TERMINAL CONSTRAINTS FOR SINGLE-STAGE MULTI-PRODUCT SCHEDULING PROBLEMS

Yachao Dong, Christos T Maravelias*
University of Wisconsin - Madison
Madison, WI 53706

Abstract

In scheduling, it is important to maintain the terminal inventory levels above threshold values to prevent stockout. However, simply enforcing the terminal inventory level above a lead-time-based threshold for each product, as it is common practice, does not exploit the relationships of inventory levels among products. To overcome this drawback, we propose a new type of terminal constraints for single-stage problems, considering single- and multi-unit facilities. Replacing the traditional lead-time-based constraints with the proposed linear terminal constraints leads to better closed-loop solutions. Using different examples, we show how to obtain the terminal constraints, and we verify their effectiveness using instances with and without uncertainty.

Keywords

Closed-loop scheduling, Terminal constraints, Single-stage network.

Introduction

In supply chain (SC) management, lead-time-based thresholds of inventory levels can be required to prevent stockout, and it has been shown that they also play an important role in scheduling (Harjunkoski et al., 2014), especially in closed-loop scheduling (Subramanian et al., 2012; Gupta and Maravelias, 2016). In the optimal solution of a scheduling model with no constraints on terminal inventory levels, inventory tends to deteriorate at the end of horizon, so that the production, transition and inventory holding cost can be minimized (Lima et al., 2011). However, if such a solution is implemented, the problem may become infeasible, or the closed-loop solution may be very expensive, after the horizon is rolled forward. This is because the inventory is depleted at the end of the previous horizon, and therefore the demand cannot be satisfied. In the SC literature, a lead-time-based inventory threshold is generally required for each product at the end of the horizon, and is calculated based on the statistics of the lead time of SC arcs and demand rate of SC nodes (Eppen and Martin, 1988; Kreipl and Pinedo, 2004). In some scheduling literature, the terminal inventory levels are required to be equal to the initial value at the start of the horizon or to one of the values in a cyclic solution (Shah et al., 1993; Subramanian et al., 2012).

However, all of the aforementioned approaches neglect the relationship of inventory levels among products. For instance, in a single-stage, two-product network, if the inventory level of one product is high, a low inventory level of the other can possibly be acceptable, because we can allocate more resources for producing the latter without leading to the stockout of the former. By doing so, the inventory holding cost can be reduced significantly. Accordingly, in this work, we present methods to generate terminal constraints for single-stage networks, which account for the interrelationships among products. These constraints are linear, and can be easily incorporated in any mixed integer programming (MIP) scheduling model. The proposed constraints can also be used when uncertainty is

*Corresponding author: christos.maravelias@wisc.edu

considered (Vin and Ierapetritou, 2001; Guillén et al., 2006; You and Grossmann, 2008).

Theoretically, we can prove that for single-unit multi-product problems without uncertainty, adding the proposed terminal constraints leads to recursive feasibility (i.e., the scheduling problem remains feasible after we roll the horizon). The proof is not shown in this paper, due to the limited space.

Background

Problem Statement

The problem we consider can be defined in terms of the following sets:

- $i \in \mathbf{I}$: products, also denoting the corresponding tasks;
- $j \in \mathbf{J}$: units;
- $\mathbf{I}_j$: products which can be produced in unit j;
- $\mathbf{J}_i$: units which can be used to produce product i.

We study a simplified scheduling problem with constant demand rate. The normalized per-period demand of product i is denoted by δ_i. If task i can be processed in unit j, we use β_{ij} to denote its batch size, and τ_{ij} to denote the processing time. The inventory level of a product should be non-negative. We use S_i to denote the terminal inventory level of product i, and $\boldsymbol{s}$ to denote the vector of terminal inventory levels; i.e., $\boldsymbol{s} = [S_1, S_2, \dots, S_{|\mathbf{I}|}]^{\mathrm{T}}$.

The scheduling problem is solved through optimization in a rolling horizon approach; i.e., after the early part of the solution is implemented, the horizon is rolled forward, and the scheduling problem with new information is re-optimized. We want to study how to constrain the terminal inventory levels in the optimization model so that we can (i) ensure recursive feasibility, and (ii) decrease the inventory levels compared to the traditional lead-time-based threshold constraints.

Feasibility Model

We first need to identify the *region of feasible terminal inventory levels*, denoted by $\mathbf{S}^{\mathrm{F}} \subseteq \mathbb{R}^{|\mathbf{I}|}$. Each $\boldsymbol{s} \in \mathbf{S}^{\mathrm{F}}$ satisfies that when $\boldsymbol{s}$ is the value of the terminal inventory levels, the scheduling problem will remain feasible in the rolling horizon approach. To obtain region $\mathbf{S}^{\mathrm{F}}$, we solve a feasibility scheduling model **MF** repeatedly.

In model **MF**, $\boldsymbol{s}$ is used as a given parameter to denote the *initial* inventory levels. If model **MF** is feasible, a scheduling solution whose *terminal* inventory level is equal to the given $\boldsymbol{s}$ will lead to recursive feasibility (Figure 1); i.e., $\boldsymbol{s} \in \mathbf{S}^{\mathrm{F}}$. By enumerating $\boldsymbol{s}$ and solving **MF** iteratively, we can obtain region $\mathbf{S}^{\mathrm{F}}$ (shown in orange in Figure 2).

Model **MF**, involving task-unit assignment and timing decisions, is as follows,

Minimize: $$\sum_{i,j\in \mathbf{J}_i,t} \alpha_{ij} W_{ijt} + \sum_{i,t} \pi_i L_{it} \quad (1)$$

Subject to: $$L_{i,t+1} = L_{it} + \sum_{j\in \mathbf{J}_i} \beta_{ij} W_{ij,t-\tau_{ij}+1} - \delta_i, \forall i, t \quad (2)$$

$$L_{i,0} = S_i - \delta_i, \forall i \quad (3)$$

$$\sum_{i\in \mathbf{I}_j, t-\tau_{ij}+1 \le t' \le t} W_{ijt'} \le 1, \forall j, t \quad (4)$$

where $t \in \{0,1,\dots,T\}$ is the index for time points; variable $L_{it} \in \mathbb{R}^+$ denotes the inventory level of product i at time t, and $W_{ijt} \in \{0,1\}$ is 1 if task i starts in unit j at time t. The objective, as shown in Eq. (1), is to minimize production cost and inventory holding cost, where α_{ij} is the production cost of task i in unit j, and π_i is the inventory holding cost of product i. Material balance is expressed in Eq. (2) and (3). At each time, only one task can be processed in a certain unit, which is enforced by Eq. (4).

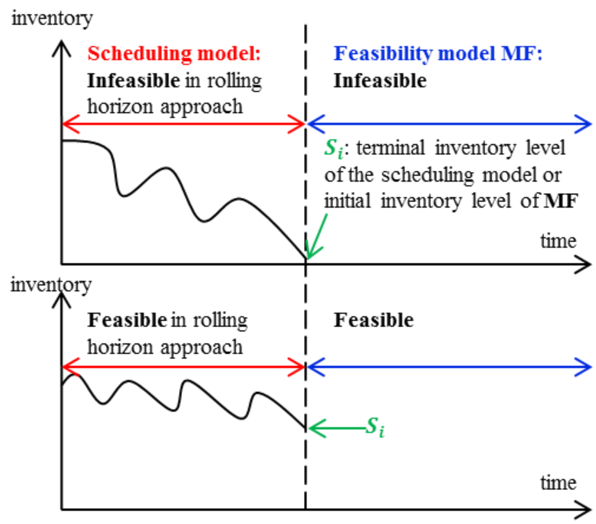

Figure 1. To check if $\boldsymbol{s} \in \mathbf{S}^F$, we solve model **MF** *with given initial inventory levels*

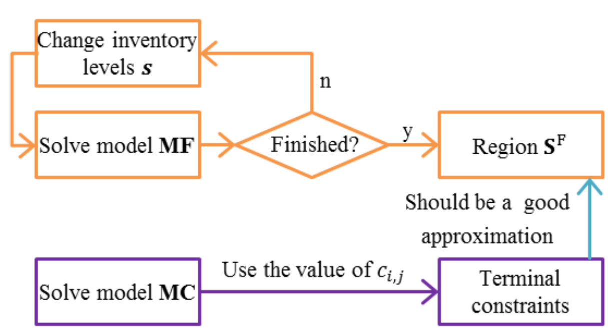

Figure 2. Methods to obtain region $\mathbf{S}^F$ and terminal constraints

Region $\mathbf{S}^{\mathrm{F}}$, obtained by solving **MF** repeatedly, might be non-convex, and therefore should be approximated by a set of linear terminal constraints. Good terminal constraints should lead to a close approximation of region $\mathbf{S}^{\mathrm{F}}$.

Campaign Model

Before writing the terminal constraints, we need to find how frequently each task is being carried out in a "typical" scheduling solution. We do so by solving an auxiliary linear programming (LP) model **MC**. The model determines how many times a task should be processed in a unit in the campaign mode. Variable $c_{ij} \in \mathbb{R}^+$ in model **MC** denotes the number of batches that task *i* is processed in unit *j*. The value of c_{ij} is to be used as a parameter when writing the terminal constraints (shown in purple in Figure 2). Model **MC** is as follows,

$$\text{Minimize:} \quad \sum_{i,j \in \mathbf{J}_i} \frac{\beta_{ij}}{\delta_i} c_{ij} \tag{5}$$

$$\text{Subject to:} \quad h \geq \sum_{i \in \mathbf{I}_j} \tau_{ij} c_{ij}, \forall j \tag{6}$$

$$\sum_{j \in \mathbf{J}_i} \beta_{ij} c_{ij} \geq \delta_i h, \forall i \tag{7}$$

$$\sum_{j \in \mathbf{J}_i} c_{ij} \geq 1, \forall i \tag{8}$$

where variable $h \in \mathbb{R}^+$ represents the campaign time. The objective function in Eq. (5) minimizes the total production in a campaign. In Eq. (6), the campaign time is required to be greater than the total production time for each unit. The production amount should be greater than the demand, as shown in Eq. (7). To avoid the trivial solution in which all variables are zero, Eq. (8) requires that each task is processed at least once. Because the values of c_{ij} appear linearly on both sides of the terminal constraints (presented in the next two sections), it is their relative ratios that are important, and therefore variables c_{ij} are defined to be continuous, rather than integer.

We assume that the scheduling problem is feasible (i.e., when the initial inventories are enough, there is always a schedule in which production can meet demand). This LP model yields the number of batches in different units to meet demand.

Terminal Constraints for Single-unit Problems

Since we consider only one unit here, the unit index *j* is dropped. Starting from a case of 2 tasks (products), the MIP model **MF** is feasible if inventory levels S_i satisfy the following constraints:

$$\frac{S_1}{\delta_1} \geq \tau_1 \tag{9}$$

$$\frac{S_2}{\delta_2} \geq \tau_2 \tag{10}$$

$$\frac{c_1 \tau_1 S_1}{\delta_1} + \frac{c_2 \tau_2 S_2}{\delta_2} \geq (c_1 \tau_1 + c_2 \tau_2)(\tau_1 + \tau_2) \tag{11}$$

where c_1 and c_2 are obtained by solving the model **MC**.

More generally, the terminal constraints can be written as,

$$\sum_{i \in \mathbf{I}_p} \frac{c_i \tau_i S_i}{\delta_i} \geq \left(\sum_{i \in \mathbf{I}_p} c_i \tau_i \right) \cdot \left(\sum_{i \in \mathbf{I}_p} \tau_i \right), \forall p \in \mathbf{P}(\mathbf{I}) \tag{12}$$

where $\mathbf{P}(\mathbf{I})$ denotes the power set of **I** (i.e., the set of all subsets of **I**) except the empty set, indexed by *p*; $\mathbf{I}_p$ denotes the products that are included in the subset *p*. Because Eq. (12) is written for each subset of **I** except the empty set, the total number of constraints is $2^{|\mathbf{I}|} - 1$.

We can prove that if inventory levels S_i satisfy Eq. (12), model **MF** will be feasible regardless of horizon length. Thus, by enforcing this type of terminal constraints, we ensure that there is always a feasible solution for the next scheduling problem after rolling the horizon.

The constraints in Eq. (12) can be categorized into |**I**| groups, denoted by group 1, group 2, etc., according to the number of products included in subset *p*. In other words, group *k* consists of the constraints with $|\mathbf{I}_p| = k$. For example, there are |**I**| constraints for group 1, as follows,

$$\frac{S_i}{\delta_i} \geq \tau_i, \forall i \tag{13}$$

To better understand Eq. (12), we define $t_i = S_i / \delta_i$, denoting the number of periods for which the inventory itself can meet the demand, and t_i is henceforth referred to as normalized inventory. Constraints of group 1 can be written as,

$$t_i \geq \tau_i, \forall i \tag{14}$$

which means that for each product, the normalized inventory should be greater than or equal to the processing time, so that the inventory is sufficient to last during the execution of the first batch.

For the other groups, we define $\rho_i = c_i \tau_i$ and $\rho_p = \sum_{i \in \mathbf{I}_p} c_i \tau_i$, denoting the production time of a task and the total production time of a subset of tasks. Eq. (12) can be rewritten as follows,

$$\sum_{i \in \mathbf{I}_p} \left(\frac{\rho_i}{\rho_p} t_i\right) \geq \sum_{i \in \mathbf{I}_p} \tau_i, \forall p \in \mathbf{P}(\mathbf{I}) \tag{15}$$

If we view *p*, the subset of tasks, as a pseudo-task, Eq. (15) can be interpreted as a generalization of Eq. (14). On the right hand side (RHS) is the processing time of the pseudo-task, which is the summation of processing time of all tasks in the subset. On the left hand side is the normalized inventory of the pseudo-task; the weight ρ_i / ρ_p is the ratio of the production time of task *i* to the production time of all the tasks in subset *p*. If task *i* has a longer production time, the initial inventory of task *i* plays a more important role in the pseudo-task, and thus the weight is higher.

Terminal Constraints for Multi-unit Problems

If *all units are identical*, there is a solution of **MC** with $c_{ij} = c_{ij'}$ for all *i, j, j'*. Thus, we can drop index *j* again. Model **MF** with initial inventory levels satisfying Eq. (12) is still guaranteed to be feasible; when all the units are synchronized to carry out the same task, the inventory profile will be the same as that in the single-unit case.

However, because units are not required to be synchronized (i.e., we have more flexibility with multiple units), Eq. (12) is too conservative. In order to make the feasible region defined by the terminal constraints larger, which would lead to lower inventory levels and lower cost, we relax some of the constraints. At the same time, we make sure that very few, if any, values of $\boldsymbol{s}$ that do not belong to region $\mathbf{S}^{\mathrm{F}}$ will be included in the feasible region defined by the relaxed terminal constraints. The terminal constraints are modified as follows,

$$\sum_{i\in\mathbf{I}_p}\frac{c_i\tau_i S_i}{\delta_i} \geq \sum_{i\in\mathbf{I}_p} c_i{\tau_i}^2 + \mu \sum_{i\in\mathbf{I}_p, i'\in\mathbf{I}_p : i'\neq i} c_i\tau_i\tau_{i'}, \quad \forall p\in\mathbf{P}(\mathbf{I}) \tag{16}$$

where μ is a pre-defined parameter between 0 and 1. When μ is 1, Eq. (16) reduces to Eq. (12), and no relaxation is performed; we suggest to use $\mu = 1/|\mathbf{J}|$, because our empirical study shows that the feasible region defined by such terminal constraints is a good approximation of the region $\mathbf{S}^{\mathrm{F}}$.

If *units are non-identical*, the exact constraints of group 1 can be written as follows,

$$S_i \geq -\min_{0\leq l\leq \max_j \tau_{ij}-1}\left\{-\delta_i - l\delta_i + \sum_{j\in\mathbf{J}_i}\left\lfloor\frac{l}{\tau_{ij}}\right\rfloor\beta_{ij}\right\}, \forall i \tag{17}$$

The RHS represents the maximum backlog of product *i*, if its initial inventory is zero and its production is started in all units at time 0. If units are identical, Eq. (17) reduces to Eq. (13).

The constraints for the other groups are harder to write, because processing times and batch sizes can vary among units. Herein, we introduce the *average* parameters for each task *i*, (index *j* is again dropped,) as follows,

$$c_i = \sum_{j\in\mathbf{J}_i} c_{ij} \tag{18}$$

$$\tau_i = \frac{1}{c_i}\sum_{j\in\mathbf{J}_i} c_{ij}\tau_{ij} \tag{19}$$

Using these average parameters, the terminal constraints can be written as in Eq. (16) for other groups.

Remarks

First, we compare the proposed terminal constraints with the traditional approach. For the single-stage multi-product problem, the lead-time-based threshold constraints (Eppen and Martin, 1988) are as follows,

$$S_i \geq \delta_i \cdot \max_{j\in\mathbf{J}_i}\left(\sum_{i'\in\mathbf{I}_j}\tau_{i'j}\right), \forall i \tag{20}$$

The region constrained by Eq. (20) is included in the region constrained by the proposed terminal constraints. We illustrate this through a 2-product example (Figure 3). When the problem is a single-unit case, the distance *r* in Figure 3 is zero; otherwise, $r \geq 0$. This implies that compared to the lead-time-based threshold constraints, the terminal constraints can decrease the inventory levels and thus the inventory holding cost.

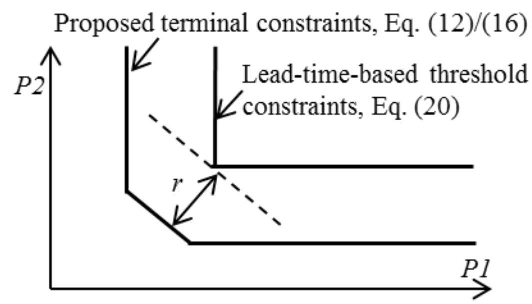

Figure 3. The proposed terminal constraints and the constraints of traditional approach

Second, when uncertainty is considered, a buffer term should be added. The main sources of uncertainty include the processing time and the demand. If the mean and variance of the processing time and the normalized demand can be characterized (denoted by $\bar{\tau}_i$, $\bar{\delta}_i$, $\sigma^2(\tau_i)$ and $\sigma^2(\delta_i)$ respectively), a buffer term Z_p defined below is added to the RHS of Eq. (12) and (16),

$$Z_p = \phi\sum_{i\in\mathbf{I}_p}\frac{c_i\tau_i\sqrt{\sigma^2(\delta_i)\bar{\tau}_i + (\bar{\delta}_i)^2\sigma^2(\tau_i)}}{\bar{\delta}_i} \tag{21}$$

in which ϕ is the inverse distribution function of a standard normal distribution based on a specified service level (Eppen and Martin, 1988).

Examples

Sinlge-unit

First, we consider a 2-product example (Table 1). Solving **MC**, we obtain c_{P1}=2, c_{P2}=1. From Eq. (12), the terminal constraints are:

$$S_{P1} \geq 4 \tag{22}$$

$$S_{P2} \geq 3 \tag{23}$$

$$2S_{P1} + 3S_{P2} \geq 35 \tag{24}$$

Region $\mathbf{S}^{\mathrm{F}}$, obtained by repeatedly solving **MF** with different initial inventory levels, is shown in Figure 4; we also show the proposed terminal constraints defined in Eq. (22)-(24), as well as the terminal constraints using other value of c_i (c_{P1}=2.571, c_{P2}=1, obtained by solving **MC** with a different objective function), which is feasible but not optimal for model **MC**. In this figure, the feasible region defined by Eq. (22)–(24) is included in region $\mathbf{S}^{\mathrm{F}}$. We also note that any c_i that is feasible for **MC** can be used to generate the terminal constraints, and neither the optimal c_i from **MC** nor the other feasible c_i leads to a better approximation.

Table 1. Data of the single-unit 2-product example

product (i)	*P1*	*P2*
processing time (τ_i)	2	3
batch size (β_i)	7	9
normalized demand (δ_i)	2	1

Second, we consider an example with three products *P1*, *P2*, *P3*, and the parameters are τ_i=2, β_i=12, δ_i=2. We obtain c_{P1}=c_{P2}=c_{P3}=1. Region $\mathbf{S}^{\mathrm{F}}$ and the proposed terminal constraints are shown in 3D-plots in Figure 5; using the terminal constraints, we approximate the non-convex region $\mathbf{S}^{\mathrm{F}}$ by a convex region, and the latter is included in the former.

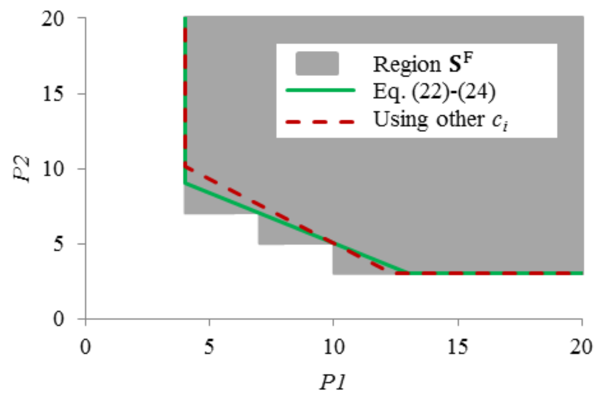

Figure 4. Region $\boldsymbol{S}^F$ and terminal constraints for single-unit 2-product example

Multi-unit

We consider an example with 2 products and 2 identical units; data in Table 1 are used, except that the normalized demand is doubled. The solution of **MC** leads to c_{P1}=1, c_{P2}=0.5. Region $\mathbf{S}^{\mathrm{F}}$ and the feasible regions defined by terminal constraints in Eq. (16) using different values of parameter μ are shown in Figure 6. When we use μ =1/|**J**| (i.e., μ=0.5 in this example), the feasible region defined by the terminal constraints is a better approximation of region $\mathbf{S}^{\mathrm{F}}$, compared to that using μ=1.

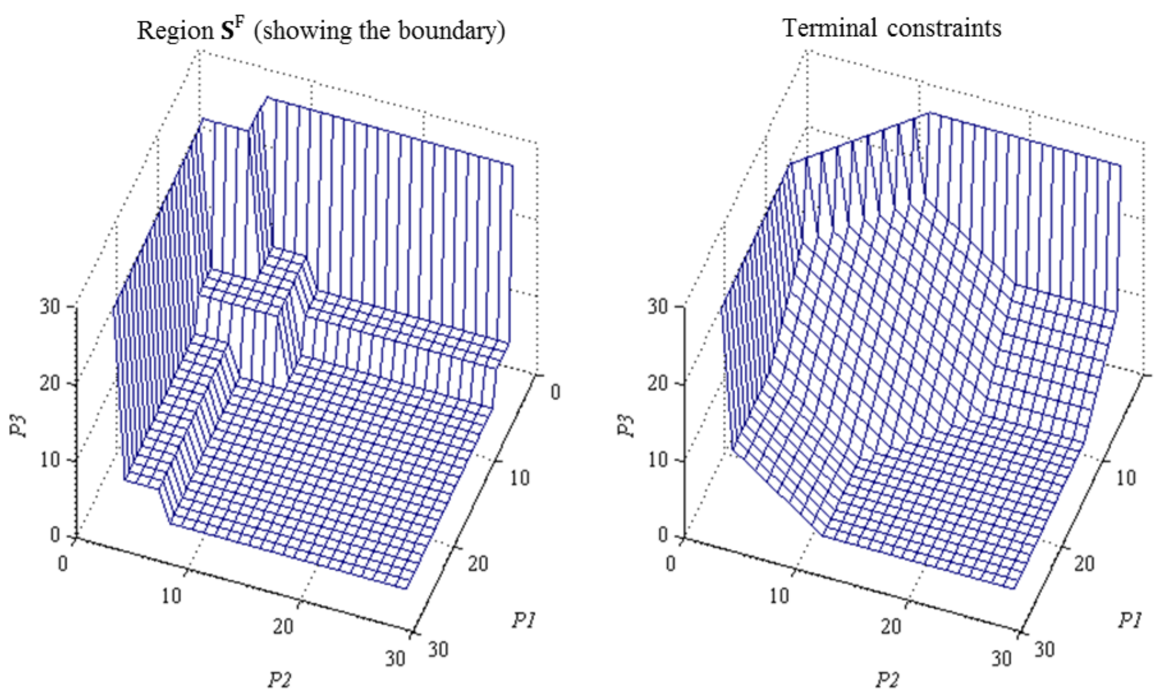

Figure 5. Region $\boldsymbol{S}^F$ and terminal constraints for single-unit 3-product example

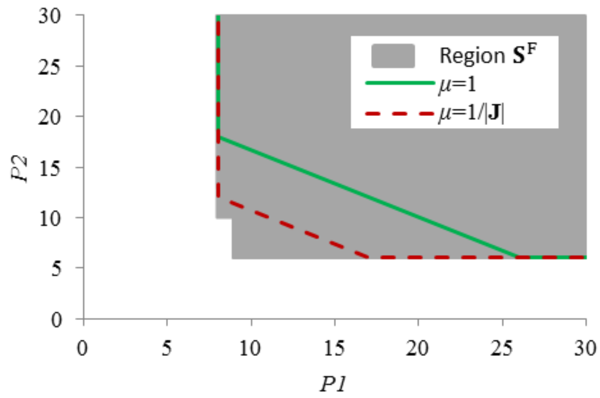

Figure 6. Region $\boldsymbol{S}^F$ *and terminal constraints for multi-unit example*

Rolling Horizon Implementation

For simplicity, we use **MF** as the scheduling model. First, we consider the case without uncertainty. Using the single-unit 2-product instance (Table 1), we choose a horizon of 12 periods, and the initial inventory levels are 16 and 8 respectively for *P1* and *P2*. After obtaining a solution, we roll the horizon forward by 12 periods, and solve **MF** of the new horizon. Model **MF** is solved for 6 iterations, and therefore the schedule from time 0 to time 72 is obtained. When no terminal constraints are applied, model **MF** becomes infeasible after one iteration; by introducing slack variables allowing negative inventory levels, we observe that 23 out of the 72 periods were subject to stockout. When adding the terminal constraints, model **MF** remains feasible for all 6 iterations (i.e., no stockout was incurred).

Using the same instance, we evaluate the terminal constraints under uncertainty. The demand in each period is subject to a normal distribution, $\mathcal{N}(\delta_i, (0.3\delta_i)^2)$, with δ_i given in Table 1. In each iteration, we solve the deterministic model, observe the uncertain demand of the first period in the model, and roll the horizon forward by 1 period. We compare the solutions using the terminal constraints to those using the traditional lead-time-based threshold constraints; i.e., model **MF** with Eq. (12) was compared to **MF** with Eq. (20). Due to the uncertainty, inventory buffers were added in the model. First, L_{it} was required to be greater than $1.6 \cdot 0.3\delta_i \cdot \tau_i$, instead of simply being non-negative; 1.6 is the value of parameter ϕ used in Eq. (21), at a service level of 95%. Second, in Eq. (12) and (20), the same buffer of $1.6 \cdot 0.3\delta_i \cdot \tau_i$ was added to variable S_i.

We obtained 200 samples for both models (with Eq.(12) and with Eq. (20)). For each sample, **MF** was repeatedly solved for 72 iterations to obtain the closed-loop schedule from time 0 to time 72. In Table 2, we show the average inventory levels (considering both products) and the average number of periods that were subject to stockout. The proposed terminal constraints reduced inventories by 19.6% on average compared to the lead-time-based threshold constraints, while both methods led to very rare stockout (less than 0.4%).

Table 2. Results comparing the proposed terminal constraints Eq. (12) with the traditional constraints Eq. (20)

	inventory levels	stockout periods
Eq. (12)	13.58	0.33
Eq. (20)	16.90	0.32

Conclusions

We proposed a novel type of terminal constraints for single-stage production scheduling problems. The proposed constraints consider the relationship among products and lead to recursive feasibility when scheduling is implemented in a rolling horizon manner. In problems under uncertainty, the proposed terminal constraints can achieve substantial savings on inventory holding cost by lowering the inventory levels, compared to the traditional lead-time-based threshold constraints.

References

Eppen, G. D., Martin, R. K. (1988). Determining safety stock in the presence of stochastic lead time and demand. *Manag. Sci.*, *34(11)*, 1380-1390.

Guillén, G., Espuña, A., Puigjaner, L. (2006). Addressing the scheduling of chemical supply chains under demand uncertainty. *AIChE J.*, *52(11)*, 3864-3881.

Gupta, D., Maravelias, C. T. (2016). On deterministic online scheduling: major considerations, paradoxes and remedies. *Comput. Chem. Eng.*, *94*, 312-330.

Harjunkoski, I., Maravelias, C.T., Bongers, P., Castro, P.M., Engell, S., Grossmann, I.E., Hooker, J., Méndez, C., Sand, G., Wassick, J. (2014). Scope for industrial applications of production scheduling models and solution methods. *Comput.Chem. Eng.*, *62*, 161–193.

Kreipl, S., Pinedo, M. (2004). Planning and scheduling in supply chains: an overview of issues in practice. *Prod. Oper. Manag.*, *13(1)*, 77-92.

Lima, R. M., Grossmann, I. E., Jiao, Y. (2011). Long-term scheduling of a single-unit multi-product continuous process to manufacture high performance glass. *Comput. Chem. Eng.*, *35(3)*, 554−574.

Shah, N., Pantelides, C. C., Sargent, R. W. H. (1993). Optimal periodic scheduling of multipurpose batch plants. *Ann. Oper. Res.*, *42(1)*, 193-228.

Subramanian, K., Maravelias, C. T., Rawlings, J. B. (2012). A state-space model for chemical production scheduling. *Comput. Chem. Eng.*, *47*, 97-110.

Vin, J.P., Ierapetritou, M.G. (2001). Robust short-term scheduling of multiproduct batch plants under demand uncertainty. *Ind. Eng. Chem. Res.*, *40(21)*, 4543-4554.

You, F., Grossmann, I.E. (2008). Design of responsive supply chains under demand uncertainty. *Comput. Chem. Eng.*, *32(12)*, 3090-3111.