantity released on timed route r. The timed route formulation is given below.

$$\min \quad \sum_{p=1}^{P}\sum_{r\in\mathcal{R}_p} w_{pr}Z_{pr} + \sum_{p=1}^{P}\sum_{t=1}^{T}(h_{pt}I_{pt} + b_{pt}B_{pt}) \tag{9}$$

$$\text{s.t.} \quad \sum_{p=1}^{P}\sum_{r\in\mathcal{R}_p}\sum_{l\in\mathcal{L}^k} a_{lt}^{pr}\alpha_{pl}Z_{pr} \le C_k \qquad \forall k\in\{1,\ldots,K\} \quad \forall t\in\{1,\ldots,T\} \tag{10}$$

$$I_{pt} \ge \sum_{r\in\mathcal{R}_p}\sum_{\tau=1}^{t} a_{|\mathcal{L}_p|\tau}^{pr}Z_{pr} - \sum_{\tau=1}^{t} D_{p\tau} \qquad \forall p\in\{1,\ldots,P\} \quad \forall t\in\{1,\ldots,T\} \tag{11}$$

$$B_{pt} \ge -\sum_{r\in\mathcal{R}_p}\sum_{\tau=1}^{t} a_{|\mathcal{L}_p|\tau}^{pr}Z_{pr} + \sum_{\tau=1}^{t} D_{p\tau} \qquad \forall p\in\{1,\ldots,P\} \quad \forall t\in\{1,\ldots,T\} \tag{12}$$

$$Z_{pr}, I_{pt}, B_{pt} \ge 0 \qquad \forall p\in\{1,\ldots,P\} \quad \forall r\in\mathcal{R}_{(p)} \quad \forall t\in\{1,\ldots,T\} \tag{13}$$

The objective function (9) minimizes the total backlog, inventory and WIP management cost induced by the selected timed routes, which is equivalent to the objective function (1). Constraints (10) model the limit on capacity consumption in each workcenter at every period, and correspond to Constraints (6). Constraints (11) and (12) ensure the inventory balance. They are equivalent to Constraints (5) but are written separately to simplify the writing of the dual problem. This formulation can be seen as a covering problem.

4.3. Generation of timed routes associated with fixed lead times

Let us show how the set of timed routes is determined when fixed lead times are considered. Due to Constraints (2) and (4) in

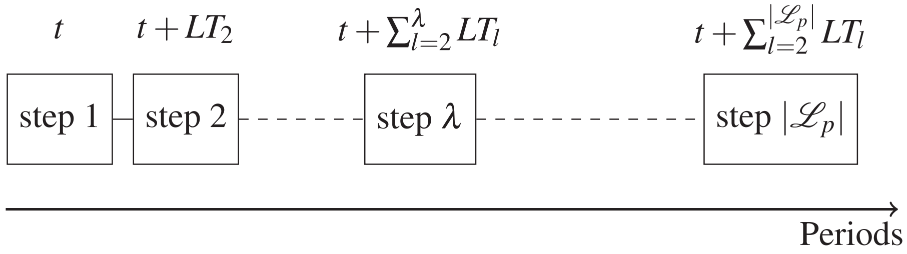
t
$t+LT_2$
$t+\sum_{l=2}^{\lambda} LT_l$
$t+\sum_{l=2}^{|\mathscr{L}_p|} LT_l$
step 1
step 2
step λ
step $|\mathscr{L}_p|$
Periods

Fig. 3. Pattern of timed routes with Fixed Lead Times.

the model with fixed lead times, all production flows on a route follow the same pattern. If t is the first period of the route and $|\mathcal{L}_p|$ the number of steps of product p, then the pattern can be designed as the timed route in Fig. 3. The pattern is used for every period t with $t \leq T - \sum_{l=2}^{|\mathcal{L}_p|} LT_l$. The algorithm generates one timed route per period for each product p. Since we need to assign a period to each of the $|\mathcal{L}_p|$ steps, the complexity of generating all the timed routes with fixed lead times for a product p is equal to $O(|\mathcal{L}_p|T)$. Thus, the complexity of the algorithm that creates all the timed routes is equal to $O(\sum_{p=1}^{P} |\mathcal{L}_p|T)$. This complexity can be written as follows: $O(P|\bar{\mathcal{L}}|T)$, where $|\bar{\mathcal{L}}|$ is the average number of steps in a route.

5. A column generation approach for flexible lead times

Because, as shown in this section, the number of timed routes with flexible lead times is exponential, we propose a column generation approach to solve the timed route formulation. In Section 5.1, a dynamic programming algorithm that generates all the timed routes when considering flexible lead times is described. The column generation approach is introduced in Section 5.2, where reduced costs associated with timed routes are evaluated and used to implement a dominance rule to strengthen the algorithm of Section 5.1.

5.1. Exhaustive generation of timed routes for flexible lead times

Using timed routes, all production flows can be described and traced. Thus, we can consider other production flows than the ones generated using fixed lead times. Considering several timed routes with different lead times for one step leads to more flexibility. This is the case with the flexible lead times presented in Section 3.2. Furthermore, when using the timed route formulation with flexible lead times, it is possible to avoid products with too large cycle times.

To establish a timed route, each step needs to be assigned to a period in the horizon. Representing this assignment by a graph, nodes are labeled (s, c, t, l) where s is the index of the current partial route, c the current partial cost, t the period and l the last step that is completed in the partial timed route s. The directed edges are the possible sequences of nodes. Due to the structure of a route, the graph can be seen as a tree with a level structure. Note that, when two or more steps $(\{l'+1, \ldots, l''\})$ are completed within the same period t (where l' is the last step completed before t), we do not create a node for each step $l \in \{l'+1, \ldots, l''\}$. Instead, we contract the steps $\{l'+1, \ldots, l''\}$ within a single node and we create a direct arc to the final step l''. This means that for a timed route r, $a_{lt}^{pr} = 1, \ \forall l \in \{l'+1, \ldots, l''\}$ if $(s1, c_1, l', t-1)$ and $(s2, c_2, l'', t)$ are successive nodes of the timed route. Fig. 4 provides an example of such graph, with 2 steps and 3 periods. Using this kind of graphs, an algorithm generating dynamically the edges and new vertices level by level will work well.

Rather than exploring the total space of possible states, the number of vertices is reduced by using $o_{max}(l)$, the maximal number of steps that can be processed after step l in the same period than l. The vertices and edges which can be used when $o_{max}(l) = 1$ for every step are traced with plain arrows and in blue in Fig. 4. Even with this reduction, the total number of timed routes for product p is still in $O(|\mathcal{L}_p|^T)$ because, at each step of the route, a period between 1 and T can be assigned.

The exhaustive generation (which becomes a dynamic program in Section 5.2) can be implemented as described in Algorithms 1 and 2. The main algorithm (Algorithm 1) generates all timed routes. It starts with a set of partial timed routes only containing the partial timed route with no period assigned, labeled (s_0,0,0,0). For each period, the algorithm tries to extend the set of partial timed routes by looking for the children nodes of each partial timed route and the initial partial timed route. This procedure is developed in Algorithm 2. Note that the generated partial timed routes are not removed in Algorithm 2. Each partial timed routes can be extended to a subsequent period and the lead time to complete the next step increases accordingly.

Algorithm 1 Generation of timed routes

```
$CTR = \emptyset$    // $CTR$: Set of complete timed routes
$PTR = \emptyset$    // $PTR$: Set of current partial timed routes
for $t$ = 1 to T do
    $ir$    // $ir$: Initial partial timed route
    laststep($ir$) = 0    // No step assigned to $ir$
    $PTR = PTR \cup \{ir\}$
    for all $s \in PTR$ do
        CreateExtensions($s, t$)
    end for
    $PTR = PTR - \{ir\}$
end for
return $CTR$
```

Algorithm 2 CreateExtensions(s, t)

```
$l$ = laststep($s$) + 1
for $e$ = 0 to $o_{max}(l)$ do
    $sr$ = $s$
    for $i$ = 0 to $e$ do
        step($sr, l+i$) = $t$
    end for
    laststep($sr$)= $l+e$
    if $l+e = |\mathcal{L}_p|$ then
        $CTR = CTR \cup \{sr\}$
    else
        $PTR = PTR \cup \{sr\}$
    end if
end for
```

In Algorithm 2, the partial time routes are returned, which extend the input partial timed route in period t. Extending a partial timed route means looking for each outgoing edge from the last

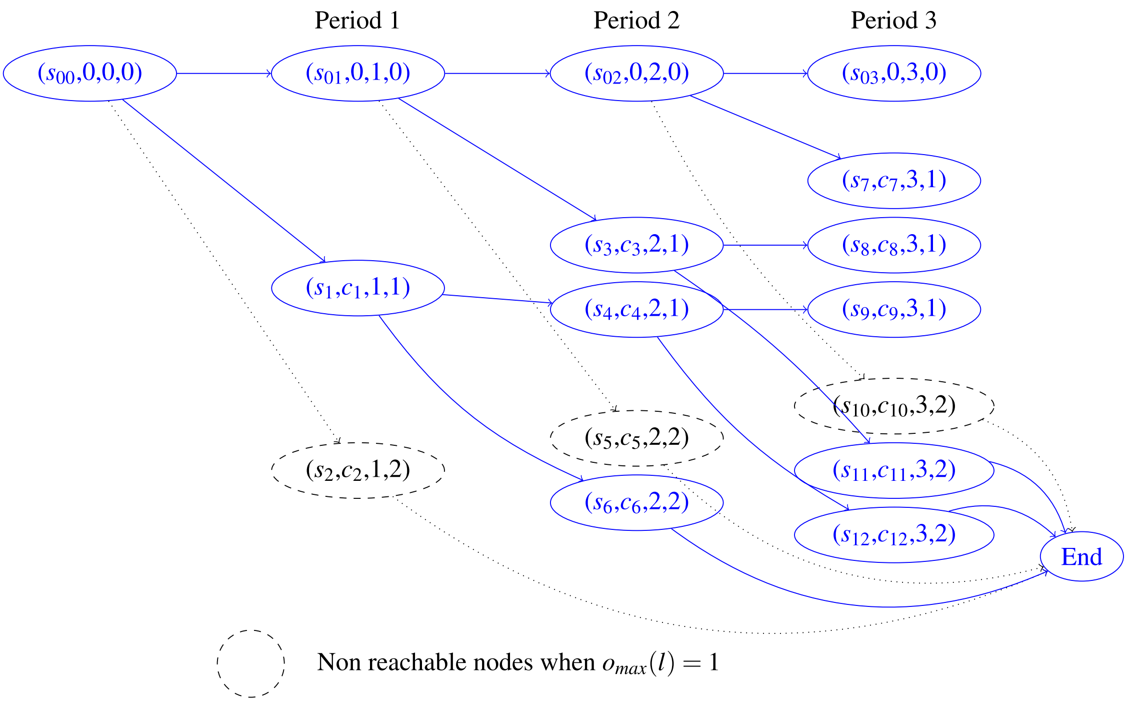

Fig. 4. Graph of states: Example with 2 steps and 3 periods

node in the graph depicted earlier. The number of partial timed routes generated is $o_{max}(l)$ where l is the last step assigned in the input partial timed route. The information on the last step is updated in the new partial timed routes.

The program explores all possibilities, which leads to an exponential number of routes. At each period t, for a product p, $o_{max}(l)$ states are evaluated. Note that in the worst case $o_{max}(l)$ is equal to the total number of steps $|\mathcal{L}_p|$. The complexity of evaluating each state is constant. If no dominance rule is used, the total complexity increases exponentially and is equal in the worst case to $O(|\mathcal{L}_p|^T)$.

Note also that it is possible to generate patterns of timed routes as in Section 4.3 for fixed lead times. However, this column generation approach is not the most relevant approach because, for a pattern, the timed routes starting at different periods may have different costs. Moreover, if all the timed routes of a selection of patterns are used, the associated useless decision variables may burden the linear program.

5.2. Column generation approach

The set of timed routes for flexible lead times is exponential, as shown by the complexity of the exhaustive generation algorithm. To handle this issue, we propose a column generation approach, in which timed routes are generated dynamically. The framework of the approach can be found in Fig. 5.

The master problem corresponds to the model in Section 4.2. Thus, the Restricted Master Problem (RMP) is written with a restricted set of timed routes for each product. The restricted set of timed routes is initialized with the timed routes generated with fixed lead times. A fast resolution of the pricing problem, that generates new improving timed routes, is critical to the success of the column generation approach. An efficient algorithm is proposed in the following section.

5.2.1. Solving the pricing problem

To determine the timed routes to insert in the RMP, we consider the reduced costs associated with timed routes. The dual problem associated with the timed route formulation corresponds to (14)–(17), where λ_{kt} denote the dual variables associated with Constraints (10), and β^+_{pt} (resp. β^-_{pt}) denote the dual variables associated with Constraints (11) (resp. Constraints (12)).

$$\max \quad -\sum_{t=1}^{T}\sum_{k=1}^{K} C_k\lambda_{kt} - \sum_{p=1}^{P}\sum_{t=1}^{T}\Big(\sum_{\tau=1}^{t} D_{p\tau}\Big)\beta^+_{pt} + \sum_{p=1}^{P}\sum_{t=1}^{T}\Big(\sum_{\tau=1}^{t} D_{p\tau}\Big)\beta^-_{pt} \tag{14}$$

$$\text{s.t.} \quad -\sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{l\in\mathcal{L}^k} a^{pr}_{lt}\alpha_{pl}\lambda_{kt} - \sum_{t=1}^{T}\sum_{\tau=1}^{t} a^{pr}_{|\mathcal{L}_p|\tau}\beta^+_{pt} + \sum_{t=1}^{T}\sum_{\tau=1}^{t} a^{pr}_{|\mathcal{L}_p|\tau}\beta^-_{pt} \le w_{pr} \quad \forall p\in\{1,\dots,P\},\ \forall r\in\mathcal{R}_p \tag{15}$$

$$\beta^+_{pt} \le h_{pt} \qquad \forall p\in\{1,\dots,P\},\ \forall t\in\{1,\dots,T\} \tag{16}$$

$$\beta^-_{pt} \le b_{pt} \qquad \forall p\in\{1,\dots,P\},\ \forall t\in\{1,\dots,T\} \tag{17}$$

$$\lambda_{kt}, \beta^+_{pt}, \beta^-_{pt} \ge 0 \qquad \forall t\in\{1,\dots,T\},\ \forall p\in\{1,\dots,P\},\ \forall k\in\{1,\dots,K\} \tag{18}$$

In the dual problem, only Constraints (15) are related to timed routes. Thus, in the column generation approach, we only need to look for timed routes which violate the most Constraints (15), i.e. timed routes with reduced cost $w_{pr} + \sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{l\in\mathcal{L}^k} a^{pr}_{lt}\alpha_{pl}\lambda_{kt} + \sum_{t=1}^{T}\sum_{\tau=1}^{t} a^{pr}_{|\mathcal{L}_p|\tau}\beta^+_{pt} - \sum_{t=1}^{T}\sum_{\tau=1}^{t} a^{pr}_{|\mathcal{L}_p|\tau}\beta^-_{pt} \le 0$. Note that since there is no constraint linking the products in the pricing problem, timed routes can be generated separately for each product.

In order to define a route, we need to assign each step l to a period t, i.e. to determine a^{pr}_{lt}. The reduced cost can be decomposed into three parts.

1. A period assignment cost which is denoted $\alpha_{pl}\lambda_{kt}$,
2. The WIP cost of the route, which can be decomposed into the WIP cost at each period,

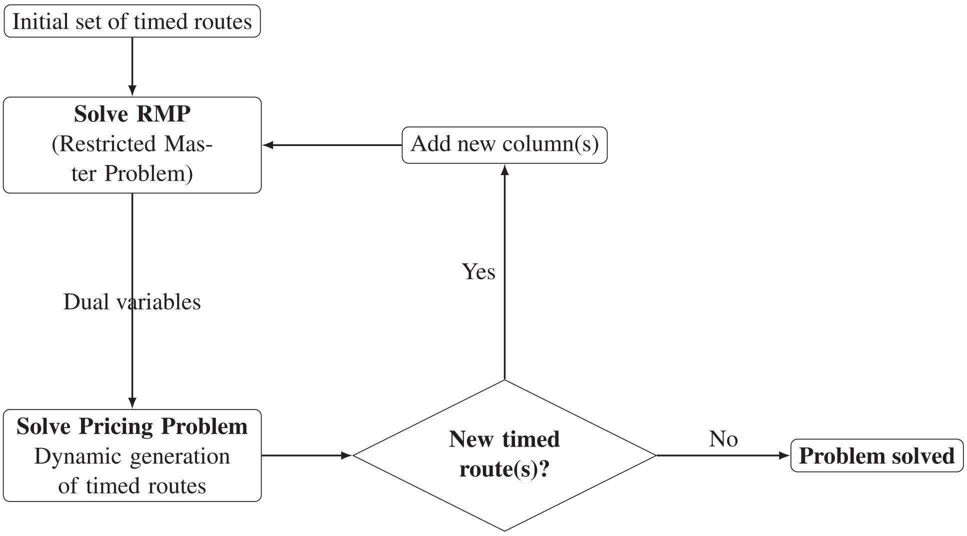

Fig. 5. Framework of column generation approach for production planning

3. Inventory and backlog costs. If the period of the last step (i.e. when the product is completed) is t^*, then the inventory and backlog costs are equal to $\sum_{t=t^*}^{T}(\beta_{pt}^{+}-\beta_{pt}^{-})$.

5.2.2. Dominance rule

With such a complexity, the dynamic program can hardly be used in practice. In order to keep the computational times under control, we consider a dominance rule that relies on Property 1.

Algorithm 3 CreateNonDominatedExtension($s, t, ND[]$)

```
// $ND[]$: Array (of size $|\mathcal{L}_p|$ for product $p$) of dominant partial timed routes up to period $t$-1 indexed by the last step reached.
$l$ = laststep($s$) + 1
for $e$ = 0 to $o_{max}(l)$ do
    $sr$ = $s$   // Extend timed route $s$ by $e$ steps to perform at period $t$
    for $i$ = 0 to $e$ do
        step($sr, l+i$) = $t$
        UpdateReducedCost($sr$)
    end for
    laststep($sr$)= $l+e$
    if $l+e = |\mathcal{L}_p|$ then
        $CTR = CTR \cup \{sr\}$
    else
        // Dominance check
        if ReduceCost($sr$) > ReducedCost($ND[l+e]$) then
            // $sr$ dominates the former dominant partial timed route, which ends at period $t$ with step $l+e$
            $PTR = PTR \cup \{sr\}$
            $PTR = PTR \setminus \{ND[l+e]\}$
            $ND[l+e] = sr$
        end if
    end if
end for
```

Property 1. *For product p at a period t, if two partial timed routes s_1 and s_2 have achieved the same number of steps l, then the route with the lowest partial reduced cost dominates the other. In other words, for s_1=(1,rc_1, t, l) and s_2 =(2,rc_2, t, l), then s_1 dominates s_2 if and only if $rc_1 \le rc_2$.*

Proof. It can be shown by contradiction that, if the periods or the last steps are different, then an arbitrary large negative reduced cost can be introduced in the complete and dominated timed route. Thus, we can introduce s_3, the optimal part to complete s_1 and s_2 to form a complete timed route. We denote rc_3 the reduced cost associated with s_3 and $s_1 \oplus s_3$ (respectively $s_2 \oplus s_3$) the complete timed route associated with s_1 (resp. s_2) and its total reduced cost $rc_{1\oplus 3}$ (resp. $rc_{2\oplus 3}$). Because $rc_{1\oplus 3} = rc_1 + rc_3$ and $rc_{2\oplus 3} = rc_2 + rc_3$, comparing the total reduced cost $rc_{1\oplus 3}$ and $rc_{2\oplus 3}$ is equivalent to comparing the partial reduced cost rc_1 and rc_2. □

Note that, if constraints on the duration of cycle times are introduced, some conditions on the start period of partial timed routes are needed to apply this dominance rule.

By applying this dominance rule in the dynamic program, the number of new partial timed routes at the end of each iteration/period is at most equal to the number of steps for a product. Thus, at iteration t of the algorithm for a given product p, the number of partial timed routes before dominance is smaller than $|\mathcal{L}_p|^2$. It reduces the complexity of Algorithms 1 and 2 to $O(|\mathcal{L}_p|^2 T)$ for each product. To implement the dominance rule, we use in Algorithm 3 an array that contains the dominant partial timed routes (at the currently explored period) for each step of the route (except for the final step). The size of this array, denoted $ND[]$, is $|\mathcal{L}_p|$, and thus it does not add any spatial complexity. Thus, the overall complexity is in $O(T\sum_{p=1}^{P}|\mathcal{L}_p|^2)$.

6. Computational experiments

Computational experiment have been conducted on industrial data to show the efficiency of the timed route formulation and our column generation approach. In Section 6.1, the design of the computational experiments is detailed. In Section 6.2, the compact formulation (1)–(7) and the timed route reformulation (9)–(13) are compared for fixed lead times. Section 6.3 compares the column generation approach with flexible lead times and the compact formulation. The advantages of using flexible lead times compared to fixed lead times are not analyzed in this paper.

6.1. Design of experiments

Experiments are conducted on industrial data of a semiconductor manufacturing facility in France. Data cannot be made public for confidentiality reasons, but can be provided on request after

Table 1
Characteristics of the industrial instances.

Horizon length	{91, 119, 147}
Number of workcenters	10 (aggregating about 500 machines)
Number of products	{15, 40, 75}
Demand scenario	{Low, Medium, High}
Number of steps per route	Between 100 and 500

Table 2
Unitary costs used in the experiments.

Backlog	50
Inventory	15
WIP management	0.001

validation by the company and certification by the researchers that they will not disclose the data to others. We would also want to recall that the main contribution of the paper is not related to the quality of the results obtained by the proposed approach but rather to the significant reduction of the computational times to solve the problem and to the genericity of the approach. The main characteristics of the instances can be found in Table 1. Crossing all choices of the characteristics, 27 scenarios are considered.

Instances are characterized by a number of steps per product that varies between 100 and 500 and cumulative processing times of products that vary between 7 and 11 periods. Note that the planning horizon should be long because products have cycle times between 40 and 80 periods. To generate demands, the historical output over 6 months was considered. With these historical data, the order frequency, the average demand and the standard deviation for each product were estimated. Then, demand scenarios were randomly generated based on these characteristics. We only consider the most produced products. For example, products with very low demands or with less than 50 steps are not considered since they are generally related to R&D and engineering projects and not customer demands. To study the influence of the number of products, we consider 3 sets of products. Each demand scenario, related to the number of products, is then adjusted with a factor on the generated demand to produce 3 scenarios where, respectively, demand is low and feasible, demand is medium but stresses the facility capacity and demand is high and cannot be fully met. The unitary costs used in our experiments can be found in Table 2.

Furthermore, three profiles of lead times are studied by solving the compact models and using the column generation approach.

1. The first profile, $\mathcal{P}_{LT}^{\text{fixed}}$, corresponds to the classical fixed lead times.
2. The second profile, $\mathcal{P}_{LT}^{\text{flex}}$, corresponds to flexible lead times and is based on $\mathcal{P}_{LT}^{\text{fixed}}$, but products can wait in every step as many periods as necessary. This implies that the minimal lead times to be respected are the fixed lead times. This lead time profile reduces the backlog and inventory costs by allowing more flexible production flows.
3. The third profile, $\mathcal{P}_{PT}^{\text{flex}}$, also corresponds to flexible lead times but is based on the actual processing times, i.e. it is not related to the two other lead time profiles. With profile $\mathcal{P}_{PT}^{\text{flex}}$, production flows are only limited by the maximum number of steps for a product that can be completed in a period, according to the cumulative process times of these steps. In a sense, it is a relaxation of the previous model where delays are not induced by exogenous parameters. Note that, contrary to $\mathcal{P}_{LT}^{\text{flex}}$ where Constraint (8) is not written, when $LT(l) = 0$ for a step l, with $\mathcal{P}_{PT}^{\text{flex}}$ Constraint (8) is written for every step.

As show in the computational results of Section 6.3.3, $\mathcal{P}_{PT}^{\text{flex}}$ leads to the most difficult problems in terms of computational time. For example, with the compact formulation, on scenarios with medium or large dimensions, there is at least a factor of ten between the computational times for $\mathcal{P}_{LT}^{\text{flex}}$ and $\mathcal{P}_{PT}^{\text{flex}}$.

All numerical experiments were executed on a computer with a processor Intel(R) Xeon(R) CPU W3550 and 16 Gigabyte of RAM Memory, using a JAVA program (JRE 1.8) and IBM ILOG CPLEX (version 12.6) with default settings.

6.2. Comparison between the compact formulation and the timed route reformulation with fixed lead times

Due to the polynomial number of timed routes with fixed lead times, all timed routes are generated and included in the model. Table 3 shows the computational times spent by IBM ILOG CPLEX for several scenarios. First, note that the computational times do not seem to change much with the demand level. Thus, only looking at the medium scenarios, it can be seen that the timed route model performs better than the compact one. On average, the computational time is decreased by 94%, with a minimum decrease of 88%. When considering the impact of the horizon length, the results show that the timed route formulation is more sensitive to the horizon length than the compact model. The gap between the computational times of both models reduces as the horizon length increases. For all these scenarios, the computational times of the timed route formulation are always smaller than the smallest computational time with the compact formulation. For fixed lead times, the timed route formulation is efficient when all the timed routes are generated. One of the reasons behind the decrease of the computational times may be that IBM ILOG CPLEX needs to eliminate much fewer columns to determine the reduced LP with the timed route formulation than with the compact formulation. We may hypothesize that the reduced LP is close to the timed route model.

6.3. Column generation approach for flexible lead times

In this section, the compact formulation and the timed route formulation with flexible lead time profiles are compared. The first flexible lead time profile studied is $\mathcal{P}_{LT}^{\text{flex}}$. The associated compact model has a lower number of lead time constraints compared to the compact model with fixed lead times. This is due to the fact that lead time constraints are only introduced for positive lead times. The second flexible lead time profile is $\mathcal{P}_{PT}^{\text{flex}}$. Its compact formulation has about the same number of constraints as the compact formulation with fixed lead times, but production flows are less constrained. The associated flexible lead time constraints are based on the actual processing times of steps.

As shown in Section 5.1, the timed route formulation with flexible lead times requires an exponential number of timed routes. To get a feeling of the resulting complexity, we generate all the timed routes for a reduced data set with 3 products with at most 23 steps and 11 machines. With 8 Gigabyte of RAM and when the horizon is larger than 15 periods, it is not possible to generate all timed routes for profile $\mathcal{P}_{LT}^{\text{flex}}$ and a memory error arises.

Note that, in this paper, the computational time is defined as the difference between the time at which the optimization process starts and the time at which the optimal solution is found and extracted. Only the time to load the data and to create the first mathematical model is omitted.

In Section 6.3.1, the parameters and strategies used in the column generation approach are detailed. The experimental results for profile $\mathcal{P}_{LT}^{\text{flex}}$ are presented in Section 6.3.2 while the results for profile $\mathcal{P}_{PT}^{\text{flex}}$ are analyzed in Section 6.3.3.

Table 3
Computational times (in seconds) for profile $\mathcal{P}_{LT}^{\text{fixed}}$ (C: Compact formulation; TR: Timed Route formulation).

Number of products	Horizon length	Low demand		Medium demand		High Demand	
		C	TR	C	TR	C	TR
Low (15)	**91**	15	0	15	0	14	0
	119	20	1	20	1	19	1
	147	25	5	25	3	25	4
Medium (40)	**91**	40	1	41	1	40	1
	119	58	3	57	4	58	4
	147	68	7	70	7	71	7
Large (75)	**91**	84	1	83	1	83	2
	119	113	4	114	8	114	5
	147	145	16	147	15	145	16

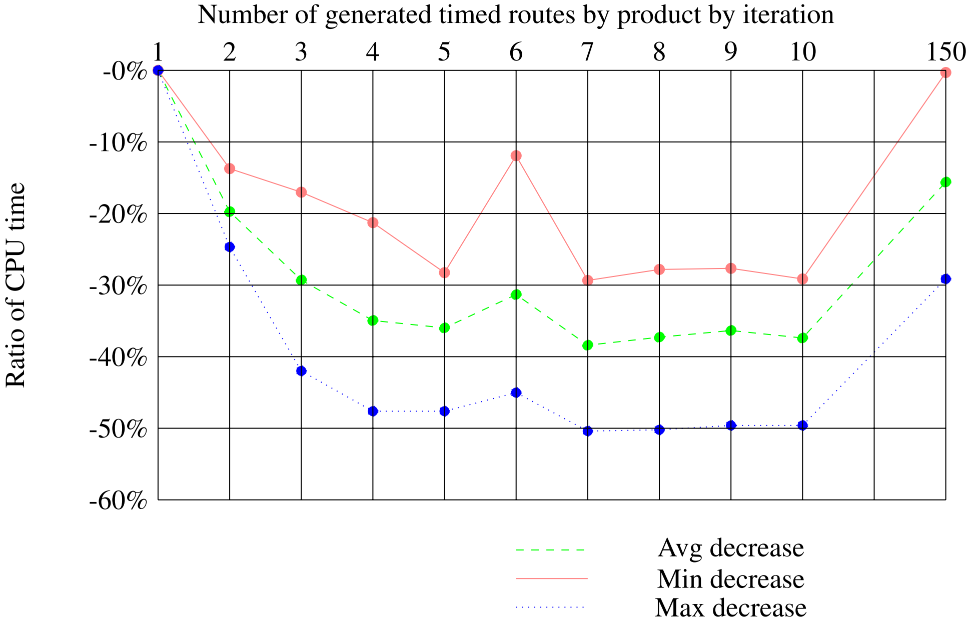

Fig. 6. Number of timed routes by product at each iteration vs. ratio of CPU time

6.3.1. *Column generation strategy*

Dominance rules are used to reduce computational times. To warm up the column generation approach, all timed routes from $\mathcal{P}_{LT}^{\text{fixed}}$ are included in the model. Due to light use of processor during the timed route generation, parallelism is enabled while generating timed routes for each product.

The last parameter to choose is how many timed routes are selected for each product at each iteration. This parameter is tuned with the case of Medium demand, with profile $\mathcal{P}_{LT}^{\text{flex}}$. Fig. 6 shows the average decrease of the computational time over all scenarios compared to the case in which only one timed route is generated by product. This case is used as a reference because it corresponds to the case with the smallest number of timed routes. This figure is completed with the maximal and minimal decrease of the computational time obtained among the 27 scenarios. Note that the average time spent to solve the timed route formulation limited to one new timed route by product at each iteration is 239 seconds. It can be seen that, when the parameter varies between 4 and 10, the decrease of the computational time is quite stable and the lowest. With up to 150 timed routes by product (which is an upper bound to the number of non dominated timed routes generated by the dynamic program when $T < 150$), it can be seen that the decrease of the computational time is similar to when the parameter is set to 2. This figure shows the trade-off between generating numerous columns to converge with fewer iterations and generating only the best columns to accelerate the resolution of the restricted master problem.

In the following experiments, the number of timed routes by product at each iteration is set to 5. This choice might not be the best in every scenario, but is relevant enough to show the strength of our approach.

6.3.2. *Comparison of computational times for profile $\mathcal{P}_{LT}^{flex}$*

Contrary to fixed lead times, the computational times for flexible lead time profiles depend on the demand scenario. Table 4 shows the computational times to solve $\mathcal{P}_{LT}^{\text{flex}}$. No simple rule can be deduced (for both formulations) from the different scenarios because the complexity of the problem depends on several parameters. Computational times to solve the timed route model are quite close with medium and high demands, and are always larger than the computational times with low demands.

The main result of the experiment is that the column generation approach always significantly performs better. On average, the computational time is reduced by 87.5% while the solution time for the compact model ranges from 2 minutes to 79 minutes. The least impressive case is 73.3% when the time spent by the compact formulation is the lowest (120 seconds). Unlike fixed lead times, we cannot conclude anything on the behavior of the compact model when the horizon increases, only that the computational times increase with the length of the horizon (which is expected due to the algorithm complexity).

Table 4
Computational times (in seconds) for profile $\mathcal{P}_{LT}^{\text{flex}}$ (C: Compact formulation; TR: Timed Route formulation).

Number of products	Horizon length	Low demand		Medium demand		High Demand	
		C	TR	C	TR	C	TR
Low (15)	**91**	192	14	142	31	120	32
	119	409	59	266	67	284	66
	147	580	95	595	121	580	124
Medium (40)	**91**	648	34	469	66	563	63
	119	1,190	109	1,086	134	1,254	153
	147	1,674	180	2,034	218	2,174	218
Large (75)	**91**	1,620	23	1,578	107	1,363	109
	119	3,693	74	3,000	236	2,145	242
	147	4,277	316	3,797	401	4,619	395

Table 5
Computational times (in seconds) for profile $\mathcal{P}_{PT}^{\text{flex}}$ (C: Compact formulation; TR: Timed Route formulation).

Number of products	Horizon length	Low demand		Medium demand		High Demand	
		C	TR	C	TR	C	TR
Low (15)	**91**	183	33	1,549	37	1,760	51
	119	272	48	2,677	83	2,836	120
	147	4,100	161	4,211	255	4,962	407
Medium (40)	**91**	5,587	84	6,254	156	6,486	233
	119	8,939	167	10,092	298	10,277	429
	147	13,014	291	14,407	587	16,152	793
Large (75)	**91**	979	95	10,516	179	11,902	246
	119	18,862	193	18,498	346	20,472	460
	147	599,443	404	323,891	678	546,596	1,273

Table 6
Number of iterations in the column generation approach with flexible lead time profiles.

Number of products	Horizon length	Low demand		Medium demand		High Demand	
		$\mathcal{P}_{LT}^{\text{flex}}$	$\mathcal{P}_{PT}^{\text{flex}}$	$\mathcal{P}_{LT}^{\text{flex}}$	$\mathcal{P}_{PT}^{\text{flex}}$	$\mathcal{P}_{LT}^{\text{flex}}$	$\mathcal{P}_{PT}^{\text{flex}}$
	91	25	11	46	12	47	17
	119	57	12	61	21	60	31
Low (15)	**147**	60	31	72	50	71 [a]	80 [a]
	91	22	12	37	23	36	35
	119	38	16	43	30	48	46
Medium (40)	**147**	47	23	46	45	47 [a]	66 [a]
	91	8	7	32	14	35	19
	119	16	11	40	18	43	25
Large (75)	**147**	42	17	46	25	46 [a]	54 [a]

[a] scenarios where the number of iterations for $\mathcal{P}_{PT}^{\text{flex}}$ is higher than for $\mathcal{P}_{LT}^{\text{flex}}$

6.3.3. Comparison of computational times for profile $\mathcal{P}_{PT}^{flex}$

Considering profile $\mathcal{P}_{PT}^{\text{flex}}$ whose computational results can be found in Table 5, some conclusions are shared with $\mathcal{P}_{LT}^{\text{flex}}$. For example, the computational times vary depending on the demand scenario, but in the case of $\mathcal{P}_{PT}^{\text{flex}}$, it can also be noted that the larger the demand, the larger the CPU time to solve the problem, and the increase depends on the scenario. The computational times are again highly reduced by the column generation approach on the timed route formulation. On average, they are reduced by 95.8%. The computational time for the compact model ranges from 3 minutes to more than 6 days (with a median of 2.5 hours).

With the compact formulation, there is a huge gap on the computational times for the three lead time profiles. Due to the extreme computational time in the scenarios with a large number of products and a long horizon, the average computational time is a biased indicator. Therefore, we prefer to analyze the median computational time. Over all the scenarios, the median computational time is 58 seconds for $\mathcal{P}_{LT}^{\text{fixed}}$, 1,190 seconds for $\mathcal{P}_{LT}^{\text{flex}}$ and 8,939 seconds for $\mathcal{P}_{PT}^{\text{flex}}$. When using the timed route formulation and the column generation approach, the computational times also increase as the lead time profile becomes more complex, but the increase is much more limited. The overall median of the computational times for the compact formulation is equal to 3 seconds for $\mathcal{P}_{LT}^{\text{fixed}}$, 109 seconds for $\mathcal{P}_{LT}^{\text{flex}}$ and 233 seconds for $\mathcal{P}_{PT}^{\text{flex}}$. One reason which can explain why computational times for the timed route formulation with $\mathcal{P}_{PT}^{\text{flex}}$ is close to $\mathcal{P}_{LT}^{\text{flex}}$, might be the difference of these two lead time profiles. It can be seen in Table 6 that, in most scenarios (except when the demand is high and the horizon is long), $\mathcal{P}_{PT}^{\text{flex}}$ needs fewer iterations of the column generation approach to converge to the optimal solution.

The reason is probably that, while the compact formulation struggles with a huge number of constraints, many useful timed routes are quickly generated in the column generation approach, thus fewer iterations are needed before converging. It could be interesting to tune the maximum number of timed routes by product at each iteration.

Additionally, the mean computational times to generate timed routes at each iteration for both flexible lead time profiles are given in Table 7. Note that the first iterations of the column generation approach usually take longer computational times but, for most iterations, the computational times are close to the average. Table 7 shows that the generation of timed routes is almost independent of the demand scenario. The computational times to generate timed routes for profile $\mathcal{P}_{PT}^{\text{flex}}$ are about 6 times larger than the computational times to generate the timed routes for profile $\mathcal{P}_{LT}^{\text{flex}}$, and this ratio is stable in all demand scenarios.

Table 7
Mean computational time to generate timed routes by iteration, with flexible lead time profiles.

Number of products	Horizon length	Low demand		Medium demand		High Demand	
		$\mathcal{P}_{LT}^{\text{flex}}$	$\mathcal{P}_{PT}^{\text{flex}}$	$\mathcal{P}_{LT}^{\text{flex}}$	$\mathcal{P}_{PT}^{\text{flex}}$	$\mathcal{P}_{LT}^{\text{flex}}$	$\mathcal{P}_{PT}^{\text{flex}}$
	91	0.4	2.6	0.4	2.7	0.4	2.6
	119	0.5	3.4	0.5	3.4	0.5	3.4
Low (15)	**147**	0.7	4.3	0.7	4.2	0.7	4.2
	91	0.9	6.0	0.9	5.9	0.9	5.8
	119	1.2	8.3	1.2	8.0	1.3	7.7
Medium (40)	**147**	1.7	9.9	1.6	10.1	1.6	9.6
	91	1.7	10.2	1.6	10.2	1.5	10.3
	119	2.3	13.9	2.3	14.1	2.3	13.8
Large (75)	**147**	3.0	17.7	3.0	18.3	3.1	17.5

7. Conclusions and perspectives

In this paper, we introduced the novel concept of timed route that enables a new model for multi-product multi-step production planning problems to be introduced. The timed route approach was validated on industrial data, and experimental results show that the new formulation significantly outperforms compact formulations for various lead time profiles. To achieve such performance and because considering flexible lead times induces an exponential number of columns, a column generation approach was presented with a polynomial dynamic program that generates the timed routes in the pricing problem.

Many research opportunities are offered by using timed routes and timed route formulations. An interesting point to investigate is the various industrial rules that could only be developed for mathematical models based on timed routes. As already discussed and by definition, timed routes allow production flows and their cycle times to be explicitly modeled. On the opposite, flexible lead time constraints in a compact mathematical model do not easily allow cycle times to be limited and production flows to be explicitly managed. Hence, many relevant industrial constraints can be taken into account through timed routes. For example, timed routes could be generated by considering minimum or maximum cycle times of products, or minimum or maximum lead times between two non-consecutive production steps. Also, a cycle time for each product could be targeted in the objective function, by introducing new costs on timed routes instead of the somehow artificial WIP management costs. These costs could be associated with the deviation to the target cycle time. In addition, costs based on the duration of the lead time in a production step could be proposed, that would be non-linear in compact models but linear in timed route models.

Moreover, the computational times of the column generation approach could be accelerated by using smart column generation heuristics. Another research perspective is to consider initial inventories in the product routes. Shorter timed routes will be required to flush the initial inventories. Finally, we would like to study whether timed routes could be used in other contexts, e.g. when modeling product flows in supply chains where the notion of ”route” is also relevant.

Acknowledgments

This project has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737459 (project Productive4.0). This Joint Undertaking receives support from the European Unions Horizon 2020 research and innovation program and Germany, Austria, France, Czech Republic, Netherlands, Belgium, Spain, Greece, Sweden, Italy, Ireland, Poland, Hungary, Portugal, Denmark, Finland, Luxembourg, Norway, Turkey.

We would like to thank STMicroelectronics Crolles, which has graciously offered access to their data in the framework of the project Productive4.0.

References

Albey, E., Bilge, Ü., & Uzsoy, R. (2017). Multi-dimensional clearing functions for aggregate capacity modelling in multi-stage production systems. *International Journal of Production Research, 55*(14), 4164–4179.

Asmundsson, J., Rardin, R. L., & Uzsoy, R. (2006). Tractable nonlinear production planning models for semiconductor wafer fabrication facilities. *IEEE Transactions on Semiconductor Manufacturing, 19*(1), 95–111.

Azi, N., Gendreau, M., & Potvin, J.-Y. (2010). An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research, 202*(3), 756–763.

Bang, J.-Y., & Kim, Y.-D. (2010). Hierarchical production planning for semiconductor wafer fabrication based on linear programming and discrete-event simulation. *IEEE Transactions on Automation Science and Engineering, 7*(2), 326–336.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., & Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research, 46*(3), 316–329.

Billington, P. J., McClain, J. O., & Thomas, L. J. (1983). Mathematical programming approaches to capacity-constrained MRP systems: Review, formulation and problem reduction. *Management Science, 29*(10), 1126–1141.

Bredström, D., Lundgren, J. T., Rönnqvist, M., Carlsson, D., & Mason, A. (2004). Supply chain optimization in the pulp mill industry–ip models, column generation and novel constraint branches. *European Journal of Operational Research, 156*(1), 2–22.

Buschkühl, L., Sahling, F., Helber, S., & Tempelmeier, H. (2010). Dynamic capacitated lot-sizing problems: a classification and review of solution approaches. *OR Spectrum, 32*(2), 231–261.

Chen, M., Sarin, S., & Peake, A. (2010). Integrated lot sizing and dispatching in wafer fabrication. *Production Planning and Control, 21*(5), 485–495.

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.

Dantzig, G. B., & Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research, 8*(1), 101–111.

Dauzère-Pérès, S., & Lasserre, J.-B. (1994). Integration of lotsizing and scheduling decisions in a job-shop. *European Journal of Operational Research, 75*(2), 413–426.

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.

Desrosiers, J., & Lübbecke, M. E. (2005). A primer in column generation. In *Column generation* (pp. 1–32). Springer.

Gamache, M., Soumis, F., Marquis, G., & Desrosiers, J. (1999). A column generation approach for large-scale aircrew rostering problems. *Operations Research, 47*(2), 247–263.

Gómez Urrutia, E. D., Aggoune, R., & Dauzère-Pérès, S. (2014). Solving the integrated lot-sizing and job-shop scheduling problem. *International Journal of Production Research, 52*(17), 5236–5254.

Graves, S. C. (1986). A tactical planning model for a job shop. *Operations Research, 34*(4), 522–533.

Hung, Y.-F., & Leachman, R. C. (1996). A production planning methodology for semiconductor manufacturing based on iterative simulation and linear programming calculations. *IEEE Transactions on Semiconductor Manufacturing, 9*(2), 257–269.

Hwang, T.-K., & Chang, S.-C. (2003). Design of a lagrangian relaxation-based hierarchical production scheduling environment for semiconductor wafer fabrication. *IEEE Transactions on Robotics and Automation, 19*(4), 566–578.

Jampani, J., & Mason, S. J. (2010). A column generation heuristic for complex job shop multiple orders per job scheduling. *Computers & Industrial Engineering, 58*(1), 108–118.

Jans, R., & Degraeve, Z. (2004). An industrial extension of the discrete lot-sizing and scheduling problem. *IIE Transactions, 36*(1), 47–58.

Kacar, N. B., Mönch, L., & Uzsoy, R. (2016). Modeling cycle times in production planning models for wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing, 29*(2), 153–167.

Kim, S., & Uzsoy, R. (2008). Exact and heuristic procedures for capacity expansion problems with congestion. *IIE Transactions, 40*(12), 1185–1197.

Leachman, R. C., & Carmon, T. F. (1992). On capacity modeling for production planning with alternative machine types. *IIE Transactions, 24*(4), 62–72.

Liberatore, M. J., & Miller, T. (1985). A hierarchical production planning system. *Interfaces, 15*(4), 1–11.

Lopes, M. J. P., & de Carvalho, J. V. (2007). A branch-and-price algorithm for scheduling parallel machines with sequence dependent setup times. *European Journal of Operational Research, 176*(3), 1508–1527.

Manne, A. S. (1958). Programming of economic lot sizes. *Management Science, 4*(2), 115–135.

Missbauer, H. (2020). Order release planning by iterative simulation and linear programming: theoretical foundation and analysis of its shortcomings. *European Journal of Operational Research, 280*(2), 495–507.

Ng, T. S., Sun, Y., & Fowler, J. (2010). Semiconductor lot allocation using robust optimization. *European Journal of Operational Research, 205*(3), 557–570.

Nisted, L., Pisinger, D., & Altman, A. (2011). Optimal wafer cutting in shuttle layout problems. *Journal of Combinatorial Optimization, 22*(2), 202–216.

Wagner, H. M., & Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management Science, 5*(1), 89–96.

Yi, J., Jia, S.-j., Du, B., & Liu, Q. (2019). Multi-objective model and optimization algorithm based on column generation for continuous casting production planning. *Journal of Iron and Steel Research International, 26*(3), 242–250.