AIChE

Adaptive Model Predictive Inventory Controller for Multiproduct Batch Plant

Gyeongbeom Yi
Dept. of Chemical Engineering, Pukyong National University, San 100, Yongdang-Dong, Nam-Ku, Busan, Korea 608-739

Gintaras V. Reklaitis
School of Chemical Engineering, Purdue University, West Lafayette, IN 47907

DOI 10.1002/aic.14783
Published online March 25, 2015 in Wiley Online Library (wileyonlinelibrary.com)

An inventory control system was developed for multiproduct batch plants with an arbitrary number of batch processes and storage units. Customer orders are received by the plant at order intervals and in order quantities that are subject to random fluctuations. The objective of the plant operation is to minimize the total cost while maintaining inventory levels within the storage or warehouse capacity by adjusting the startup times, the quantities of raw material orders, and production batch sizes. An adaptive model predictive control algorithm was developed that uses a periodic square wave model to represent the flows of the material between the processes and the storage units. The boundedness of the control output and the convergence of the estimated parameters in implementations of the proposed algorithm were mathematically proven under the assumption that disturbances in the orders are bounded. The effectiveness of this approach was demonstrated by performing simulations. *AIChE J*, 61: 1867–1880, 2015
Keywords: adaptive model predictive control, multiproduct batch plant, periodic square wave model, batch-storage network

Introduction

In recent years, the operational emphasis of chemical plants has been on providing fast and timely customer response while maintaining the lowest possible inventory levels, which shifts to the production sites, the burden of efficient small lot production with rapid product changeovers in response to highly uncertain demand fluctuations. The need for a reliable supply chain management system has increased significantly. This study aimed to facilitate the simultaneous automation and optimization of the scheduling of multiproduct batch plants that consist of multiple batch processes and storage or warehouse units connected each other according to feedstock compositions and product yields, and produce multiple products from multiple raw materials.[1] This type of plant is used in many sectors of the processing industry, such as those that manufacture packaged lubricants, pelletized polymer products, cosmetics, and other consumer products. Note that this methodology can be extended to multitasking batch or semicontinuous plants[2] which belong to the multipurpose batch plant category, but, for simplicity, this study only discusses multiproduct batch plants. The simultaneous automation and optimization of the scheduling of supply chain systems has been investigated using two principal research approaches: reactive scheduling[3,4] as emphasized by optimization-oriented researchers; and model predictive control (MPC)[5,6] as emphasized by control-oriented researchers. Reactive scheduling uses a full scale optimization model and repeatedly revises the schedule in response to uncertain future demand forecasts and input data in an approach that is similar to the offline operation of feedback control mechanism. This methodology is currently used in actual plants and all schedulers do it in one way or another. The drawback of reactive scheduling is that long-term stability is not guaranteed as it is difficult to implement stabilizing mechanisms in scheduling optimization models. In contrast, the MPC approach typically uses a linear system model known as the state-space model, which does provide theoretical approaches to ensuring system stability but cannot account for scheduling details that do not fit into a linear system.[7] Most MPC models for supply chain systems use inventory replenishing equations with some subsidiary constraints and the supply chain system structure is limited to simple forms such as serial or parallel systems. Recent research has shown that a large-scale multiperiod mixed integer linear programming (MILP) scheduling model can be transformed into a state-space model suitable for control system design.[8] This conclusion suggests that both the optimality and stability of scheduling can be pursued simultaneously. However, the multiperiod MILP scheduling model produces a lumped solution for each period that cannot be directly applied to real plants because the period interval is

Correspondence concerning this article should be addressed to G. Yi at gbyi@pknu.ac.kr.

not sufficiently short to provide continuity. This limitation can be overcome using a continuous-time scheduling model as usually used in reactive scheduling[3]; however, this approach has not yet been reported for the state-space framework. Another drawback of such scheduling models is known as the End-of-World problem in the field: the models have a finite time horizon, so the final inventory levels of the optimal solutions become zeroes or decrease to assigned minimum values, which is obviously not optimal for a long-term horizon. It can easily be shown that this solution is meaningless if the scheduling horizon is insufficiently long. Suppose that the uncertain ending inventory volume is half the monthly sales volume and that the raw material costs amount to 70% of the total sale revenue. Then, the uncertainty of the ending inventory is 3% of the sales revenue. The optimization model becomes useful when it produces savings that are significantly more than 3% of the sales revenue. Considering that most commercialized technologies such as advanced process control and/or real-time optimization offer a yearly benefit less than 5%, this End-of-World problem needs to be resolved carefully. The current approaches to this problem include: (1) estimate the final inventory using a statistical simulation,[9] (2) estimate the salvage value of the final inventory,[10] (3) create a separate long-term planning model that provides inventory targets to the short-term scheduling model,[11] and (4) use a cyclic scheduling model to create an infinite horizon. The cyclic scheduling approach is probably the most appealing; however, it requires nonconvex mixed integer nonlinear programming, which is difficult to implement. Recently, a reformulation-linearization method for the transformation of the nonlinear cyclic scheduling model into a linear model has been developed.[12] The application of this linear model to MPC has not yet been reported.

Another cyclic model for supply chain system is the periodic square wave (PSW) model, which is applicable to generalized batch-storage networks (GBSNs).[13,14] A PSW model represents periodic square shaped material flows between a storage unit and a batch process. A GBSN is a general multistage production/transportation and inventory system that accounts for financial flows and the recycling of material flows. A GBSN can be used to model most supply chain components such as raw material purchases, production, transportation, and customer demand. The drawback of GBSNs with respect to other models such as state-task networks is that their storage units are dedicated to only one material. The optimization of a GBSN using the PSW model requires a nonconvex nonlinear programming; however, an analytic solution for lot sizing equations is available and this approach has a special structure that enables easier computation. The PSW model used in GBSN optimization can also be used to design MPC systems for supply chain facilities, such as single product storage system[15] or single multiproduct warehouse system.[16] In this study, we extended a previous study to describe batch-storage networks (BSNs) that do not include transportation processes other than single product delivery and are thus suitable for multiproduct batch plants. Of course, the MPC design method of this study can also be applied to GBSNs. The final goal of this long-term project is the development of an MPC system for a multinational supply chain network consisting of production/transportation processes and material/currency inventory units, including multiple currency/material flows.

In this study, we address a particular form of such networks in which a single multiproduct batch plant provides multiple products to a set of customers and purchases raw materials from multiple vendors. Each of the processes operates under a periodic schedule with regular lot sizes that can be adjusted as required. Purchasing lots are delivered to the plant at fixed intervals using regularly scheduled transport vehicles with delivery lead times. Customer orders occur with a fixed order interval and in regular order quantities; however, both are subject to random fluctuations. We seek an operational state in which the product inventories are maintained within specified upper and lower levels (set band), so that a minimum level of safety stock is always available, by varying the purchasing/producing lot sizes and intervals so as to meet customer requirements. This demand-driven system can be characterized as a discrete process control problem involving discrete material flows. In the proposed feedback control framework, deviations from the expected cycle times and order sizes are used to adjust the startup times and lot sizes of the purchase/production flows so as to ensure that the inventory level of each product neither exceeds the storage/warehouse capacity nor falls below a minimum safety stock level. The system uses an MPC algorithm based on a detailed nonlinear inventory prediction model derived from the basic material balance equations with best fitting PSW material flows approximating random discrete flows. The adaptive control algorithm combines a stabilized minimum variance type control input calculation with suitable predictions of the input and output streams. The optimization model for scheduling used in this study includes constraints on the upper limits of the inventory levels, which were not present in previous research.[1] The boundedness of the system outputs and the convergence of the estimated parameters for given bounded random disturbances with respect to customer demands are mathematically proven.

The remainder of this article is organized as follows. First, the optimization model, solution technique, control law, and parameter estimation scheme are described. The boundedness of the control output is mathematically proven. Finally, some computational results that highlight the advantages of the proposed approach are provided and conclusions are offered. The performance of the suggested algorithm with respect to long lead times is evaluated and simulation examples are discussed in detail. In this study, we assume for simplicity that there are no uncertainties in the vendors' delivery and lead times.

System Description

A general multiproduct batch plant can be represented by the BSN[1] shown in Figure 1. The BSN involves multiple materials (circles) $j=1,2,...,|J|$, multiple warehouses $w=1,2,...,|W|$, customers (outbound arrows) $m=1,2,...,|M|$, raw material vendors (inbound arrows) $k=1,2,...,|K|$, and batch processes (squares) $i=1,2,...,|I|$. Where $|J|$ is the number of elements in material set J, $|W|$ is the number of elements in warehouse set W, $|M|$ is the number of elements in customer set M, $|K|$ is the number of elements in vendor set K, and $|I|$ is the number of elements in process set I. Each warehouse w stores one or more than one material $j \in J(w)$. Each process requires multiple feedstock materials of fixed composition (f_i^j) and produces multiple products with fixed product yield (g_i^j). Assuming that the material

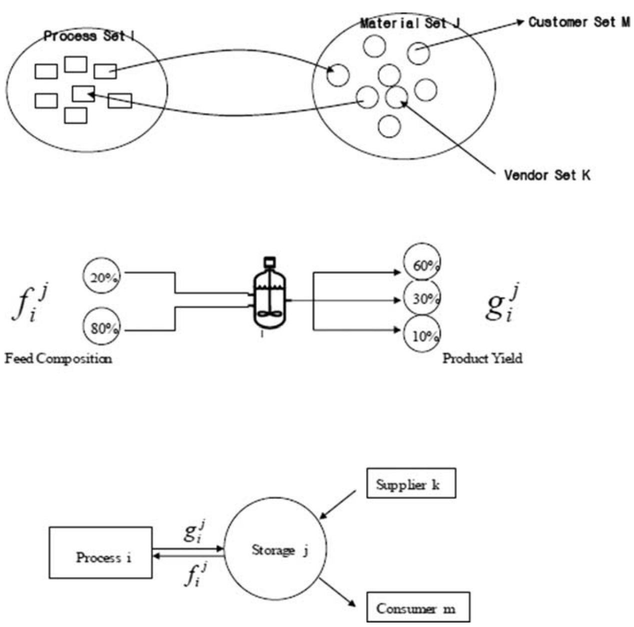

Figure 1. General structure of BSN.

flows can be described with PSWs, the flows can be expressed in terms of four variables: the startup time, the lot size, the cycle time, and the storage operation time fraction (SOTF).[1] The disturbances transmitted through the customer demand flows can thus be characterized with the following three equations

$$\mathbf{B}_m^j=\overline{B_m^j}+\mathbf{e}_{\mathbf{B}_m^j} \tag{1}$$

$$\boldsymbol{\omega}_m=\overline{\omega_m}+\mathbf{e}_{\boldsymbol{\omega}_m} \tag{2}$$

$$\mathbf{x}_m^j=\overline{x_m^j}+\mathbf{e}_{\mathbf{x}_m^j} \tag{3}$$

where the bar denotes the expectation operator and the bold character denotes a random variable. The quantities $\mathbf{e}_{\mathbf{B}_m^j}$, $\mathbf{e}_{\boldsymbol{\omega}_m}$, and $\mathbf{e}_{\mathbf{x}_m}$ are noises, $\mathbf{B}_m^j$ is the size of the order for product j from customer m, $\boldsymbol{\omega}_m$ is the cycle time of the order of customer m, and $\mathbf{x}_m^j$ (≤ 1) is the SOTF for the order from customer m. The startup time occurs once and we will not consider it as a random variable. The intensity of $\mathbf{e}_{\mathbf{x}_m}$ is assumed to be relatively small and $\mathbf{e}_{\mathbf{x}_m}$ is assumed to be zero, for simplicity

$$\mathbf{e}_{\mathbf{x}_m}=0 \tag{4}$$

Figure 2 provides the definitions of the variables and parameters used to describe the vendor supply and customer demand PSW flows. The incoming flow into a batch process is similar to the vendor supply flow and the outgoing flow from the batch process is similar to the customer flow, both of which are not shown in this figure. Any of the order sizes, startup times, or SOTFs of the flows can be used as control variables. From a practical point of view, the SOTFs are not the best control variables and were assumed to be constant throughout this study. Two types of control inputs are included here: the order/production startup times (t_k^j and $\grave{t}_i$) and the order/production lot sizes (B_k^j and B_i). The cycle times for raw material orders and batch production, ω_k^j and ω_i, respectively, are actually determined by multiple startup time decisions. The average flow rates of material j from vendor k and that of process i (D_k^j and D_i) can be used as control inputs instead of the lot sizes, as

$$B_k^j=D_k^j\omega_k^j \text{ and } B_i=D_i\omega_i \tag{5}$$

The cycle times have the minimum value $\underline{\underline{\omega_i}}$, that is, $\omega_i \geq \underline{\underline{\omega_i}}$ and $\underline{\underline{\omega_k^j}}$ and $\omega_k^j \geq \underline{\underline{\omega_k^j}}$. The slack times $\overline{\overline{\omega_i}}-\underline{\underline{\omega_i}}$ and $\omega_k^j-\underline{\underline{\omega_k^j}}$ are used to compute startup times for control purposes. The above choices of control variables and parameters have been successfully used to describe a single product intermediate storage system[15] and a multiproduct warehouse system.[16] The definitions of the design variables are shown graphically in Figure 2. The inventory levels and flows are assumed to be measurable. The system parameters $\overline{B_m^j}$, $\overline{\omega_m^j}$, x_m^j, x_k^j, $\grave{x}_i$, and x_i' are treated as known constants, but their values can be updated with an adaptive mechanism, where $\grave{x}_i$ and x_i' are the SOTFs for the feeding flow into process i and the discharging flow from process i, respectively. We also consider the lead time of each raw material purchase, $\tilde{t}_k^j$, which is a random parameter in the real world, but for simplicity did not account for such randomness in this study. The production processes and customer demands also have lead times but we will not consider them, for simplicity. The main control objective of the BSN system is that the inventory levels should not exceed the upper limits of the warehouse's capacity to store one or multiple materials and that the inventory level of each material should not be depleted below its safety stock level

$$\sum_{j\in J(w)}\overline{\overline{V^j}} \leq \overline{\overline{S^w}} \tag{6}$$

$$\underline{\underline{V^j}} \geq \underline{\underline{S^j}} \tag{7}$$

Here, $\overline{\overline{V^j}}$ and $\underline{\underline{V^j}}$ are the upper and lower bounds, respectively, for the inventory level of material j, $\overline{\overline{S^w}}$ is the total capacity of the warehouse to store one or multiple materials $j \in J(w)$, and $\underline{\underline{S^j}}$ is the safety stock level for material j. From an economic perspective, the objective of inventory control is to minimize the total costs associated with the purchasing/production setup, the inventory holding, the capital costs of the warehouse/purchasing/production facilities, and raw material prices. Other cost factors such as the interest rate, taxes, the exchange rate, and so forth, can also be considered;[13] however, for the simplicity, we omitted these factors. The total cost TC (\$/day) is

$$\mathrm{TC}=\sum_{j=1}^{|J|}\sum_{k=1}^{|K(j)|}\left[\frac{A_k^j}{\omega_k^j}+a_k^jD_k^j\omega_k^j+P_k^jD_k^j\right]+\sum_{i=1}^{|I|}\left[\frac{A_i}{\omega_i}+a_i\omega_i\sum_{j=1}^{|J|}f_i^jD_i\right]+\sum_{j=1}^{|J|}\left[H^j\overline{V^j}+b^j\overline{\overline{V^j}}\right] \tag{8}$$

where A_i (\$/batch) is the setup cost of production process i, H^j (\$/L/day) is the inventory holding cost of material j, P_k^j (\$/L) is the cost of purchasing material j from vendor k, and $\overline{V^j}$ is the average inventory level of material j. Note that we add two terms $\zeta\sum_{i=1}^{|I|}\sum_{m=1}^{|M|}|D_k^j-\overline{D_k^j}|+\zeta\sum_{i=1}^{|I|}|D_i-\overline{D_i}|$ to Eq. 8 to improve the dynamic properties of the control algorithm; $\overline{D_k^j}$ and $\overline{D_i}$ are the averages of the measured values of

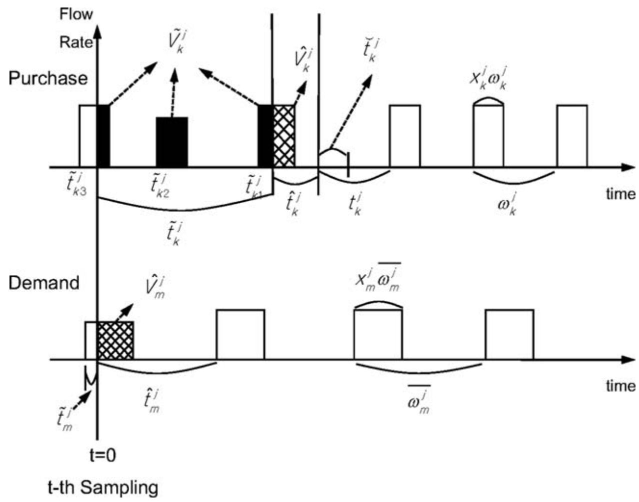

Figure 2. Definition of variables and parameters based on PSW flow.

D^j_k and D_i, respectively, and ζ is a small positive number. This term prevents the arbitrary variation of the control input but does not change the total cost given by Eq. 8. The control problem can be summarized as follows.

Control problem

Adjust purchase/production lot sizes B^j_k/B_i and purchase/production startup times $t^j_k/\grave{t}_i$ to minimize the total cost according to Eq. 8 under the inventory level constraints Eqs. 6 and 7 with respect to the random customer order lot sizes $\mathbf{B}^j_m$ and cycle time $\boldsymbol{\omega}^j_m$.

where $t^j_k/\grave{t}_i$ appear in $\overline{\overline{V^j}}$ and $\underline{\underline{V^j}}$ which will be computed later.

Adaptive computer control concepts are implemented using a nonlinear process model. The controller consists of four functional modules for the calculation of the control input, the prediction of customer demand variables, the prediction of purchase/production variables, and the calculation of ordered quantities during lead time. At a given sampling moment, the controller gathers all current information from the measuring devices or the prediction modules and predicts the future inventory trajectory based on an inventory model developed from the PSW model. The state variables predicted at the previous sampling moment are used at the present time point and the resulting predictions of the state variables are reserved for use in the subsequent sampling moment. Whereas the predictions of the purchase/production variables are expected to be accurate, the prediction of customer demand variables will be affected by random disturbances. Despite numerous process time constraints, the control system is designed to allow sampling intervals of arbitrary length. The sampling moment can occur during processing or storage operations, during which times the control input cannot be processed. In this case, the current control input is processed immediately after the processing and storage operations and before the next sampling moment.

Development of the Control Law

Figure 2 shows the future material flows projected from the current sampling moment. The time index (k) is omitted in the discussion that follows when doing so does not introduce confusion. We assume that no process disturbances will occur in the future. The customer demand flows can be completely characterized using the design values of the system parameters, $\overline{B^j_m}$, $\overline{\omega^j_m}$, x^j_m, and the delay time ($\hat{t}^j_m$). The current operation of the purchase/production flows accounts for the waiting delay times ($\hat{t}^j_k$ / $\grave{\hat{t}}_i$) and the control inputs (t^j_k / $\grave{t}_i$). If the controller desires to increase (decrease) an inventory level, the control inputs for the raw material purchase flows are selected to be less (more) than the advanced delay time ($\breve{t}^j_k$). The opposite is true for production flows, for which the advanced delay time is $\grave{\breve{t}}_i$. Each purchase order has a lead time ($\tilde{t}^j_k$), that is, the time interval between the time of order release and the time at which the ordered material is received. It is thus possible that lots have been ordered but have not yet arrived. The remaining arrival times are denoted $\tilde{t}^j_{kl}$ $(\le \tilde{t}^j_k)$, $l=1,\ 2,\ \ldots,\ |L^j_k|$ where $|L^j_k|$ is a large number. The total quantity of each product ordered during the lead time is indicated by $\tilde{V}^j_k$ which is measurable.

Suppose time = 0 at the current (t-th) sampling moment, the prediction of the future inventory profile can be obtained by integrating the ordinary differential material balance equation for each material[1]

$$
\begin{aligned}
V^j(\text{time}) = {} & V^j(0) + \sum_{k=1}^{|K(j)|}\left[\tilde{V}^j_k(\tilde{t}^j_k) + \hat{V}^j_k(\hat{t}^j_k)\right] \\
& + \sum_{i=1}^{|I|} g^j_i \tilde{V}'_i(t'_i) - \sum_{i=1}^{|I|} f^j_i \grave{\hat{V}}_i(\grave{t}_i) - \sum_{m=1}^{|M|} \hat{V}^j_m(\hat{t}^j_m) \\
& + \sum_{k=1}^{|K(j)|} B^j_k \left[\text{int}\left[\frac{\text{time} - \left(\tilde{t}^j_k + t^j_k\right)}{\omega^j_k}\right] + \min\left\{1, \frac{1}{x^j_k}\text{res}\left[\frac{\text{time} - \left(\tilde{t}^j_k + t^j_k\right)}{\omega^j_k}\right]\right\}\right] \\
& + \sum_{i=1}^{|I|} (g^j_i B_i)\left[\text{int}\left[\frac{\text{time} - t'_i}{\omega_i}\right] + \min\left\{1, \frac{1}{x'_i}\text{res}\left[\frac{t - t'}{\omega_i}\right]\right\}\right] \\
& - \sum_{m=1}^{|M(j)|} B^j_m \left[\text{int}\left[\frac{\text{time} - \hat{t}^j_m}{\omega^j_m}\right] + \min\left\{1, \frac{1}{x^j_m}\text{res}\left[\frac{t - \hat{t}^j_m}{\omega^j_m}\right]\right\}\right] \\
& - \sum_{i=1}^{|I|} (f^j_i B_i)\left[\text{int}\left[\frac{\text{time} - \grave{t}_i}{\omega_i}\right] + \min\left\{1, \frac{1}{\grave{x}_i}\text{res}\left[\frac{t - \grave{t}}{\omega_i}\right]\right\}\right]
\end{aligned}
\tag{9}
$$

where

$$
\hat{V}^j_m(\hat{t}^j_m) = \frac{B^j_m}{x^j_m}\max\left\{0,\ \frac{\hat{t}^j_m}{\omega^j_m} - 1 + x^j_m\right\} \tag{10}
$$

$$
\hat{V}^j_k(\hat{t}^j_k) = \frac{B^j_k}{x^j_k}\max\left\{0,\ \frac{\hat{t}^j_k + \breve{t}^j_k}{\omega^j_k} - 1 + x^j_k\right\} \tag{11}
$$

$$
\grave{\hat{V}}_i(\grave{\hat{t}}_i) = \frac{B_i}{\grave{x}_i}\max\left\{0,\ \frac{\grave{\hat{t}}_i + \grave{\breve{t}}_i}{\omega_i} - 1 + \grave{x}_i\right\} \tag{12}
$$

$$
t'_i = \grave{t}_i + \omega_i(1 - x'_i) \tag{13}
$$

and where $V^j(\text{time})$ is the inventory level of material j, $\tilde{V}^j_k(\tilde{t}^j_k)$ and $\tilde{V}'_i(t'_i)$ are the in-process purchase/production volumes during $\tilde{t}^j_k$ and t'_i, and $\hat{V}^j_m(\hat{t}^j_m)$, $\hat{V}^j_k(\hat{t}^j_k)$, and $\grave{\hat{V}}_i(\grave{\hat{t}}_i)$ are the partial volumes of the current orders/batches, as depicted in Figure 2. t'_i and x'_i are the startup time and SOTF, respectively, for the discharging flow from process i whereas $\grave{t}_i$ and $\grave{x}_i$ are those of the feeding flow to process i. t'_i is removed using Eq. 13.[13] $\hat{t}_m^j$, $\hat{t}_k^j$, and $\backprime\hat{t}_i$ are reserved times due to previous operations, where $\hat{t}_k^j$ is defined graphically in Figure 2. Note that $\hat{t}_m^j \equiv \overline{\omega_m^j} - \tilde{t}_m^j$, and $\backprime t_i \geq \backprime\hat{t}_i$, where $\tilde{t}_m^j$ is a passed processing time for a shipping activity triggered by a previous customer order, defined graphically in Figure 2. The exact minimum inventory of $V^j(\text{time})$ cannot be identified, but the upper and lower bounds can be calculated from the properties of the PSW flow.[1] The upper bound of inventory $\overline{\overline{V^j}}$ is computed by adding the upper bound of the incoming flow to the initial inventory and subtracting the lower bound of the outgoing flows from the initial inventory. The lower bound of inventory $\underline{\underline{V^j}}$ is computed in the opposite manner, by adding the lower bound of the incoming flows to the initial inventory and subtracting the upper bound of the outgoing flows from the initial inventory

$$\begin{aligned}\underline{\underline{V^j}} = & V^j(0) + \sum_{k=1}^{|K(j)|}\left[\tilde{V}_k^j(\tilde{t}_k^j) + \hat{V}_k^j(\hat{t}_k^j)\right] + \sum_{i=1}^{|I|} g_i^j \tilde{V}'_i(t_i') - \sum_{i=1}^{|I|} f_i^j \backprime\hat{V}_i(\backprime\hat{t}_i) \\ & - \sum_{m=1}^{|M|}\hat{V}_m^j(\hat{t}_m^j) - \sum_{i=1}^{|I|} g_i^j D_i t_i' - \sum_{i=1}^{|I|}(1-\backprime x_i) f_i^j D_i \omega_i + \sum_{i=1}^{|I|} f_i^j D_i \backprime t_i \\ & - \sum_{k=1}^{|K(j)|} D_k^j\left[\tilde{t}_k^j + t_k^j\right] - \sum_{m=1}^{|M(j)|}(1-x_m^j)\overline{B_m^j} + \sum_{m=1}^{|M(j)|}\overline{D_m^j}\hat{t}_m^j\end{aligned} \tag{14}$$

$$\begin{aligned}\overline{\overline{V^j}} = & V^j(0) + \sum_{k=1}^{|K(j)|}\left[\tilde{V}_k^j(\tilde{t}_k^j) + \hat{V}_k^j(\hat{t}_k^j)\right] + \sum_{i=1}^{|I|} g_i^j \tilde{V}'_i(t_i') - \sum_{i=1}^{|I|} f_i^j \backprime\hat{V}_i(\backprime\hat{t}_i) \\ & - \sum_{m=1}^{|M|}\hat{V}_m^j(\hat{t}_m^j) + \sum_{i=1}^{|I|}(1-x_i') g_i^j D_i \omega_i - \sum_{i=1}^{|I|} g_i^j D_i t_i' + \sum_{i=1}^{|I|} f_i^j D_i \backprime t_i \\ & + \sum_{k=1}^{|K(j)|}(1-x_k^j) D_k^j \omega_k^j - \sum_{k=1}^{|K(j)|} D_k^j\left[\tilde{t}_k^j + t_k^j\right] + \sum_{m=1}^{|M(j)|}\overline{D_m^j}\hat{t}_m^j\end{aligned} \tag{15}$$

The average inventory level $\overline{V^j}$ is computed by adding and subtracting the linear averages of all flows to and from the initial inventory

$$\begin{aligned}\overline{V^j} = & V^j(0) + \sum_{k=1}^{|K(j)|}\left[\tilde{V}_k^j(\tilde{t}_k^j) + \hat{V}_k^j(\hat{t}_k^j)\right] + \sum_{i=1}^{|I|} g_i^j \tilde{V}'_i(t_i') - \sum_{i=1}^{|I|} f_i^j \backprime V_i(\backprime t_i) - \sum_{m=1}^{|M|}\hat{V}_m^j(\hat{t}_m^j) \\ & + \sum_{k=1}^{|K(j)|}\frac{(1-x_k^j)}{2} D_k^j \omega_k^j - \sum_{k=1}^{|K(j)|} D_k^j\left[\tilde{t}_k^j + t_k^j\right] + \sum_{i=1}^{|I|}\frac{(1-x_i')}{2} g_i^j D_i \omega_i - \sum_{i=1}^{|I|} g_i^j D_i t_i' \\ & - \sum_{i=1}^{|I|}\frac{(1-\backprime x_i)}{2} f_i^j D_i \omega_i + \sum_{i=1}^{|I|} f_i^j D_i \backprime t_i - \sum_{m=1}^{|M(j)|}\frac{(1-x_m^j)}{2}\overline{B_m^j} + \sum_{m=1}^{|M(j)|}\overline{D_m^j}\hat{t}_m^j\end{aligned} \tag{16}$$

where the average flow rate of material j to customer m is $\overline{D_m^j} \equiv \frac{\overline{B_m^j}}{\overline{\omega_m}}$. Note that Eqs. 14–16 were developed by assuming that future flows are completely periodic without disturbances. Therefore, $\overline{\overline{V^j}}$, $\underline{\underline{V^j}}$, and $\overline{V^j}$ are predicted values, and the true values will be different in the presence of uncertainty. The following overall material balance should be satisfied

$$\sum_{i=1}^{|I|} g_i^j D_i + \sum_{k=1}^{|K(j)|} D_k^j = \sum_{i=1}^{|I|} f_i^j D_i + \sum_{m=1}^{|M(j)|}\overline{D_m^j} \tag{17}$$

The control input is calculated so that the objective function in Eq. 8 reaches a minimum under the constraints established by Eqs. 6, 7, and 17. Suppose D_k^j and D_i are initially specified constants, the Kuhn–Tucker conditions are then solved with respect to ω_k^j, ω_i, t_k^j, and $\backprime t_i$. The resulting objective function is minimized with respect to D_k^j and D_i. This two-level approach is equivalent to the original formulation for this specially structured problem.[1]

Problem 1. *Minimizing the objective function in Eq. 8, under the constraints Eqs. 6 and 7, and the equations given in Eqs. 14–16, with respect to the design variables* ω_k^j, ω_i, t_k^j, and $\backprime t_i$.

Note that the Kuhn–Tucker conditions of Problem 1 has a global optimum because the objective function is convex (the sum of reciprocal and linear terms) and the constraints are linear. The Kuhn–Tucker conditions for optimization Problem 1 can be obtained in analytical form by means of the algebraic manipulations summarized in Appendix A. The optimal cycle times for raw material purchases are given by

$$\omega_k^j = \sqrt{\frac{A_k^j}{D_k^j(1-x_k^j)\overline{\overline{\lambda^w}} + D_k^j\Psi_k^j}} \tag{18}$$

where $\Psi_k^j = \left(\frac{H^j}{2} + b^j\right)(1-x_k^j) + a_k^j$. The optimal cycle times for production processes are given by

$$\omega_i = \sqrt{\frac{A_i}{\left[\sum_{j=1}^{|J|}\left[(1-\backprime x_i)f_i^j + \left(1-x_i'\right)g_i^j\right]\overline{\overline{\lambda^w}} + \Psi_i\right]D_i}} \tag{19}$$

where $\Psi_i \equiv a_i\sum_{j=1}^{|J|} f_i^j + (1-\backprime x_i)\sum_{j=1}^{|J|}\left(\frac{H^j}{2} + b^j\right)f_i^j + \left(1-x_i'\right)\sum_{j=1}^{|J|}\left(\frac{H^j}{2} + b^j\right)g_i^j$. Under the optimal conditions, the lower bound on the inventory is the safety level, that is

$$\underline{\underline{S^j}} = \underline{\underline{V^j}} \tag{20}$$

The optimal objective value is

$$\begin{aligned}{}^{*}\mathrm{TC}(D_k^j,\ D_\mathrm{i}) = & 2\sum_{j=1}^{|J|}\sum_{k=1}^{|K(j)|}\sqrt{A_k^j\left((1-x_k^j)\overline{\overline{{}^{*}\lambda^w}} + \Psi_k^j\right)D_k^j} \\ & + 2\sum_{i=1}^{|I|}\sqrt{A_i\left(\sum_{j=1}^{|J|}\left[(1-\backprime x_i)f_i^j + \left(1-x_i'\right)g_i^j\right]\overline{\overline{{}^{*}\lambda^w}} + \Psi_i\right)D_i} \\ & + \sum_{j=1}^{|J|}\left(\frac{H^j}{2} + b^j\right)\sum_{m=1}^{|M(j)|}\overline{B_m^j}(1-x_m^j) - \sum_{w=1}^{|W|}\overline{\overline{{}^{*}\lambda^w}}\,\overline{\overline{S^w}}\end{aligned} \tag{21}$$

The Lagrangian multipliers $\overline{\overline{\lambda^w}} \geq 0$ are computed using the following equation

$$\begin{aligned}\overline{\overline{\lambda^w}}\Bigg[\sum_{j\in J(w)}\Bigg\{ & \sum_{i=1}^{|I|}\left[(1-\backprime x_i)f_i^j + (1-x_i')g_i^j\right]D_i\omega_i \\ & + \sum_{k=1}^{|K(j)|}(1-x_k^j)D_k^j\omega_k^j + \sum_{m=1}^{|M(j)|}(1-x_m^j)\overline{B_m^j}\Bigg\} - \overline{\overline{S^w}}\Bigg] = 0\end{aligned} \tag{22}$$

Note that the above optimal solution set holds for $|J| \leq |I|$ and that, for simplicity, the case involving the other solution set is not considered. The above solution set does not provide the optimal average flow rates through the network. The optimal average flow rates can be obtained by solving another optimization problem, summarized as Problem 2:

PROBLEM 2. *Minimize the objective function Eq. 21, under the constraints Eqs. 17 and 22, with respect to* D_k^j, D_i, *and* $\overline{\overline{\lambda^w}}$.

Additional constraints, such as $\overline{\overline{D_i}} \geq \sum_{j=1}^{|J|} D_i \geq \underline{\underline{D_i}}$ and $\overline{\overline{B_i}} \geq B_i \geq \underline{\underline{B_i}}$, can be introduced into Problem 2; the upper double bar denotes the upper bound and the lower double bar denotes the lower bound. After obtaining the optimal average flow rates and the Lagrangian multipliers ${}^*D_k^j$, ${}^*D_i$, and $\overline{\overline{\lambda^w}}$ by solving Problem 2, we compute the optimal cycle times ${}^*\omega_k^j$ and ${}^*\omega_i$ and startup times ${}^*t_k^j$ and ${}^*\grave{t}_i$ by solving Eqs. 18–20, respectively. Both Problem 1 and Problem 2 are nonconvex NLPs that are difficult to apply in a real-time optimization procedure; however, Problem 2 has a special structure that can be transformed into a more easily applied form. Consider optimization Problem 3, which is obtained by specifying $\overline{\overline{\lambda^w}}=0$ in Problem 2. Equation 22 then becomes

$$\sum_{j \in J(w)} \left\{ \sum_{i=1}^{|I|} \left[(1-\grave{x}_i) f_i^j + (1-x_i') g_i^j \right] D_i \omega_i + \sum_{k=1}^{|K(j)|} (1-x_k^j) D_k^j \omega_k^j + \sum_{m=1}^{|M(j)|} (1-x_m^j) \overline{B_m^j} \right\} < \overline{\overline{S^w}} \tag{23}$$

PROBLEM 3. *Minimize the objective function Eq. 21, under the constraints Eqs. 17 and 23, with respect to* D_k^j *and* D_i.

Note that the solution of Problem 3 provides a good feasible solution of Problem 2 as $\overline{\overline{\lambda^w}}=0$ minimizes each term of Eq. 21 independently. Problem 3 is actually a planning model and it can be replaced with other planning models without destroying subsequent optimality. Its objective function is the sum of concave univariable functions and it has linear constraints. Thus, it can be solved without difficulty using separable programming techniques.[1] After computing the average flow rates, we can use $\omega_k^j = \sqrt{\frac{A_k^j}{D_k^j \Psi_k^j}}$, $\omega_i = \sqrt{\frac{A_i}{\Psi_i D_i}}$, $B_k^j = D_k^j \omega_k^j$, and $B_i = D_i \omega_i$.

MPC Algorithm

Equation 20 can be rewritten as

$$\begin{aligned} &\sum_{i=1}^{|I|} \left(g_i^j - f_i^j \right) D_i \grave{t}_i + \sum_{k=1}^{|K(j)|} D_k^j \left[t_k^j + \tilde{t}_k^j \right] - \sum_{m=1}^{|M(j)|} \overline{D_m^j}(n) t_m^j = V^j(0) - \underline{\underline{S^j}} \\ &+ \sum_{k=1}^{|K(j)|} \left[\tilde{V}_k^j(\tilde{t}_k^j) + \hat{V}_k^j(\hat{t}_k^j) \right] + \sum_{i=1}^{|I|} g_i^j \tilde{V}'_i(t_i') - \sum_{i=1}^{|I|} f_i^j \grave{V}_i(\grave{\hat{t}}_i) - \sum_{m=1}^{|M|} \hat{V}_m^j(\hat{t}_m^j) \\ &- \sum_{i=1}^{|I|} \left[(1-\grave{x}_i) f_i^j + g_i^j (1-x_i') \right] {}^*B_i - \sum_{m=1}^{|M(j)|} (1-x_m^j) \overline{B_m^j}(n) \end{aligned} \tag{24}$$

where the index $n(t) = 1, 2, \ldots$ is the number of batch sequences from the start of operation at a current sampling time t. Note that the decision variables t_k^j and $\grave{t}_i$ have lower bounds of $\hat{t}_k^j$ and $\grave{\hat{t}}_i$ and that Eq. 24 is no longer feasible when the current inventory $V^j(0)$ is significantly lower than the safety level $\underline{\underline{S^j}}$ due to unexpectedly large disturbances. If Eq. 24 is not feasible, optimality is destroyed. In order for Eq. 24 to be always feasible, a new variable $t_m^j \geq \hat{t}_m^j$, which is the delay time of customer demand, is added. A value of $t_m^j > \hat{t}_m^j$ results in backlogging or lost sales, so t_m^j should be minimized. Note that ${}^*B_i = \sqrt{\frac{A_i {}^* D_i}{\Psi_i}}$ is already determined. Solving Eq. 24 requires linear programming with respect to t_m^j, t_k^j , and $\grave{t}_i$. t_k^j and $\grave{t}_i$ should be close to their advanced values $\breve{t}_k^j$ and $\grave{\breve{t}}_i$, respectively. $t_k^j > \breve{t}_k^j$ and $\grave{\hat{t}}_i \leq \grave{t}_i < \grave{\breve{t}}_i$ decreases the inventory of material j which feeds into the process i. The opposite is true for $\hat{t}_k^j \leq t_k^j < \breve{t}_k^j$ and $\grave{t}_i > \grave{\breve{t}}_i$.

PROBLEM 4. *Minimize the objective function*

$$\rho_k^j \left[\left(t_k^j \right)^+ + \left(t_k^j \right)^- \right] + \rho_i \left[(\grave{t}_i)^+ + (\grave{t}_i)^- \right] + \rho_m^j t_m^j \tag{25}$$

under the constraints Eq. 24 and

$$t_k^j - \left(\breve{t}_k^j + \hat{t}_k^j \right) = \left(t_k^j \right)^+ - \left(t_k^j \right)^- \qquad \left(t_k^j \right)^+ \geq 0, \quad \breve{t}_k^j \geq \left(t_k^j \right)^- \geq 0$$

$$\grave{t}_i - \left(\grave{\breve{y}}_i + \grave{\hat{y}}_i \right) = (\grave{t}_i)^+ - (\grave{t}_i)^- \qquad (\grave{t}_i)^+ \geq 0, \quad \grave{\breve{t}}_i \geq (\grave{t}_i)^- \geq 0$$

with respect to $t_m^j \geq \hat{t}_m^j$, $t_k^j \geq \hat{t}_k^j$ and $\grave{t}_i \geq \grave{\hat{t}}_i$.

where rho's are weighting factors for each term of objective function. Note that Problem 4 is for practical purpose a scheduling model. $V^j(0)$, $\tilde{V}_k^j(\tilde{t}_k^j)$, $\tilde{V}'_i(t_i')$, and $\tilde{t}_m^j$ are the measured variables. The parameters $\hat{t}_m^j$, $\hat{t}_k^j$, $\grave{\hat{t}}_i$, $\hat{V}_k^j(\hat{t}_k^j)$, $\grave{\hat{V}}_i(\grave{\hat{t}}_i)$, $\hat{V}_m^j(\hat{t}_m^j)$, $\overline{B_m^j}(n)$, and $\overline{\omega_m^j}(n)$ can be estimated at each sampling time from the previously measured data $\mathbf{B}_m^j(n')$ and $\boldsymbol{\omega}_m^j(n')$, where $n' = 1, 2, \ldots, n$. $\overline{B_m^j}(n)$ and $\overline{\omega_m^j}(n)$ are estimated by calculating simple arithmetic means

$$\bar{B}_m^j(n) = \frac{\sum_{n'=1}^{n} \phi^{n-n'} \mathbf{B}_m^j(n')}{\sum_{n'=1}^{n} \phi^{n-n'}} \text{ and } \bar{\omega}_m^j(n) = \frac{\sum_{n'=1}^{n} \phi^{n-n'} \boldsymbol{\omega}_m^j(n')}{\sum_{n'=1}^{n} \phi^{n-n'}} \tag{26}$$

where $0 < \phi \leq 1$ is a forgetting factor. Equation 26 also has a recursive form.[17] This real-time parameter estimation scheme makes the algorithm adaptive.

In real applications, Eq. 24 might not prevent the depletion of inventories, as described in Eq. 7, because the disturbance can be larger than the expected value; thus, we introduce a tuning parameter $\sigma^j \geq 0$ such that $\underline{\underline{S^j}}$ is replaced with $\underline{\underline{S^j}} + \sigma^j$.

The predictions of $\hat{t}_k^j(t+1)$, $\grave{\hat{t}}_i(t+1)$, $\breve{t}_k^j(t+1)$, and $\grave{\breve{t}}_i(t+1)$ from $\hat{t}_k^j(t)$, $\grave{t}_i(t)$, $\breve{t}_k^j(t)$, and $\grave{\breve{t}}_i(t)$ are summarized in Table 1.[16] Now, we are in a position to propose a control algorithm.

MPC Algorithm:

<1> At sampling time, t, measure $\mathbf{B}^j_m(n)$, $\boldsymbol{\omega}^j_m(n)$, $V^j(0)$, $\tilde{V}^j_k(\tilde{t}^j_k)$, $\tilde{V}'_i(t'_i)$, and $\tilde{t}^j_m$.

<2> Estimate , ${}^{\backprime}V_i({}^{\backprime}t_i)$, $\hat{V}^j_m(\hat{t}^j_m)$, $\overline{B^j_m}(n)$ and $\overline{\boldsymbol{\omega}_m}(n)$ by using Eqs. 10, 11, 12, and 26.

<3> Compute the control inputs ${}^{*}B^j_k(t)$, ${}^{*}B_i(t)$, ${}^{*}\boldsymbol{\omega}^j_k(t)$, ${}^{*}\boldsymbol{\omega}_i(t)$, ${}^{*}t^j_k$, and ${}^{*}t_i(t)$ according to Problem 3, and Problem 4 in sequence. Note that the computed t^j_k and ${}^{\backprime}t_i$ are activated only when $t^j_k < \hat{t}^j_k$ and ${}^{\backprime}t_i < {}^{\backprime}\hat{t}_i$ after the rolling horizon.

<4> Predict $\hat{t}^j_k(t+1)$, ${}^{\backprime}\hat{t}_i(t+1)$, $\breve{t}^j_k(t+1)$, and ${}^{\backprime}\breve{t}_i(t+1)$ according to Table 1.

<5> Update $t+1$ to t and go to step **<1>**.

We can prove the following stability theorem for the MPC Algorithm.

Theorem 2. *The Boundedness of the Control Output of the MPC Algorithm*

It is assumed that $\mathbf{B}^j_m(n)$ *has a symmetrical distribution function such that* $\underline{\underline{B^j_m}} \le \mathbf{B}^j_m(n) \le \overline{\overline{B^j_m}}$ *and that* $\boldsymbol{\omega}^j_m(n)$ *has a nonsymmetrical distribution function such that* $\underline{\underline{\omega^j_m}} \le \boldsymbol{\omega}^j_m(n)$. *Suppose also that* $\mathbf{B}^j_m(n)$ *and* $\boldsymbol{\omega}^j_m(n)$ *have identical independent distribution functions with respect to (n). We will consider only the case* $\phi=1$. *For the convergence limits* $0<\varepsilon_1,\varepsilon_2 \ll 1$ *and confidence levels* $0<\delta_1,\delta_2 \ll 1$,

1. there exists an integer N such that

$$P\left\{\left|\frac{1}{N}\sum_{n=1}^{N}\mathbf{B}^j_m(n)-\overline{B^j_m}\right| < \varepsilon_1\right\} \ge 1-\delta_1$$
$$\text{and } P\left\{\left|\frac{1}{N}\sum_{l=1}^{N}\mathbf{B}^j_m(n)-\overline{\omega^j_m}\right| < \varepsilon_2\right\} \ge 1-\delta_2 \tag{27}$$

From Chebycheff's inequality, $N \ge \frac{\mathrm{Var}\left(\mathbf{B}^j_m(n)\right)}{\delta_1 {\varepsilon_1}^2}$ and $N \ge \frac{\mathrm{Var}\left(\boldsymbol{\omega}^j_m(n)\right)}{\delta_2 {\varepsilon_2}^2}$.

2.

$$|\underline{\underline{S^j}} - \underline{\underline{{}^{*}V^j}}| \le \sum_{m=1}^{|M|}\varepsilon_1(x^j_m)+\sum_{m=1}^{|M|}\tilde{t}_m\left[\frac{\overline{\omega^j_m}\varepsilon_1+\overline{B^j_m}\varepsilon_2}{\underline{\underline{\omega^j_m}}\,\overline{\omega_m}}\right]$$
$$+\left|\sum_{m=1}^{|M|}\overline{B^j_m}\left[\alpha^j_m\theta^j_m-(1-x^j_m)(1-\alpha^j_m)\right]\right|$$
$$+\left[\max{}_m\left\{\frac{\overline{\omega^j_m}\varepsilon_1+\overline{B^j_m}\varepsilon_2}{\overline{B^j_m}\,\underline{\underline{\omega^j_m}}}\right\}+1\right]\sum_{m=1}^{|M|}\overline{D^j_m}\left(t^j_m-\hat{t}^j_m(n)\right)$$

and

$$\sum_{m=1}^{|M|}\left[\left(\frac{\overline{\overline{B^j_m}}}{\underline{\underline{\omega^j_m}}}-\overline{D^j_m}\right)\frac{N\underline{\underline{\omega^j_m}}}{2}+\max\left\{0,\ \left(\frac{\underline{\underline{B^j_m}}}{\underline{\underline{\omega^j_m}}}-\overline{D^j_m}\right)\right\}\frac{N\underline{\underline{\omega^j_m}}}{2}+2\overline{\overline{B^j_m}}\right]$$
$$>\sum_{m=1}^{|M|}\overline{D^j_m}\left(t^j_m-\hat{t}^j_m(n)\right)$$

for $n \ge N$

where $\theta^j_m=\left[\left(\frac{1}{\alpha^j_m}-1\right)+\frac{\left(\alpha^j_m-\beta^j_m\right)^{+}}{2\alpha^j_m}\right]N$, $\alpha^j_m \equiv \frac{\underline{\underline{\omega^j_m}}}{\omega^j_m}$, $\beta^j_m \equiv \frac{\underline{\underline{B^j_m}}}{B^j_m}$, $\overline{B^j_m} \equiv \frac{\overline{\overline{B^j_m}}+\underline{\underline{B^j_m}}}{2}$.

$N=\max\left\{\mathrm{int}\left[\frac{\mathrm{Var}\left(\mathbf{B}^j_m(n)\right)}{\delta_1{\varepsilon_1}^2}\right],\ \mathrm{int}\left[\frac{\mathrm{Var}\left(\boldsymbol{\omega}^j_m(n)\right)}{\delta_2{\varepsilon_2}^2}\right]\right\}+1$ if the least integer is chosen.

Here, Var(.) is the variance operator and ${}^{*}V^j$ is the true lower bound of V^j(time) under conditions of $\overline{\overline{\text{unce}}}$rtainty.

Proof. 1. This is the weak law of large numbers.[16]

2. See Appendix B.

Note that for simplicity Theorem 2 does not consider the impact of the tuning parameter o^j. Note also that this Theorem guarantees only the boundedness of the control output. A convergent algorithm is expected in the future development. The MPC Algorithm is more compact and effective than Algorithm 4 developed in previous research.[16] Algorithm 4[16] combined the MPC design method with a traditional control scheme to deal with the problem that Eq. 24 could not be met due to the limit of variables $t^j_k \ge \hat{t}^j_k$ and ${}^{\backprime}t_i \ge {}^{\backprime}\hat{t}_i$. The MPC Algorithm overcomes this limitation by relaxing the fixed parameter t^j_m as hypothetical variables and minimizing them. Thus, Eq. 24 is always satisfied and boundedness is ensured in Theorem 2. ■

Heuristic Algorithms

To the best of our knowledge, there is no established production and inventory control algorithm for BSN whose processes have multiple feedstocks and products. We suggest the following bang–bang style control policy similar to that of manual operations.

Heuristic Algorithm

<1> Before initiating the algorithm, the optimal order/batch quantities and cycle times at $t=0$ are identified by using the planning model. The order/batch quantities are fixed.

<2> The upper limit of each inventory level $\overline{\overline{S^j}}$ is defined such that $\sum_{j\in J(w)}\overline{\overline{S^j}}=\overline{\overline{S^w}}$ and $\overline{\overline{S^j}}-\underline{\underline{S^j}}=\left(\overline{\overline{S^w}}-\sum_{j\in J(w)}\underline{\underline{S^j}}\right)\frac{BB^j}{\sum_{j\in J(w)}\left[BB^j\right]}$.

Table 1. Prediction of Next Step Parameters

	${}^{\backprime}\hat{t}_i(t+1)$	${}^{\backprime}t_i(t)$	${}^{\backprime}\breve{t}_i(t+1)$
${}^{\backprime}\hat{t}_i \le \tau < \tau_4$	0.0	${}^{\backprime}t_i(t)$	$\tau_3-\tau$
$\tau_1 \le \tau < \tau_2$	$\tau_2-\tau$	${}^{\backprime}t_i(t)$	$\omega_i-\underline{\underline{\omega_i}}$
$\tau_4 \le \tau < \tau_1$	0.0	q	0.0
${}^{\backprime}\hat{t}_i \le \tau < \tau_4$	0.0	q	$\tau_0-\tau$
$0 \le \tau < {}^{\backprime}\hat{t}_i$	$-q$	0.0	${}^{\backprime}\breve{t}_i(k)$

$\tau_0={}^{\backprime}\hat{t}_i(t)+{}^{\backprime}\breve{t}_i(t)$

$\tau_1={}^{\backprime}\hat{t}_i(t)+\overline{\overline{t^j}}(t)$

$\tau_4=\min\{\tau_0,\ \tau_1\}$

$T=\max\{0,(\tau-\tau_1)/\omega_i\}$

$\tau_2=\tau_1+\mathrm{int}[T]\omega_i+\underline{\underline{\omega_i}}$

$\tau_3=\tau_1+\mathrm{int}[T]\omega_i+\omega_i$

$q=\tau-{}^{\backprime}\hat{t}_i(t)$

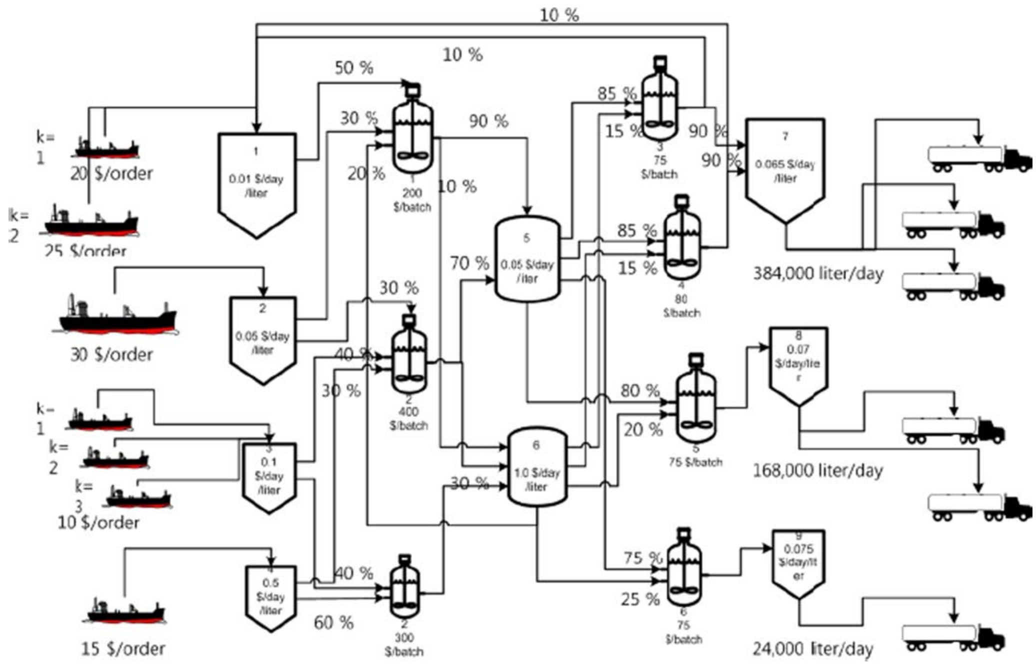

Figure 3. Example multiproduct batch plant.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

where

$$BB^j \equiv \sum_{k=1}^{|K|}(1-x_k^j)B_k^j + \sum_{m=1}^{|M|}(1-x_m^j)\overline{B_m^j} + \sum_{i=1}^{|I|}\left[(1-{}^{`}x_i)f_i^j + g_i^j(1-x_i')\right]B_i.$$

<3> At each sampling time, t, if current inventory levels are less than the safety level, set the next startup times of the connected incoming flows to the minimum values and set those of the connected outgoing flows to the maximum values. If current inventory levels are greater than the upper inventory limits computed at **<2>**, set the next startup times of the connected incoming flows to the maximum values and those of the connected outgoing flows to the minimum values. If a current inventory level is within the feasible zone, set the next startup time of the connected incoming and outgoing flows to the optimal cycle times computed at **<1>**.

<4> Apply <3> to all inventories from raw materials to finished products; in each case, the last decision about the startup time will override the previous one. Occasionally, update <1> and <2> with new data. The update frequency strongly influences the performance.

Note that the calculated startup time is accepted if it is less than the lead time. If this is not the case, the order/batch

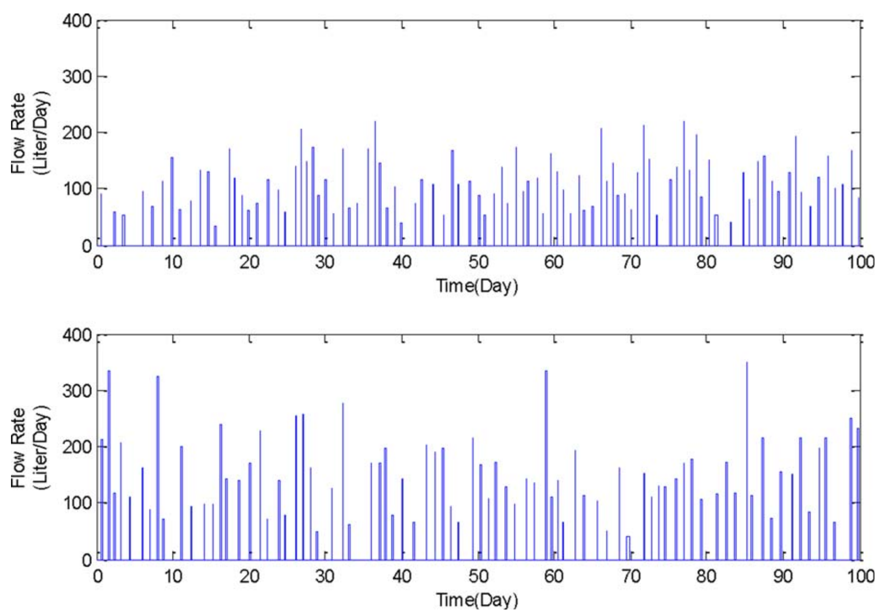

Figure 4. Random two customer demand flows of product 9 (average flow rate = 100, 140 and average cycle time = 1.1, 1.2).

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

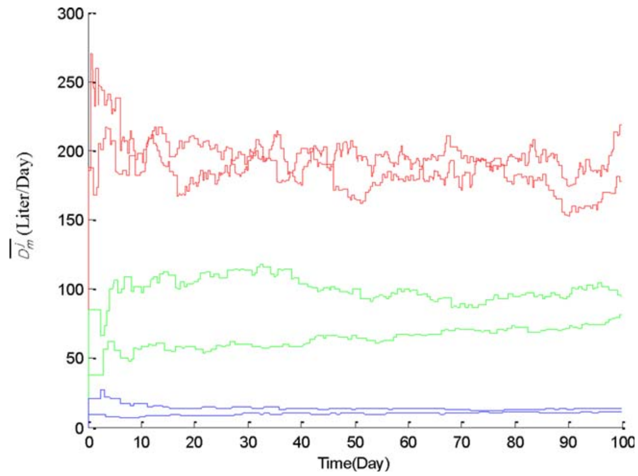

Figure 5. Estimated average flow rates of customer demands.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

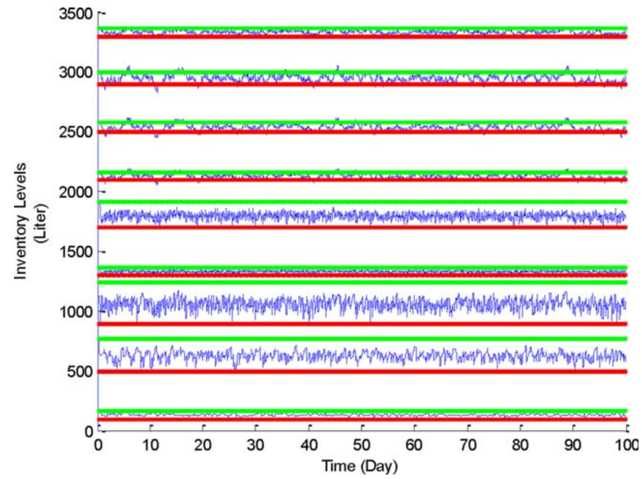

Figure 7. Inventory levels controlled by MPC algorithm.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

is discarded because the order/batch can be submitted in the subsequent sampling moment. The Heuristic Algorithm does not use an iterative routine except for occasional updates and thus, the computational time is very short. Our simulations showed that the Heuristic Algorithm is very stable when the sampling interval is sufficiently short in spite of its simplicity although it has many critical design parameters that must be adjusted carefully.

Discussion and Simulation Examples

Figure 3 shows an example of a plant composed of highly interlinked batch and storage units as well as recycling streams. Cost factors were included in the simulations of the two algorithms compared here: inventory holding cost = 1 $/L/day, backlogging cost = 10,000 $/L/day, excess inventory cost = 10,000 $/L/day, purchase setup costs 10–30 $/order, and production setup costs 75–400 $/batch. Here, backlogging arises when the inventory level has dropped below the safety level, and excess inventory arises when the inventory level exceeds the warehouse capacity. The total cost is the sum of the inventory holding cost, the backlogging cost, the excess inventory cost, and the purchasing/production setup cost over a 100 day simulation. The capital costs are ignored without loss of generality. The sampling interval τ can be any real number and was set at 0.01 day, which is a suitable interval for the Heuristic Algorithm. Simulations were carried out for a batch plant comprising 9 materials, 7 processes, and 1 warehouse. The minimum/maximum lot sizes of each order/batch were specified as 0.5/1.5 times of the design value. The sizes of the customer orders were subject to random disturbances comprising variations of up to 50% distributed uniformly over the size variation interval. Customer order cycle times were subject to second-order gamma-distributed random disturbances and computed using the following MATLAB command

$$\mathbf{B_m^j} = \overline{B_m^j} * (1 + (\text{rand}\,(.) - 0.5)/0.5 * 0.5)$$

$$\boldsymbol{\omega}_m^j = 0.5 * \overline{\omega_m^j} * (1 + 0.5 * \text{gam rnd}\,(2, 1, \ \ldots))$$

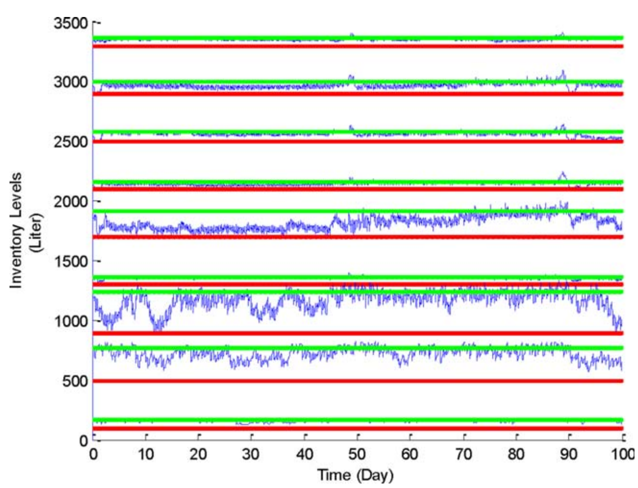

Figure 6. Inventory levels controlled by Heuristic algorithm.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

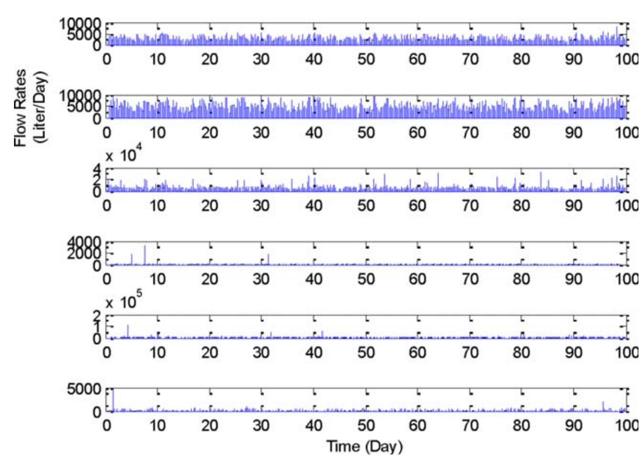

Figure 8. Batch processing computed by MPC algorithm.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 2. Summary of Simulation Results

	Heuristic Algorithm	MPC Algorithm
Volume of overstock (L)	7.3275	1.6010
Volume of backlog (L)	0.3732	2.8332
Volume of inventory (L)	933.4266	581.7437
Number of setup	6908	6940
Setup cost (\$/day)	7.2493e+003	7.2794e+003
Total cost (\$/day)	2.0765e+004	1.5623e+004
Computation time (s)	38.519368	2853.081967

where $\overline{B_m^j} = 11$–92 L and $\overline{\omega_m^j} = 0.3$–$1.2$ days. Customer orders did not have lead times. The purchase orders from vendors had lead times of $\tilde{t}_k^j = 1$ day. The algorithm performances were found to depend strongly on the lead times. Other relevant parameters were as follows: SOTFs = 0.1, minimum cycle times = 0.5*design values, the initial delay times of customer demand were in the range 0.03–0.72, and the initial startup times = 0. The first values of the lot sizes and the cycle times were computed by solving Problem 3. $\overline{\overline{S^j}} - \underline{\underline{S^j}}$ where $\overline{\overline{S^j}}$ satisfies $\overline{\overline{S^w}} = \sum_{j \in J(w)} \overline{\overline{S^j}}$ were set at values twice those of design values, which were computed by BB^j[1] and the values $\underline{\underline{S^j}}$ were selected to facilitate visualization of the simulation results. The MPC tuning parameters were set to $o^j = 0.2 * \left(\overline{\overline{S^j}} - \underline{\underline{S^j}}\right)$ and $\zeta = 0.001$. The Heuristic Algorithm was implemented with $o^j = 0$ and the average flow rates were updated twice during the whole horizon. Any strategy to adjust o^j for the Heuristic Algorithm did not work. The weighting coefficients of Problem 4, ρ_k^j, ρ_i, and ρ_m^j, were found to have significant influences on control performance. $\rho_k^j = D_k^j$, $\rho_i = D_i$, and $\rho_m^j = 100 * D_m^j$ were selected by performing simulation studies that emphasized large values of ρ_m^j so as to reduce backlogging.

Figure 4 shows the customer order variations for product 9 which has two customers. Figure 5 shows the estimated values of $D_m^j(t)$ for each product j (different colors) and customer m (two customers per each product) using Eq. 26. Figures 4 and 5 show that the variations of randomness are very intensive. Figures 6 and 7 show the inventory profiles of each material j for the Heuristic and MPC Algorithms, respectively. The red lines represent safety level $\underline{\underline{S^j}}$, the green lines represent $\overline{\overline{S^j}}$, and the blue lines represent inventory level. Note that the initial inventory levels were intentionally distributed to visualize all inventory profiles. Figures 6 and 7 show that the frequency of violating inventory targets for Heuristic Algorithm is more than that for MPC Algorithm. Figure 8 shows the batch flows that are the control inputs for the MPC Algorithm. The resulting simulation data are summarized in Table 2. The total cost implementing the MPC Algorithm was 75% of that of the Heuristic Algorithm, even though the computational time increased by a factor of 73. When the values of $\overline{\overline{S^j}} - \underline{\underline{S^j}}$ were reduced to 1.5 times the design values (which is called a narrow set band), the total cost of implementing the MPC Algorithm was 77% of that of the Heuristic Algorithm, as are shown at Figures 9 and 10. We can observe the oscillation of inventory profiles for the Heuristic Algorithm. The performances of the algorithms were compared by changing the lead time from 0.1 to 1 day and the results are summarized in Table 3. The total cost of implementing the MPC Algorithm was found to be 64% of that of the Heuristic Algorithm on average before failure occurs.

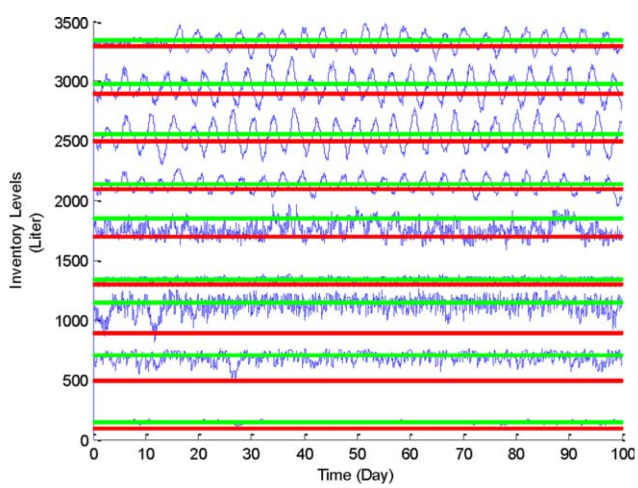

Figure 9. Inventory levels controlled by Heuristic algorithm for narrow set band.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

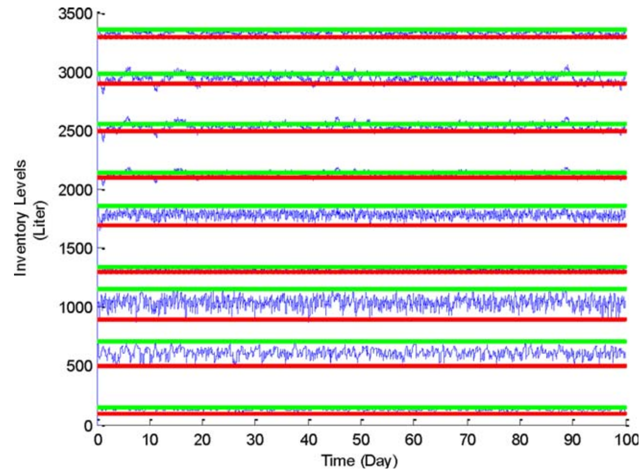

Figure 10. Inventory levels controlled by MPC algorithm for narrow set band.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Solving Problem 3 is the most time-consuming step in the MPC Algorithm. Problem 3 is a nonlinear programming system composed of a separable concave objective function comprising a sum of square roots and linear constraints to be minimized. Ordinary nonlinear programming methods exhibit poor performance in the minimization of concave cost functions. The objective function can be linearized using separable programming techniques,[1] and the nonlinear

Table 3. Total Costs with Respect to Lead Times

Lead Time (day)	Total Cost of Heuristic (\$/day)	Total Cost of MPC (\$/day)
0	1.1059e+004	8.2868e+003
0.2	1.7640e+004	8.2490e+003
0.4	1.9894e+004	1.0013e+004
0.6	2.2887e+004	1.4888e+004
0.8	1.1411e+005	2.0751e+004
1	2.8082e+005	2.7605e+004
1.2	Failed	3.5390e+004
1.4	Failed	4.4159e+004

programming system was converted into a MILP system using specially ordered sets. Many commercial optimization software packages, such as GAMS/CPLEX, GAMS/XPRESS, and GAMS/GUROBI include a function that defines specially ordered sets.[18] The square root terms in the objective function can be approximated with acceptable accuracy using at least five linear functions.[1]

Conclusions

An MPC scheme was developed for the control of a multi-product batch plants in which the material flows between a processes and storage units can be described with a PSW model. The startup times and purchase/production quantities were controlled to maintain inventory levels within warehouse capacity while meeting randomly distributed customer demands. The solution to the Kuhn–Tucker conditions for the optimization problem produces an analytic equation describing the cycle times and a concave cost minimization network flow problem with respect to average flow rates. In our approach, this problem can be solved using commercial optimization software that facilitates separable programming techniques. The average cycle time and order quantity under random customer demand conditions were computed recursively from measured customer orders and were updated into the PSW model to create an adaptive control scheme. The tuning parameters were the control target inventory levels. The boundedness of the control output was mathematically proven under the assumption that the disturbances are bounded.

Simulations were performed that demonstrated the effectiveness of this automatic production and inventory optimization scheme. The performance of the proposed MPC scheme was compared with a well-tuned heuristic control scheme with an approach that is similar to that of manual operations. The costs associated with constraint violations, such as overstocking/backlogging, were assigned a cost that is a factor of 10,000 greater than the inventory holding cost associated with constraint conformity. The MPC scheme was found to reduce the total costs by approximately 23%, although the computation time is increased by a factor of more than 73. MPC approaches are notoriously difficult to implement because of their long computational times; however, we successfully implemented our MPC scheme using a high-speed CPU.

Notation

Sets

i = batch process index
j = material index
k = vendor index
l = purchase order sequence during lead time
m = customer index
$n(t)$ = number of batch sequences from the start of the operation at a current sampling time t
w = warehouse index
t = sampling moment

Operators

Var(.) = variance operator
int[.] = truncation function used to produce an integer
res[x] = residual function computed by x-int[x]
$|X|$ = number of elements in set X
$\underline{\overline{X}}$ = average of X
$\overline{\overline{X}}$ = maximum, upper bound or limit of X
$\underline{X}$ = minimum, lower bound or limit of X
$^{*}\overline{X}$ = optimum of X

Normal letters

a_i = annualized capital cost of process i, \$/L/day
b^j = annualized capital cost of storage facility, \$/L/day
A_i = setup cost of batch process i, \$/batch
A_k^j = setup cost of purchasing material j from vendor k, \$/order
B_k^j = lot size of raw material purchase for material j from vendor k, control input, L
B_i = lot size of batch process, control input, L
$\mathbf{B}_m^j$ = random order size of product j from customer m, L
$\overline{B_m^j}$ = average of $\mathbf{B}_m^j$, L
BB^j = $BB^j \equiv \sum_{k=1}^{|K|}(1-x_k^j)B_k^j+\sum_{m=1}^{|M|}(1-x_m^j)\overline{B_m^j} + \sum_{i=1}^{|I|}\left[(1-{}^{`}x_i)f_i^j+g_i^j(1-x_i')\right]B_i$, L
D_k^j = average flow rate of raw material purchase for material j from vendor i, L/day
D_i = average flow rate of process i, L/day
D_m^j = average flow rate of product j to customer m, L/day
$\overline{D_m^j}$ = average of D_m^j under uncertainty, L/day
$\mathbf{e}_{\mathbf{B}_m^j}$ = error of random variable $\mathbf{B}_m^j$, L
$\mathbf{e}_{\boldsymbol{\omega}_m^j}$ = error of random variable $\boldsymbol{\omega}_m^j$, day
$\mathbf{e}_{\mathbf{x}_m^j}$ = error of random variable $\mathbf{x}_m^j$
f_i^j = feedstock composition of material j for process i
g_i^j = product yield of material j for process i
H^j = inventory holding cost of material j, \$/L/day
I = set of batch processes
J = set of materials
K = set of vendors
M = set of customers
N = defined by $N=\max\left\{\text{int}\left[\frac{\text{Var}(\mathbf{B}_m^j(n))}{\delta_1\varepsilon_1^2}\right],\ \text{int}\left[\frac{\text{Var}(\boldsymbol{\omega}_m^j(n))}{\delta_2\varepsilon_2^2}\right]\right\}+1$
P_k^j = purchasing cost of raw material j from vendor k, \$/L
$\overline{\overline{S^w}}$ = total capacity of the warehouse w, L
S^j = safety stock level for material j, L
${}^{`}t_i$ = startup time of feeding flow of batch process i, control input, day
t_i' = startup time of discharging flow of batch process i, day
t_k^j = startup time of purchase order of material j from vendor k, control input, day
t_m^j = startup time of customer demand of material j from customer m, day
$\tilde{t}_m^j$ = past processing time for a shipping activity triggered by customer demand, day
$\tilde{t}_k^j$ = purchasing order lead time for material j from vendor k, day
$\tilde{t}_{kl}^j$ = remaining arrival time of order sequence l in waiting list from vendor k, day
$\hat{t}_k^j$ = waiting delay time due to previous operation for purchase order of material j from vendor k, day
${}^{`}\hat{t}_i$ = waiting delay time due to previous operation for process i, day
$\breve{t}_k^j$ = advanced delay time for purchase order of material j from vendor k, day
$\breve{t}^i$ = advanced delay time for process i, day
$\left(t_k^j\right)^+$ = defined by $t_k^j-\left(\breve{y}_k^j+\hat{y}_k^j\right)\omega_k^j=\left(t_k^j\right)^+-\left(t_k^j\right)^-$ $\left(t_k^j\right)^+\ge 0$, $\left(t_k^j\right)^-\ge 0$
$\left(t_k^j\right)^-$ = defined by $t_k^j-\left(\breve{y}_k^j+\hat{y}_k^j\right)\omega_k^j=\left(t_k^j\right)^+-\left(t_k^j\right)^-$ $\left(t_k^j\right)^+\ge 0$, $\left(t_k^j\right)^-\ge 0$
$({}^{`}t_i)^+$ = defined by ${}^{`}t_i-\left({}^{`}\breve{y}_i+{}^{`}\hat{y}_i\right)\omega_i=({}^{`}t_i)^+-({}^{`}t_i)^-$ $({}^{`}t_i)^+\ge 0$, $({}^{`}t_i)^-\ge 0$
$({}^{`}t_i)^-$ = defined by ${}^{`}t_i-\left({}^{`}\breve{y}_i+{}^{`}\hat{y}_i\right)\omega_i=({}^{`}t_i)^+-({}^{`}t_i)^-$ $({}^{`}t_i)^+\ge 0$, $({}^{`}t_i)^-\ge 0$
TC = total cost, \$/day
V^j(time) = inventory level of material j at time, (L), control output
$\tilde{V}_k^j$ = total purchasing quantity of material j from vendor k during the lead time, L
$\tilde{V}_{kl}^j$ = purchase order quantity of product j from vendor k in l-th waiting list, L
$\hat{V}_k^j$ = partial quantity of the current purchasing order of material j from vendor i, L

$\hat{V}^j_m$ = partial volumes of the current demand order of material j to customer m, L

$`\hat{V}_{,i}$ = partial volumes of the current batch feeding to process i, L

$\hat{V}'_i$ = partial volumes of the current batch discharging from process i, L

$\underline{\underline{V^j}}$ = lower bound of V^j(time), L

$\overline{\overline{V^j}}$ = upper bound of V^j(time), L

$\overline{V^j}$ = average of V^j(time), L

$\underline{\underline{{}^*V^j}}$ = true lower bound of V^j(time) under conditions of uncertainty, L

W = set of warehouses

$`x_i$ = SOTF, storage operation time divided by cycle time for feeding flow to process i

x'_i = SOTF, storage operation time divided by cycle time for discharging flow to process i

x^j_k = SOTF, storage operation time divided by cycle time for purchase order of material j from vendor k

$\mathbf{x}^j_m$ = random SOTF for the order from customer m of material j

Greek letters

α^j_m = defined by $\alpha^j_m \equiv \frac{\omega^j_m}{\underline{\underline{\omega^j_m}}}$

β^j_m = defined by $\beta^j_m \equiv \frac{B^j_m}{\underline{\underline{B^j_m}}}$

δ_1, δ_2 = confidence levels

ζ = a small positive number

$\varepsilon_1, \varepsilon_2$ = convergence limits

$\underline{\underline{\lambda^j}}$ = Lagrangian multipliers for safety level constraints

$\overline{\overline{\lambda^w}}$ = Lagrangian multiplier for warehouse capacity

ϕ = forgetting factor $0 < \phi \le 1$

Ψ^j_k = aggregated cost parameter defined as $\Psi^j_k = \left(\frac{H^j}{2} + b^j\right)(1 - x^j_k) + a^j_k$, \$/L

Ψ_i = aggregated cost parameter defined as $\Psi_i \equiv a_i \sum_{j=1}^{|J|} f^j_i + (1 - `x_i) \sum_{j=1}^{|J|} \left(\frac{H^j}{2} + b^j\right) f^j_i + (1 - x'_i) \sum_{j=1}^{|J|} \left(\frac{H^j}{2} + b^j\right) g^j_i$, \$/L

ρ^j_k = weighting factor for raw material purchase startup times in Problem 4

ρ_i = weighting factor for production startup times in Problem 4

ρ^j_m = weighting factor for customer demand startup times in Problem 4

o^j = tuning parameter in MPC Algorithm

θ^j_m = defined by $\theta^j_m = \left[\left(\frac{1}{\alpha^j_m} - 1\right) + \frac{\left(\alpha^j_m - \beta^j_m\right)^+}{2\alpha^j_m}\right] N$

τ = sampling interval, day

ω_i = cycle time of process i, day

ω^j_k = cycle time of purchase order for material j from vendor k, day

$\boldsymbol{\omega}^j_m$ = random cycle time of the order of customer m for material j, day

Literature Cited

1. Yi G, Reklaitis GV. Optimal design of batch-storage network with recycle streams. *AIChE J*. 2003;49:3084–3094.
2. Yi G, Reklaitis GV. Optimal design of batch-storage network with multitasking semi-continuous processes. *AIChE J*. 2006;52:269–281.
3. Li Z, Ierapetritou MG. Reactive scheduling using parametric programming. *AIChE J*. 2008;54:2610–2623.
4. Kopanos GM, Pistikopoulos EN. Reactive scheduling by a multiparametric programming rolling horizon framework: a case of a network of combined heat and power units. *Ind Eng Chem Res*. 2014; 53:4366−4386.
5. Subramanian K, Rawlings JB, Maravelias CT. Economic model predictive control for inventory managementin supply chains. *Comput Chem Eng*. 2014;64:71–80.
6. Sarimveisa H, Patrinosa P, Tarantilisb CD, Kiranoudisa CT. Dynamic modeling and control of supply chain systems: a review. *Comput Oper Res*. 2008;35:3530–3561.
7. Ignaciuk P, Bartoszewicz A. Linear–quadratic optimal control strategy for periodic-review inventory systems. *Automatica* 2010;46: 1982–1993.
8. Subramanian K, Maravelias CT, Rawlings JB. A state-space model for chemical production scheduling. *Comput Chem Eng*. 2012;47: 97–110.
9. Jung JY, Blaua G, Pekny JF, Reklaitis GV, Eversdyk D. Integrated safety stock management for multi-stage supply chains under production capacity constraints. *Comput Chem Eng*. 2008;32:2570–2581.
10. Cachon GP, Kok AG. Implementation of the newsvendor model with clearance pricing: how to (and how not to) estimate a salvage value. *Manuf Serv Oper Manage*. 2007;9:276–290.
11. Liu S, Shah N, Papageorgiou LG. Multiechelon supply chain planning with sequence-dependent changeovers and price elasticity of demand under uncertainty. *AIChE J*. 2012;58:3390–3403.
12. Yue D, Guillen-Gosalbez G, You F. Global optimization of large-scale mixed-integer linear fractional programming problems: a reformulation-linearization method and process scheduling applications. *AIChE J*. 2013;59:4255–4272.
13. Yi G, Reklaitis GV. Optimal design of generalized batch-storage network considering shortage. *AIChE J*. 2013;59:2454–2470.
14. Yi G, Lee B, Lee ES. Optimal design of batch-storage network under sporadic operating time loss. *Ind Eng Chem Res*. 2013;52: 7116−7126.
15. Yi G, Reklaitis GV. Model based feedback inventory controller for intermediate storage in batch processes. *J Process Control*. 1994;4: 227–237.
16. Yi G, Reklaitis GV. Adaptive model predictive inventory controller for multiproduct warehouse system. *Automatica* 2014;50:2245–2253.
17. Aumi S, Mhaskar P. An adaptive data-based modeling approach for predictive control of batch systems. *Chem Eng Sci*. 2013;91: 11–21.
18. Rosenthal RE. *GAMS—A User's Guide*. Washington, DC: GAMS Development Corporation, 2014:159–161.
19. Yi G. Optimal design of batch-storage network under joint uncertainties. *AIChE J*. 2008;54:2567–2580.

Appendix A: Solution of Kuhn–Tucker Conditions for Problem 1

The Lagrangian for Problem 1 is

$$L = \mathrm{TC} - \sum_{j=1}^{|J|} \underline{\underline{\lambda^j}} \left(\underline{\underline{V^j}} - \underline{\underline{S^j}} \right) - \sum_{w=1}^{|W|} \overline{\overline{\lambda^w}} \left(\overline{\overline{S^w}} - \sum_{j \in J(w)} \overline{\overline{V^j}} \right) \quad \text{(A1)}$$

Differentiating Eq. A1 with respect to ω^j_k, ω_i t^j_k, and $`t_i$ gives

$$\frac{\partial L}{\partial t^j_k} = \sum_{j=1}^{|J|} \left[H^j \frac{\partial \overline{V^j}}{\partial t^j_k} + b^j \frac{\partial \overline{\overline{V^j}}}{\partial t^j_k} - \underline{\underline{\lambda^j}} \frac{\partial \underline{\underline{V^j}}}{\partial t^j_k} \right] + \sum_{w=1}^{|W|} \sum_{j \in J(w)} \overline{\overline{\lambda^w}} \frac{\partial \overline{\overline{V^j}}}{\partial t^j_k}$$
$$= \sum_{j=1}^{|J|} \left[-H^j - b^j + \underline{\underline{\lambda^j}} - \overline{\overline{\lambda^w}} \right] D^j_k = 0 \quad \text{(A2)}$$

$$\frac{\partial L}{\partial `t_i} = \sum_{j=1}^{|J|} \left[ H^j \frac{\partial \overline{V^j}}{\partial `t_i} + b^j \frac{\partial \overline{\overline{V^j}}}{\partial `t_i} - \underline{\underline{\lambda^j}} \frac{\partial \underline{\underline{V^j}}}{\partial `t_i} \right] + \sum_{w=1}^{|W|} \sum_{j \in J(w)} \overline{\overline{\lambda^w}} \frac{\partial \overline{\overline{V^j}}}{\partial `t_i}$$
$$= \sum_{j=1}^{|J|} \left[-H^j - b^j + \underline{\underline{\lambda^j}} - \overline{\overline{\lambda^w}} \right] \left(f^j_i - g^j_i \right) D_i = 0 \quad \text{(A3)}$$

$$\frac{\partial L}{\partial \omega^j_k} = -\frac{A_i}{\left(\omega^j_k\right)^2} + a^j_k D^j_k + \sum_{j=1}^{|J|} \left[H^j \frac{\partial \overline{V^j}}{\partial \omega^j_k} + b^j \frac{\partial \overline{\overline{V^j}}}{\partial \omega^j_k} - \underline{\underline{\lambda^j}} \frac{\partial \underline{\underline{V^j}}}{\partial \omega^j_k} \right] + \sum_{w=1}^{|W|} \sum_{j \in J(w)} \overline{\overline{\lambda^w}} \frac{\partial \overline{\overline{V^j}}}{\partial \omega^j_k}$$
$$= -\frac{A^j_k}{\left(\omega^j_k\right)^2} + a^j_k D^j_k + \left[\left(\frac{H^j}{2} + b^j + \overline{\overline{\lambda^w}} \right) (1 - x^j_k) + a^j_k \right] D^j_k = 0 \quad \text{(A4)}$$

$$\frac{\partial L}{\partial \omega_i} = -\frac{A_i}{(\omega_i)^2} + a_i \sum_{j=1}^{|J|} f_i^j D_i + \sum_{j=1}^{|J|} \left[H^j \frac{\partial \overline{V^j}}{\partial \omega_i} + b^j \frac{\partial \overline{\overline{V^j}}}{\partial \omega_i} - \underline{\underline{\lambda^j}} \frac{\partial \underline{\underline{V^j}}}{\partial \omega_i} \right] + \sum_{w=1}^{|W|} \sum_{j \in J(w)} \overline{\overline{\lambda^w}} \frac{\partial \overline{\overline{V^j}}}{\partial \omega_i}$$

$$= -\frac{A_i}{(\omega_i)^2} + a_i \sum_{j=1}^{|J|} f_i^j D_i - \sum_{j=1}^{|J|} \frac{H^j}{2} \left[(1-\grave{x}_i) f_i^j + (1-x_i') g_i^j \right] D_i + \sum_{j=1}^{|J|} \underline{\underline{\lambda^j}} \left[(1-\grave{x}_i) f_i^j + (1-x_i') g_i^j \right] D_i = 0 = 0 \quad \text{(A5)}$$

Solving Eqs. A2 and A3 with $|J| \leq |I|$ gives

$$\underline{\underline{\lambda^j}} = H^j + b^j + \overline{\overline{\lambda^w}} \text{ and } \underline{\underline{V^j}} = \underline{\underline{S^j}} \quad \text{(A6)}$$

$$\overline{\overline{\lambda^w}} \left[\sum_{j \in J(w)} \left\{ \sum_{i=1}^{|I|} \left[(1-\grave{x}_i) f_i^j + (1-x_i') g_i^j \right] D_i \omega_i + \sum_{k=1}^{|K(j)|} (1-x_k^j) D_k^j \omega_k^j + \sum_{m=1}^{|M(j)|} (1-x_m^j) \overline{B_m^j} \right\} - \overline{\overline{S^w}} \right] = 0 \quad \text{(A7)}$$

Solving Eqs. A4 and A5 gives Eqs. 18 and 19 in the main text. Inserting Eqs. 18 and 19 into Eq. A1 gives Eq. 21 in the main text. Equation A7 is Eq. 22.

Appendix B: Proof of Theorem 2

Proof: According to the reasoning in the previous study,[19] the lower and upper linear bounds of $\int_0^t \mathbf{F}_m^j(\tau) d\tau$ are

$$\overline{B_m^j} \left[\frac{t - \hat{t}_m^j}{\overline{\overline{\omega_m^j}}} - \alpha_m^j \theta_m^j \right] \leq \int_0^t \mathbf{F}_m^j(\tau) d\tau \leq \overline{B_m^j} \left[\frac{t - \hat{t}_m^j}{\overline{\overline{\omega_m^j}}} + (1-x) \alpha_m^j + \alpha_m^j \theta_m^j \right] \quad \text{(B1)}$$

where $\mathbf{F}_m^j(\tau)$ is composed of $\mathbf{B}_m^j(n)$ and $\boldsymbol{\omega}_m^j(n)$. Then, the true lower bound of $V^j(t)$ under uncertainty is

$$\underline{\underline{{}^*V^j}} = V^j(0) + \sum_{k=1}^{|K(j)|} \left[\tilde{V}_k^j(\tilde{t}_k^j) + \hat{V}_k^j(\hat{t}_k^j) \right] + \sum_{i=1}^{|I|} g_i^j \tilde{V}'_i(t_i') - \sum_{i=1}^{|I|} f_i^j \grave{V}_i(\grave{t}_i) - \sum_{m=1}^{|M|} \hat{V}_m^j(\hat{t}_m^j) - \sum_{i=1}^{|I|} \left[(1-\grave{x}_i) f_i^j + g_i^j (1-x_i') \right] B_i + \sum_{i=1}^{|I|} \left(f_i^j - g_i^j \right) B_i \frac{\grave{t}_i}{\omega_i}$$
$$- \sum_{m=1}^{|M|} \overline{B_m^j} \left[(1-x_m^j) - \frac{\hat{t}_m^j}{\overline{\overline{\omega_m^j}}} \right] - \sum_{k=1}^{|K(j)|} B_k^j \frac{t_k^j + \tilde{t}_k^j}{\omega_k^j} - \sum_{m=1}^{|M|} \overline{B_m^j} \left[\alpha_m^j \theta_m^j - (1-x_m^j)(1-\alpha_m^j) \right] \quad \text{(B2)}$$

The control input is computed based on Eq. 24 and the control error $\| \underline{\underline{S^j}} - \underline{\underline{{}^*V^j}} \|$ is computed from Eqs. 24 and B2.

$$|\underline{\underline{S^j}} - \underline{\underline{{}^*V^j}}| = \left| -\sum_{m=1}^{|M|} \overline{B_m^j}(n) \left[(1-x_m^j) - \frac{t_m^j - \hat{t}_m^j(n) + \overline{\omega_m^j}(n) - \tilde{t}_m}{\overline{\overline{\omega_m^j}}(n)} \right] + \sum_{m=1}^{|M|} \overline{B_m^j} \left[(1-x_m^j) - \frac{\overline{\omega_m^j} - \tilde{t}_m}{\overline{\overline{\omega_m^j}}} \right] + \sum_{m=1}^{|M|} \overline{B_m^j} \left[\alpha_m^j \theta_m^j - (1-x_m^j)(1-\alpha_m^j) \right] \right|$$

$$= \left| \sum_{m=1}^{|M|} \left[\overline{B_m^j}(n) - \overline{B_m^j} \right] (x_m^j) - \sum_{m=1}^{|M|} \tilde{t}_m \left[\frac{\overline{B_m^j}(n)}{\overline{\overline{\omega_m^j}}(n)} - \frac{\overline{B_m^j}}{\overline{\overline{\omega_m^j}}} \right] + \sum_{m=1}^{|M|} \overline{B_m^j} \left[\alpha_m^j \theta_m^j - (1-x_m^j)(1-\alpha_m^j) \right] + \sum_{m=1}^{|M|} \overline{B_m^j}(n) \frac{t_m^j - \hat{t}_m^j(n)}{\overline{\overline{\omega_m^j}}(n)} \right|$$

$$\leq \left| \left| \sum_{m=1}^{|M|} \left[\overline{B_m^j}(n) - \overline{B_m^j} \right] (x_m^j) \right| + \left| \sum_{m=1}^{|M|} \tilde{t}_m \left[\frac{\overline{B_m^j}(n)}{\overline{\overline{\omega_m^j}}(n)} - \frac{\overline{B_m^j}}{\overline{\overline{\omega_m^j}}} \right] \right| \right| + \left| \sum_{m=1}^{|M|} \overline{B_m^j} \left[\alpha_m^j \theta_m^j - (1-x_m^j)(1-\alpha_m^j) \right] \right| + \sum_{m=1}^{|M|} \overline{B_m^j}(n) \frac{t_m^j - \hat{t}_m^j(n)}{\overline{\overline{\omega_m^j}}(n)}$$

$$= \left| \sum_{m=1}^{|M|} \left[\overline{B_m^j}(n) - \overline{B_m^j} \right] (x_m^j) \right| + \left| \sum_{m=1}^{|M|} \tilde{t}_m \left[\frac{\overline{B_m^j}(n) \overline{\omega_m^j} - \overline{B_m^j} \overline{\omega_m^j}(n)}{\overline{\overline{\omega_m^j}}(n) \overline{\overline{\omega_m^j}}} \right] \right| + \left| \sum_{m=1}^{|M|} \overline{B_m^j} \left[\alpha_m^j \theta_m^j - (1-x_m^j)(1-\alpha_m^j) \right] \right| + \sum_{m=1}^{|M|} \overline{B_m^j}(n) \frac{t_m^j - \hat{t}_m^j(n)}{\overline{\overline{\omega_m^j}}(n)}$$

$$
\begin{aligned}
&= \left| \sum_{m=1}^{|M|} \left[\overline{B_m^j}(n) - \overline{B_m^j} \right] (x_m^j) \right| + \left| \sum_{m=1}^{|M|} \tilde{t}_m \left[\frac{\overline{\omega_m^j} \left[\overline{B_m^j}(n) - \overline{B_m^j} \right] + \overline{B_m^j} \left[\omega_m^j - \overline{\omega_m^j}(n) \right]}{\underline{\underline{\omega_m^j}}\, \overline{\omega_m^j}} \right] \right| \\
&+ \left| \sum_{m=1}^{|M|} \overline{B_m^j} \left[\alpha_m^j \theta_m^j - (1 - x_m^j)(1 - \alpha_m^j) \right] \right| + \left| \sum_{m=1}^{|M|} \left(t_m^j - \hat{t}_m^j(n) \right) \left[\frac{\overline{B_m^j}(n)}{\overline{\omega_m^j}(n)} - \frac{\overline{B_m^j}}{\overline{\overline{\omega_m^j}}} \right] \right| + \sum_{m=1}^{|M|} \frac{\overline{B_m^j}}{\overline{\overline{\omega_m^j}}} \left(t_m^j - \hat{t}_m^j(n) \right) \\
&\le \sum_{m=1}^{|M|} \varepsilon_1 (x_m^j) + \sum_{m=1}^{|M|} \tilde{t}_m \left[\frac{\overline{\omega_m^j} \varepsilon_1 + \overline{B_m^j} \varepsilon_2}{\underline{\underline{\omega_m^j}}\, \overline{\omega_m}} \right] + \left| \sum_{m=1}^{|M|} \overline{B_m^j} \left[\alpha_m^j \theta_m^j - (1 - x_m^j)(1 - \alpha_m^j) \right] \right| \\
&+ \left| \sum_{m=1}^{|M|} \left(t_m^j - \hat{t}_m^j(n) \right) \left[\frac{\overline{\omega_m} \varepsilon_1 + \overline{B_m^j} \varepsilon_2}{\underline{\underline{\omega_m}}\, \overline{\omega_m}} \right] \right| + \sum_{m=1}^{|M|} \frac{\overline{B_m^j}}{\overline{\overline{\omega_m^j}}} \left(t_m^j - \hat{t}_m^j(n) \right)
\end{aligned}
\tag{B3}
$$

For $n \ge N$, Eq. 27 gives

$$
\begin{aligned}
\left| \underline{\underline{S^j}} - \underline{\underline{V^j}} \right| &\le \sum_{m=1}^{|M|} \varepsilon_1 (x_m^j) + \sum_{m=1}^{|M|} \tilde{t}_m \left[\frac{\overline{\omega_m^j} \varepsilon_1 + \overline{B_m^j} \varepsilon_2}{\underline{\underline{\omega_m^j}}\, \overline{\omega_m}} \right] \\
&+ \left| \sum_{m=1}^{|M|} \overline{B_m^j} \left[\alpha_m^j \theta_m^j - (1 - x_m^j)(1 - \alpha_m^j) \right] \right| \\
&+ \left[\max_m \left\{ \frac{\overline{\omega_m^j} \varepsilon_1 + \overline{B_m^j} \varepsilon_2}{\overline{B_m^j}\, \underline{\underline{\omega_m^j}}} \right\} + 1 \right] \sum_{m=1}^{|M|} \overline{D_m^j} \left(t_m^j - \hat{t}_m^j(n) \right)
\end{aligned}
\tag{B4}
$$

To find the upper bound of $\sum_{m=1}^{|M|} \overline{D_m^j} \left(t_m^j - \hat{t}_m^j(n) \right)$, consider the worst case of $0.5N$ times of maximum batch size $\overline{\overline{B_m^j}}$ with minimum cycle time $\underline{\underline{\omega_m^j}}$, $0.5N$ times of minimum batch size $\underline{\underline{B_m^j}}$ with minimum cycle time $\underline{\underline{\omega_m^j}}$, and a total dead time $\left(\frac{1}{\alpha_m^j} - 1 \right) N \underline{\underline{\omega_m^j}}$ within repeated long cycle times. Then, for $n \ge N$, Eq. 24 gives

$$
\begin{aligned}
&\sum_{m=1}^{|M|} \left[\left(\frac{\overline{\overline{B_m^j}}}{\underline{\underline{\omega_m^j}}} - \overline{D_m^j} \right) \frac{N \underline{\underline{\omega_m^j}}}{2} + \max \left\{ 0, \left(\frac{\underline{\underline{B_m^j}}}{\underline{\underline{\omega_m^j}}} - \overline{D_m^j} \right) \right\} \frac{N \underline{\underline{\omega_m^j}}}{2} + 2 \overline{\overline{B_m^j}} \right] \\
&> \sum_{m=1}^{|M|} \overline{D_m^j} \left(\overline{\overline{t_m^j}} - \hat{t}_m^j(n) \right)
\end{aligned}
\tag{B5}
$$

This proves Theorem 2.

Manuscript received Sep. 20, 2014, and revision received Feb. 20, 2015.