An Attainable Region Approach for Production Planning of Multiproduct Processes

Charles Sung and Christos T. Maravelias
Dept. of Chemical and Biological Engineering, University of Wisconsin-Madison, Madison, WI 53706

DOI 10.1002/aic.11167
Published online April 3, 2007 in Wiley InterScience (www.interscience.wiley.com).

A novel approach is presented for the solution of production planning problems for multiproduct processes. A mixed-integer programming (MIP) scheduling model is analyzed off-line to obtain a convex approximation of feasible production levels and a convex underestimation of total production cost as a function of production levels. The two approximating functions are expressed via linear inequalities that involve only planning variables yet provide all the relevant scheduling information necessary to solve the planning problem with high quality. A rolling horizon algorithm is also presented for generation (if necessary) of detailed schedules. © 2007 American Institute of Chemical Engineers *AIChE J,* 53: 1298–1315, 2007
Keywords: production planning, scheduling, attainable region

Introduction

In 2005, the US chemical industry contributed over 2.3%, $540 billion,[1] to the gross domestic product (GDP). It is the second largest manufacturing sector and one of the largest private sector investors in R&D, with chemical patents accounting for 21% of total awarded patents in 2004.[2] At the same time, however, the chemical industry faces a number of major challenges such as migration of customer industries, saturation of markets, increased global competition, continued environmental regulation, and energy price and availability.[3,4] To remain healthy in today's competitive environment, chemical companies must operate efficiently by simultaneously optimizing multiple levels of operation.[5,6] To achieve this, advanced modeling methods and optimization tools for the integration and solution of large-scale supply chain optimization models are necessary.

In this article, we are specifically interested in medium-term production planning of multiproduct processes. Production planning seeks to determine optimal production targets and product inventories for a given demand forecast. To effectively solve this problem, we should also account for the capacity of manufacturing facilities and production costs, which is currently achieved via the integration of planning and scheduling models. However, integrated models are notoriously difficult to solve despite improvements in computer hardware and optimization software. Current state of the art methods are insufficient to meet industrial needs, while ad hoc approaches are not adequate due to the complexity of the chemical manufacturing facilities (e.g., batch splitting/mixing, recycle streams, coproduction, different storage policies, utility constraints, etc.). To address this problem, we propose developing a set of constraints to accurately define the space of feasible production levels (amounts) and associated production costs. The strength of this framework is being able to locate feasible production levels without needing to *on-line* obtain a detailed schedule for achieving each target.

This article is organized as follows. We begin by reviewing the area of supply chain management (SCM), defining the production planning problem, discussing integration of planning and scheduling, and reviewing previously proposed methods. Next, we introduce the idea of process attainable region and present an algorithm for generating constraints to describe this region. Generalizations of the method are also provided. Finally, we present a rolling horizon algorithm for the generation of detailed schedules over long planning horizons.

Background

SCM in the chemical industry

A supply chain (SC) is a network of facilities and distribution options that performs the following functions: procure-

Correspondence concerning this article should be addressed to C. T. Maravelias at maravelias@wisc.edu.

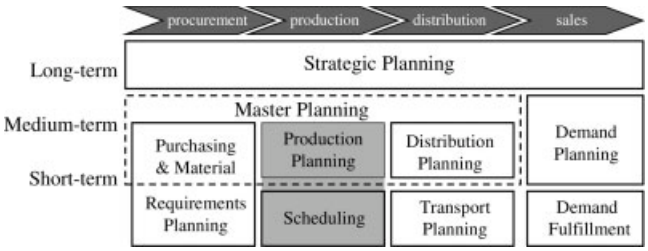

Figure 1. Supply chain planning matrix used by software systems.

Modified from Meyr.[9]

ment of raw materials (RMs), transformation of RM into intermediate and finished products, and distribution of finished products to customers. In 2003, inventory in US supply chains was about 10% of the GDP, or almost $1 trillion, resulting in 50%+ inefficiency.[7] This implies that there is a tremendous opportunity to improve the economy by reducing inventories. The current environment places an even greater premium on reducing inventories in industrial sectors with small profit margins such as the commodity portion of the chemical industry. The goal of SCM is the integration of organizational units to coordinate material, information, and financial flows to improve the competitiveness of the SC as a whole.[8,9]

The SC planning matrix shown in Figure 1 includes models for procurement, production, distribution, and sales decisions along the x-axis, and models for long-, medium-, and short-term decisions in the y-axis.* Because of the interconnections between different levels of the supply chain and the interdependence of the decisions made at the various geographical locations, SC decisions should be coordinated via integration. Strategic or long-term planning affects the (infra-) structure of the supply chain and is carried out over a horizon of several years. Examples of strategic decisions are the location of production sites and warehouses, the capacity of these facilities, and transportation methods and routes. Master or medium-term planning seeks to most efficiently fulfill customer demand over a medium-term horizon of months. Decisions involve the assignment of production targets to production sites, transportation amounts, and inventory profiles. Finally, short-term planning is carried out for each individual site upon receiving targets from master planning. At the production level, short-term planning is referred to as scheduling. The scheduling horizon is in days or weeks, and scheduling decisions include batch-sizing, assignment of tasks to equipment units, and sequencing of tasks.

In this article, we are primarily interested in addressing the medium-term production planning problem subject to scheduling constraints. The financial incentives for better production planning are substantial: Exxon Chemicals estimated that a major initiative reduced annual operating costs by 2% and operating inventory by 20%, while a single polymers business of DuPont was able to reduce its working capital tied up in inventory from $165 to $90 million.[10] Production planning becomes even more important as clients become more demanding and pressure builds for maximizing the productivity of existing assets.

This problem is challenging because in process industries production planning decisions may be limited by short-term scheduling constraints. To determine optimal production planning targets, it would seem necessary to combine production planning and scheduling (highlighted in Figure 1) in an integrated model. However, this may be computationally expensive or intractable. Essentially, scheduling models define a combinatorially immense number of (feasible) detailed schedules. Each detailed schedule corresponds to a feasible production amount. Production amounts are either feasible because at least one detailed schedule corresponds to it, or are infeasible because no detailed schedule corresponds to it. Ideally, however, we would like to solve the production planning problem without solving an integrated planning-scheduled model.

Production planning

The goal in production planning is to meet customer demand at minimum total (i.e., production + holding + backlog) cost over a medium-term time horizon. Given are

- A fixed planning horizon H (2–12 months), divided into planning periods $t \in \{1, 2, \ldots, T\}$ of uniform duration $H_t = H/T$.
- A set of final products $k \in K^{\mathrm{FP}} = \{1, 2, \ldots, K\}$.
- Demand Dem_{kt} for product k at the end of the planning period t.
- Holding cost h_k and penalty u_k for unmet demand for product k.
- Process capacities.
- Production costs.

Decision variables include

- The production target P_{kt} of product k in period t.
- The inventory level I_{kt} of product k in period t.
- The shipment level D_{kt} of product k in period t.
- Unmet (backlogged) demand U_{kt} for product k in period t.
- Total cost CT.

Backlogged demand is modeled in order to (a) penalize outstanding orders and (b) ensure feasibility of the planning problem. The objective is minimization of the total cost CT. If revenue from sales is considered, then the objective is the maximization of profit.

A general formulation for production planning involves constraints 1–8. Feasible production targets are implicitly modeled via function $F(P_{kt})$ in Eq. 2, and production cost Cp_t in period t is expressed via function $C(P_{kt})$ in Eq. 3. Holding cost Ch_t and backlog cost Cu_t are calculated in Eqs. 4 and 5, respectively. Inventory and backlog flow balance constraints for product k at the end of period t are maintained in Eqs. 6 and 7, respectively. We will refer to this production planning model as (PP1).

$$\min \mathrm{CT} = \sum_{t \in T} (\mathrm{Cp}_t + \mathrm{Ch}_t + \mathrm{Cu}_t) \tag{1}$$

$$F(P_{kt}) \leq 0 \quad \forall t \tag{2}$$

$$\mathrm{Cp}_t = C(P_{kt}) \quad \forall t \tag{3}$$

*Note that the models in the SC planning matrix differ also in terms of industrial sectors. For example, distribution planning in high-value, low-volume goods such as pharmaceuticals is quite different than distribution planning in commodities such as crude oil.

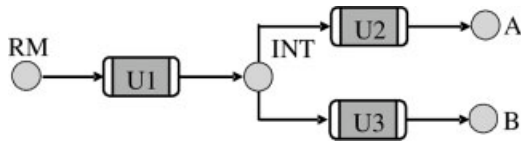

Figure 2. Process network of motivating example.

$$\mathrm{Ch}_t = \sum_k h_k I_{kt} \quad \forall t \tag{4}$$

$$\mathrm{Cu}_t = \sum_{k \in K^{FP}} u_k U_{kt} \quad \forall t \tag{5}$$

$$I_{kt} = I_{kt-1} + P_{kt} - D_{kt} \quad \forall k, t \tag{6}$$

$$U_{kt} = U_{kt-1} + Dem_{kt} - D_{kt} \quad \forall k, t \tag{7}$$

$$P_{kt}, I_{kt}, D_{kt}, U_{kt} \geq 0 \quad \forall k, t \tag{8}$$

Since functions $F(P_{kt})$ and $C(P_{kt})$ are not readily available, constraints that approximate process capacity and total production cost are employed. For example, in discrete manufacturing industries, aggregate capacity constraints based on bottleneck units are often used. However, these approximate representations do not capture the complexities of chemical plants, leading to production targets that may be infeasible or suboptimal.

Motivating example

To illustrate why aggregate constraints are insufficient, we consider the process network in Figure 2. Raw material (RM) is converted by unit U1 into unstable intermediate INT, which is immediately converted by unit U2 into final product A or by unit U3 into final product B. The maximum rate of unit U1 is 300 kg/day. For technical reasons, U1 is operated above rate 200 kg/day or else turned off. Note that U1 is able to achieve an *average* rate lower than 200 kg/day by being turned off for some time. Similarly, unit U2 is operated between 150 and 200 kg/day or else turned off. Unit U3 is operated at 150 kg/day or else turned off. All three units have a residence time of 1 h.

Given the topology of Figure 2, unit U1 appears to be the bottleneck of the process network. The maximum processing rate of unit U1 is used to develop the weekly aggregate capacity constraint shown in Figure 3a. Treating every unit as a potential bottleneck, we derive a tighter region in Figure 3b. However, the true feasible production region is the one shown in Figure 3c. Lack of storage at INT causes A to be produced in equal amount whenever B is produced and prevents units U1 and U2 from operating at full capacity simultaneously. Also, the residence time of unit U1 causes units U2 and U3 to sit idle for the first hour, which is not accounted for by aggregate capacity constraints. The regions shown in Figures 3a, b overestimate the true feasible region by 203% and 94%, respectively.

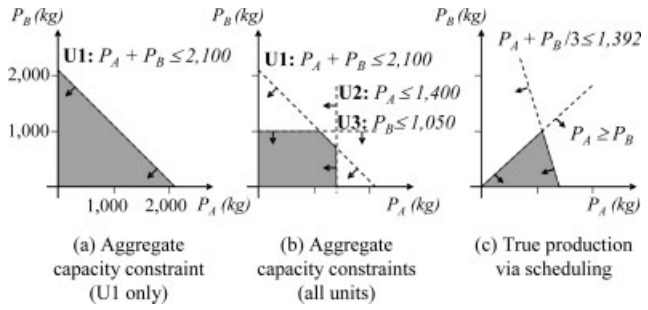

Figure 3. Alternative attainable regions of process network of Figure 2: (a) aggregate capacity constraint (U1 only), (b) aggregate capacity constraints (all units), and (c) true production via scheduling.

Integration with scheduling

To address the limitations of aggregate representations, researchers have proposed large-scale integrated models where a detailed scheduling model is integrated with a production planning model. The advantage of these models is that production targets P_{kt} are feasible and total production costs Cp_t are accurate because they satisfy a detailed schedule. To illustrate integration, here, we briefly present the State-Task Network (STN) formulation[11] for discrete-time scheduling. To keep the presentation simple, we do not consider changeovers and utility requirements.

Chemicals ("states") $k \in K$ are transformed from RMs to final products by a series of tasks, which can be batch $i \in I^{\mathrm{B}}$ or continuous $i \in I^{\mathrm{C}}$. Each task consumes specific states and produces different states. The number of batches (i.e., number of instances that a task is run), their size, their assignment to a single unit $j \in J$, and their sequencing are all determined as part of the optimization. Each planning period $t \in \{1, 2, \ldots, T\}$ is divided into N scheduling periods of length $\Delta t = H_t/N$, thus defining $N + 1$ time points $n \in \{0, 1, 2, \ldots, N\}$ (Figure 4). Time points t of the planning problem correspond to a *big-bucket* (1–4 weeks) planning time grid, while the time points n of the scheduling problem correspond to a *small-bucket* (1–24 h) scheduling time grid (Figure 4).

Batch tasks $i \in I^{\mathrm{B}}$ start on unit j at time point n if W_{ijn} is equal to one. Batch tasks finish at time point $n + \tau_i$ where $\tau_i \Delta t$ is the processing time of task i. Equation 9 forbids batch tasks that would finish beyond the scheduling horizon, where $I(j)$ is the set of tasks i that can be carried out in unit j:

$$W_{ijn} = 0 \quad \forall j \in J \quad i \in I^{\mathrm{B}} \cap I(j) \quad n > N - \tau_i \tag{9}$$

Continuous tasks $i \in I^{\mathrm{C}}$ are processed on unit j during time period following n if W_{ijn} is equal to 1. The residence time

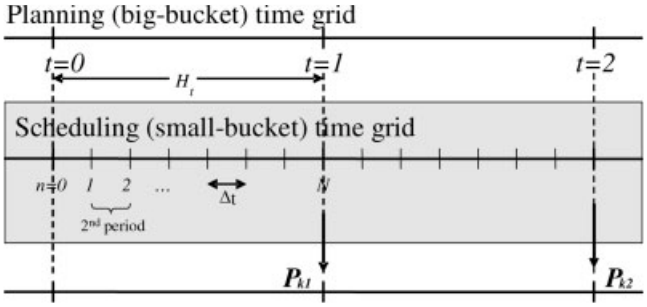

Figure 4. Time grids for planning and scheduling.

of continuous tasks is assumed to be one-time period. Equation 10 forbids continuous tasks that would finish beyond the scheduling horizon.

$$W_{ijn} = 0 \quad \forall j \in J, \quad i \in I^{\mathrm{C}} \cap I(j), \quad n = N \tag{10}$$

During each time period, each unit j can be assigned to only one task:

$$\sum_{i \in I^{\mathrm{B}} \cap I(j)} \sum_{n' \geq n-\tau_i+1}^{n' \leq n} W_{ijn'} + \sum_{i \in I^{\mathrm{C}} \cap I(j)} W_{ijn} \leq 1 \quad \forall j \in J, n \tag{11}$$

The batch size B^{B}_{ijn} of a batch task is restricted between the minimum V^{MIN}_{ij} and maximum V^{MAX}_{ij} capacity, while the amount B^{C}_{ijn} processed by a continuous task is restricted by the minimum r^{MIN}_{ij} and maximum r^{MAX}_{ij} processing rate:

$$V^{\mathrm{MIN}}_{ij} W_{ijn} \leq B^{\mathrm{B}}_{ijn} \leq V^{\mathrm{Max}}_{ij} W_{ijn} \quad \forall j \in J, \ i \in I^{\mathrm{B}} \cap I(j), n \tag{12}$$

$$(r^{\mathrm{MIN}}_{ij} \Delta t) W_{ijn} \leq B^{\mathrm{C}}_{ijn} \leq (r^{\mathrm{Max}}_{ij} \Delta t) W_{ijn} \quad \forall j \in J \quad i \in I^{\mathrm{C}} \cap I(j), n \tag{13}$$

The inventory S_{kn} of chemical (state) k in time point n is given by Eq. 14, where ρ_{ik} is the stoichiometric coefficient of chemical k in task i, $I(k)/O(k)$ is the set of tasks consuming/producing chemical k, and S^{MAX}_k is the storage capacity dedicated to chemical k:

$$S_{kn} = S_{k,n-1} + \sum_{j \in J} \left[\sum_{i \in I^{\mathrm{B}} \cap I(j) \cap O(k)} \rho_{ik} B^{\mathrm{B}}_{ij,n-\tau_1} + \sum_{i \in I^{\mathrm{B}} \cap I(j) \cap I(k)} \rho_{ik} B^{\mathrm{B}}_{ijn} + \sum_{i \in I^{\mathrm{C}} \cap I(j) \cap O(k)} \rho_{ik} B^{\mathrm{C}}_{ij,n-1} + \sum_{i \in I^{\mathrm{C}} \cap I(j) \cap I(k)} \rho_{ik} B^{\mathrm{C}}_{ijn} \right] \leq S^{\mathrm{MAX}}_k \quad \forall k, n \tag{14}$$

Finally, total production cost is calculated by Eq. 15 where we assume that RM and utility costs are lumped into a fixed and a variable term for each task i:

$$\mathrm{Cp}_t = \sum_{j \in J} \sum_{i \in I(j)} \sum_{n \in N} \left(\alpha_i W_{ijn} + \beta^{\mathrm{B}}_i B^{\mathrm{B}}_{ijn} + \beta^{\mathrm{C}}_i B^{\mathrm{C}}_{ijn} \right) \tag{15}$$

The scheduling model (S1) consists of Eqs. 9–16.

$$W_{ijn} \in \{0,1\}, B^{\mathrm{B}}_{ijn}, B^{\mathrm{C}}_{ijn}, S_{kn} \geq 0 \quad \forall j \in J \quad i \in I(j), n \tag{16}$$

Model (S1) is linked to production planning via Eq. 17, which, for every planning period, sets the ending inventory of each final product in the scheduling model equal to its production level in the planning model. A separate scheduling model is formulated for each planning period, so the parameters, variables, and equations of model (S1) are all implicitly indexed by t.

$$P_{kt} = S_{kn} \quad \forall k \in K^{\mathrm{FP}}, t, n = N \tag{17}$$

Alternatively, if the production planning problem is modeled such that a single scheduling model spans the entire planning horizon, then index n should be expanded to $n \in \{0, 1, 2, \ldots, NT\}$, RMs deliveries need to be formulated, and Eq. 17 should be replaced with the following:

$$P_{kt} = S_{kn} \quad \forall k \in K^{\mathrm{FP}}, t, n = tN$$

The integrated planning-scheduling formulation (PP2) consists of Eqs. 1 and 4–17 where Eqs. 9–17 replace generic Eqs. 2 and 3 to accurately describe what production targets P_{kt} are feasible and determine total production cost Cp_t. Binary scheduling variable W_{ijn} is indexed by i, j, and n (and t), causing the integrated formulation to be a large-scale mixed-integer programming (MIP) model. Such models are tractable for simple process networks over short planning horizons. However, short planning horizons lead to myopic solutions that cannot account for future demand peaks, scheduled maintenance, and (infrequent) transportation dates.

It is interesting to note here that integrated planning-scheduling models were developed as a means to obtain feasible production targets, not as a means for having detailed schedules for several months. The actual schedule is dynamic and will be updated as demand and supplies change and production upsets occur. Therefore, what we want is a surrogate model that can accurately and more tractably replace the scheduling model.

Literature review

Researchers have developed integrated planning-scheduling models and decomposition schemes for their practical solution. Direct optimization of integrated (full-space) planning-scheduling models is theoretically optimal but generally intractable. Most schemes solve planning and scheduling models separately to take advantage of the different frequencies in which planning and scheduling decisions are made and different time scales in which they have effect. Generally, the planning model is solved first using an approximation of the scheduling model; if a strict over-approximation is used, then the approximated model provides a bound on the full-space model.[12] Next, the scheduling model is solved to give a feasible solution that is "optimal" relative to the planning solution. Hierarchal decompositions terminate here. However, some proposed schemes can update their planning and scheduling models such that the bound and feasible solution are improved with each reoptimization.

Graves[13] presents a Lagrangean decomposition for separating a LP planning-scheduling model. Birewar and Grossmann[14] present a planning model with multistage scheduling within each planning time period. Since cycle times for producing a single batch and slack times for consecutive cycles can be calculated off-line for specific multistage instances, the planning model is solved for the number of batches in each period, subject to total cycle and slack time (instead of makespan) being constrained to be less than each planning time period. Wellons and Reklaitis[15] present a planning–scheduling model using campaigns identified off-line as being dominant. Papageorgiou and Pantelides[16] present a model and hierarchal decomposition for scheduling same-cycle campaigns. Karimi and McDonald[17] present two models

for scheduling semicontinuous batches subject to due dates. Zhu and Majozi[18] present a planning model linked by production targets to State Sequence Network scheduling models and comment on the block angular structure. A hierarchal decomposition is proposed in which the planning problem is solved to assign production targets, subject to aggregate capacity constraints. Romero et al.[19] present a financial-planning model with detailed single-stage scheduling in the first planning time period and sequence-independent, approximated scheduling in latter time periods. Guillén et al.[20] present a variation using STN scheduling. Erdirik-Dogan and Grossmann[21] present a model and an iterative decomposition scheme for planning continuous batches on a single unit subject to due dates. Wilkinson et al.[22] present a temporal aggregation scheme for overapproximating Resource-Task Network scheduling models. Dimitriadis et al.[23] present a rolling horizon algorithm for this. Wan et al.[24] use simulation to approximate individual models as support vector machines and optimize a supply chain using these approximations.

From an industrial perspective, Shobrys and White[10] discuss the integration of planning, scheduling, and control, and McKay and Wiers[25] discuss the integration of scheduling software (e.g. APS) with larger Manufacturing/Enterprise Resource Planning (MRP and ERP) systems. Lasschuit and Thijssen[26] describe integrating strategic and operational planning at Shell, and Berning et al.[27] discuss an integration system for SCM at Bayer. Other integrated models and decompositions are available as well, and not limited to planning and scheduling. Kallrath[28] discusses the integration of strategic and operational planning, and Iyer and Grossmann[29] integrate capacity expansion decisions with planning. Extended discussions on integrating, planning, and scheduling can be found in Shah,[30] Kallrath,[31] Pinto et al.,[32] and Bodington.[33]

Proposed Framework

Our goal is the development of a systematic framework for obtaining, from a scheduling model, production feasibility and cost information in a form that is amenable to integration with the production planning problem. We wish to develop approximating functions $F(P_{kt})$ and $C(P_{kt})$ using linear inequalities that involve only planning variables P_{kt} and Cp_t. These inequalities define the Process Attainable Region (PAR) of the process network.

To illustrate how this can be achieved, we revisit the motivating example in Figure 2. The true feasible region shown in Figure 3c can be equivalently defined using only three inequalities: $P_\mathrm{A} \geq P_\mathrm{B}$, $P_\mathrm{A} + P_\mathrm{B}/3 \leq 1392$, and $P_\mathrm{B} \geq 0$. Despite involving only variables P_A and $\mathrm{P_B}$, these three inequalities provide the same feasibility information as Eqs. 9–17 of a detailed MIP scheduling model involving hundreds of variables and constraints. These inequalities can be expressed in the form shown in Eq. 18, where w_k^l is the coefficient for product k in the lth constraint, and Π^l is the RHS of the lth constraint:

$$\sum_{k \in K^{\mathrm{FP}}} w_k^l \, P_{kt} \leq \Pi^l \quad \forall l \in L_F \tag{18}$$

Similarly, linear inequalities can be formulated to provide a convex underestimation of total production cost Cp_t in terms of planning levels P_{kt} (Figure 5):

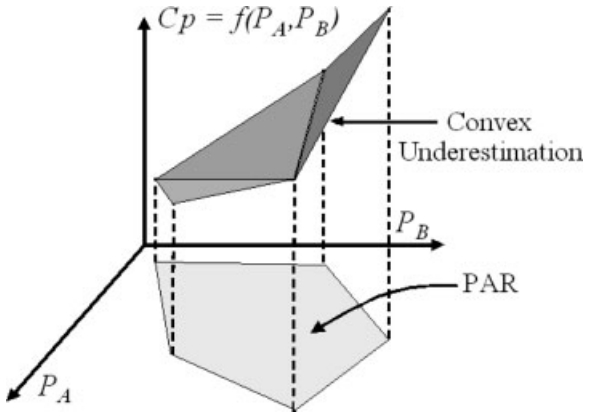

Figure 5. Convex underestimation of Cp as function of P_A and P_B.

$$\mathrm{Cp}_t \geq \sum_{k \in K^{\mathrm{FP}}} w_k^l \, P_{kt} + \gamma^l \quad \forall l \in L_\mathrm{O},$$

which can be rearranged to take on the form shown in Eq. 19:

$$\sum_{k \in K^{\mathrm{FP}}} w_k^l \, P_{kt} + w_C^l \mathrm{Cp}_t \leq \Pi^l \quad \forall l \in L_\mathrm{O} \tag{19}$$

Generally, sets L_F and L_O are not known a priori for a given process network. The goal of this article is a systematic framework for identifying Eqs. 18 and 19, which are extremely compact and free of all scheduling variables not explicitly found in the production planning problem (PP1).

The idea of describing the attainable space of feasible production amounts and total production cost is similar to the attainable region approach[34–36] for describing the space of chemical species concentrations that can be reached by a set of chemical reactions. However, in the case of process networks, attainable production depends not only on (intensive) stoichiometry, thermodynamics, and rates, but also on (extensive) equipment capacity, initial inventory, inventory capacity, utility constraints, changeover times, and other characteristics of the network.

Conceptually, the idea of describing the attainable space is also similar to Benders Decomposition,[37–39] where both seek to replace the subproblem with linear constraints. However, the implementation of what and how linear inequalities are added is significantly different. Benders Decomposition is well-defined for linear subproblems, but remains an open topic for mixed-integer subproblems. Several recent ideas are based on a mixed-integer Farkas lemma,[40] Reformulation-Linearization Technique,[41] logic-based cuts,[42] and minimally relaxed cuts.[43]

Algorithm

This section presents our algorithm for locating the PAR. It is based on the convergence of two n-dimensional convex polytopes[44,45]: an underestimation (UE) and an overestimation (OE) of the convex hull of the true feasible region projected onto P_k ($k = 1, \ldots, K$) and Cp. When developing the

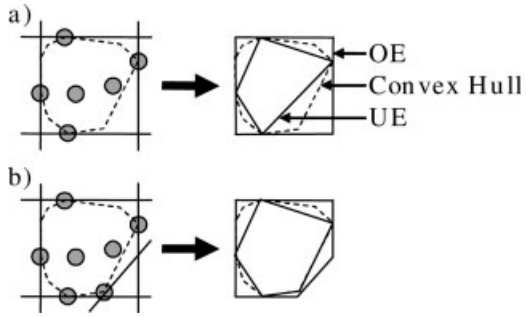

Figure 6. Convex regions from known feasible points (UE) and from found linear inequalities (OE).

inequalities in Eq. 18 (feasibility cuts), we set n equal to K: production of each final product is represented as a separate dimension. When developing the inequalities of Eq. 19 (optimality cuts), we set n equal to $K + 1$: total production cost is an extra dimension. We note that convex polytopes can be expressed by their vertices (V-representation) or by their facets (linear inequalities or half-spaces, H-representation). The UE polytope is the convex hull of a set of feasible solutions that have been found. The OE polytope is the convex region defined by valid linear inequalities. By definition, a valid inequality must be satisfied by all feasible solutions. As seen in Figure 6, the UE (OE) polytope is always less (greater) than or equal to the convex hull of all feasible solutions. In Figure 6b, adding a new feasible solution (linear inequality) monotonically inflates (deflates) UE (OE) polytope.

Our method for identifying new feasible solutions and new linear inequalities involves iteratively solving model (S2), which is the original scheduling model (S1) plus linking Eq. 17 and the following weighted objective function:

$$z = \max \sum_{k \in K^{\mathrm{FP}}} w_k P_k + w_C \mathrm{Cp} \tag{20}$$

where $w_C = 0$ when generating feasibility cuts. The solution of (S2), even if nonoptimal, provides a feasible point for the UE and a valid inequality OE^m for the OE:

$$\mathrm{OE}^m = \sum_{k \in K^{\mathrm{FP}}} w_k^m P_k + w_C \mathrm{Cp} \leq \Pi^m \quad \forall m \tag{21}$$

where Π^m is the best upper bound of (S2). If (S2) is solved to optimality, then the best upper bound is the optimal objective value. The procedure for generating feasibility cuts is shown in Table 1.

The initial $2K$ search directions (i.e. weights) maximize and minimize individual production of each product P_k to obtain $2K$ inequalities that set up the initial OE and $2K$ (not necessarily unique) feasible points that set up the initial UE. This guarantees that the initial OE is bounded. Optimization problem (M1) is then solved to identify a new search direction, as well as determine the maximum perpendicular distance (MPD) from the UE to the OE. We begin iterating: using the new search direction, we solve (S2) to obtain a new inequality for the OE and a new feasible point for the UE. We update the OE by adding the new inequality to the set of found inequalities that define the OE. We update the UE by discarding all previous UE inequalities and then passing all feasible points, both new and previously found, into the Quickhull algorithm[46] for conversion into "updated" UE inequalities. For clarity of presentation, the set of UE inequalities is indexed by the iteration m in which they are updated, e.g., $l \in L_{\mathrm{F}}(m)$. Note that if (S2) is solved to optimality, then the new feasible point satisfies the new OE inequality as equality. We repeat the process until convergence is achieved, convergence cannot improve by our algorithm, or iteration or cumulative resource limit is reached. Finally, schedules found by model (S2) are saved for later use.

The procedure for generating optimality cuts is almost identical. Search direction $\boldsymbol{w}^{\boldsymbol{m}}$ is augmented by w_C^m to become $\boldsymbol{w}^{\boldsymbol{m}} = [w_1^m, w_2^m, \ldots, w_K^m, w_C^m]$. If the algorithm is run twice, once to generate feasibility (Eq. 18) and once to generate optimality (Eq. 19) cuts, then optimality cuts with $w_{\mathrm{C}} \geq 0$ may be discarded because they provide redundant feasibility information for P_k and irrelevant upper bounding information for Cp. Next we discuss in detail the steps of the algorithm.

Solution of model (S2)

If the discrete-time STN scheduling model is used, then the optimization model (S2) consists of scheduling Eqs. 9–17 plus objective function Eq. 20. Note that any scheduling formulation or scheduling-specific algorithm can be used instead. To keep the presentation simple, we assume that the attainable region of a process network is independent of time period, i.e., in this section, we drop index t from variables P_{kt} and Cp_t. In each iteration m, we use search direction $\boldsymbol{w}^{\boldsymbol{m}} = [w_1^m, w_2^m, \ldots, w_K^m, w_C^m]^{\mathrm{T}}$ to maximize the weighted production of final products $k \in K^{\mathrm{FP}}$ and total production cost Cp.

The Quickhull algorithm

This study makes extensive use of the Quickhull algorithm[46] (www.qhull.org) to convert convex polytopes from V-representation to H-representation, to convert from H-representation to V-representation, to calculate polytope volume, and to filter inequalities. At the heart of the Quickhull algorithm, an n-dimensional convex hull is constructed using $(n - 1)$-dimensional facets that are each defined by n points. For $n = 2$, the convex hull is constructed from lines that are each defined by two points. For $n = 3$, the convex hull is

Table 1. Outline of Proposed Algorithm

0	Set m^* to 0; choose initial search directions for $1 \leq m \leq 2K$: $\boldsymbol{w}^{\boldsymbol{m}} = [w_1^m, w_2^m, \ldots, w_K^m]^T$.
1	Increase m^* by 1.
2	Solve (S2) with weight $\boldsymbol{w}^{\boldsymbol{m}^*}$ to obtain a feasible point for the UE and an inequality for the OE.
–	If $m^* < 2K$, then return to 1; otherwise, continue to 3.
3	Run the Quickhull algorithm to convert feasible points into UE inequalities.
4	Solve (M1) to find the MPD and identify new search direction $\boldsymbol{w}^{\boldsymbol{m}^*+\boldsymbol{1}}$.
–	If (MPD > 0) AND ($\boldsymbol{w}^{\boldsymbol{m}} \neq \boldsymbol{w}^{\boldsymbol{m}^*+\boldsymbol{1}}$ $\forall$ $m \leq m^*$) AND ($m^* <$ MAX), return to 1; otherwise, continue to 5.
5	Set PAR inequalities Eq. 18 equal to UE inequalities.
6	Filter to remove redundant OE inequalities.

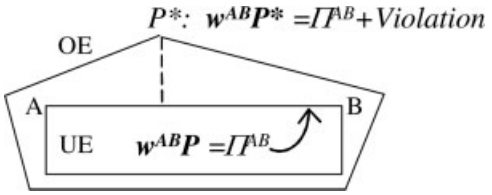

Figure 7. Violation of an UE inequality.

constructed from planes that are each defined by three points. Every facet can be written as a linear inequality of the following form:

$$\sum_{k \in K^{\mathrm{FP}}} w_k P_{kt} \leq \Pi \quad \forall \text{UE facet} \tag{22}$$

where P_k are coordinates, $\boldsymbol{w}$ is the outward-pointing unit normal vector of the facet, and Π is a scalar. For each facet, $\boldsymbol{w}$ and Π can be computed from the coordinates of the n points defining that facet. Different sets of points may lead to the same inequality. Note that the Quickhull algorithm automatically discards interior points. One method for filtering redundant inequalities from the set of OE inequalities is to convert that set into a set of vertices and then back into a set of inequalities.

Solution of model (M1)

Search directions should point toward unexplored regions of the convex hull where the OE and UE do not match. To achieve this, we seek a point $\boldsymbol{P} = [P_1, P_2, \ldots, P_K]^{\mathrm{T}}$ that is within the OE but outside the UE. To be outside the UE polytope, at least one UE inequality must be violated. On the basis of the normal vector $\boldsymbol{w}$ and RHS Π^l of an UE inequality, its violation by $\boldsymbol{P}$ is easy to calculate: Violation = max $(0, \boldsymbol{w}^T\boldsymbol{P} - \Pi^l)$. If w is normalized to length one, then the magnitude of violation is the Euclidean distance from $\boldsymbol{P}$ to that UE inequality.

If the OE and UE polytopes have not converged, then at least one point still in the OE violates at least one UE inequality. Hence, our algorithm chooses the next search direction from among the outward-pointing normal vectors of current UE inequalities. Optimization model (M1) consists of Eqs. 23–28 and is formulated to identify the UE inequality perpendicularly most violated by some point $\boldsymbol{P}$ that is restricted within the OE:

$$\max \mathrm{MPD} = \sum_{l \in L_F(m^*)} \mathrm{slack}^l \tag{23}$$

$$\sum_{k \in K^{\mathrm{FP}}} w_k^m P_k \leq \Pi^m \quad \forall m \leq m^* \tag{24}$$

$$\sum_{l \in L_F(m^*)} Z^l = 1 \tag{25}$$

$$\sum_{k \in K^{\mathrm{FP}}} w_k^l P_k \geq \Pi^l + \mathrm{slack}^l - M(1 - Z^l) \quad \forall l \in L_F(m^*) \tag{26}$$

$$\mathrm{slack}^l \leq MZ^l \quad \forall l \in L_F(m^*) \tag{27}$$

$$\mathrm{MPD} \geq 0 \quad \mathrm{slack}^l \geq 0 \quad \forall l \in L_F(m^*) \quad Z^l \in \{0,1\} \quad \forall l \in L_F(m^*) \tag{28}$$

where ω_k^m (ω_k^l) and Π^m (Π^l) are coefficients and RHS of OE (UE) inequalities, and $L_{\mathrm{F}}(m^*)$ is the set of UE inequalities at iteration m^*. Equation 24 restricts point $\boldsymbol{P}$ within the OE. Because of Eq. 25, exactly one UE inequality is "active" ($Z^l = 1$). For the active inequality, Eq. 26 is satisfied as equality, and slack^l records the violation of P measured in Euclidean distance. Because of Eq. 27, slack^l is allowed to be nonzero only for the active inequality. Model (M1) is set up to maximize the nonzero slack^l, so $\boldsymbol{P}$ is driven to a vertex of the OE and the objective function yields the maximum perpendicular distance (MPD). The next search direction is set perpendicular to the active UE inequality: $\boldsymbol{w}^{m^*+1} = \boldsymbol{w}^{l=active}$. In Figure 7 solving (M1) identifies as active the UE inequality defined by vertices A and B ($Z^{\mathrm{AB}} = 1$). The MPD is measured perpendicularly from this facet to $\boldsymbol{P}^*$. Normal vector $\boldsymbol{w}^{\mathbf{AB}}$ is chosen as the next search direction.

Selecting a search direction from among current UE inequalities has several advantages: (a) UE inequalities are generated from feasible points previously found by the algo-

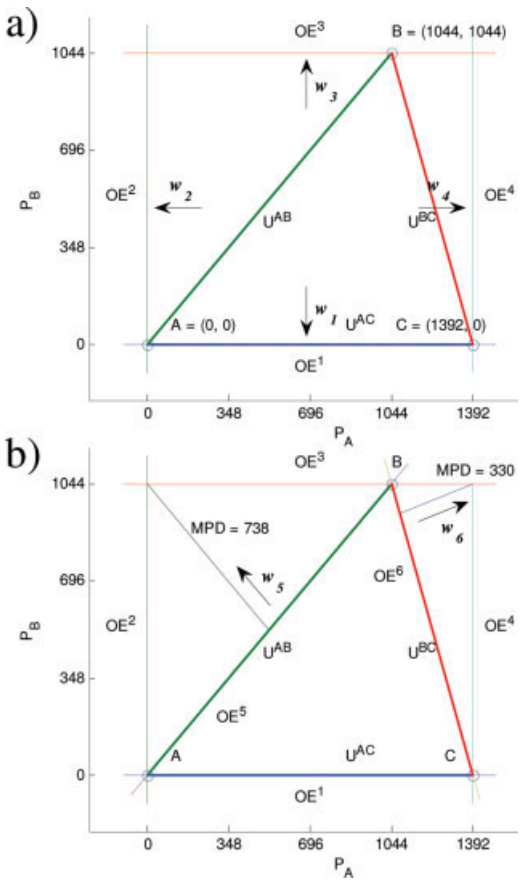

Figure 8. Development of inequalities for the process network of Figure 2.

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

Table 2. Solution of (S2) for Figure 8

	m	Solution	w^m	Π^m	Active UE	MPD	Vol. Ratio (%)
Initial	1	A or C	[0,−1]	0	–	738	50
	2	A	[−1,0]	0	–		
	3	B	[0,1]	1,044	–		
	4	C	[1,0]	1,392	–		
Iterations	5	A or B	$[-1,1]/\sqrt{2}$	0	UE^{AB}	330	80
	6	B or C	$[3,1]/\sqrt{10}$	$4,176/\sqrt{10}$	UE^{BC}	0	100

rithm, (b) UE inequalities in the investigated region are likely to be replaced via the Quickhull algorithm by new tighter UE inequalities, and (c) UE inequalities away from the investigated region are unchanged. In addition, the solution of (M1) can be used to create a redundancy check for the next iteration. The new OE inequality found by (S2) should only be added if it deflates the OE, i.e. if its RHS Π^m is strictly less than Π^{active} + MPD from the previous (M1).

Termination

The algorithm terminates if a resource/iteration limit is reached or the MPD stops improving. MPD is attractive as a possible measure of convergence because it: (a) is already calculated in every iteration, (b) is necessarily positive if the UE and OE do not match, and zero if they do match. However, the MPD does not necessarily decrease monotonically. An alternative convergence criterion that improves monotonically is the ratio Vol(UE)/Vol(OE) of the volumes of the two polytopes. In each iteration m*, solving model (S2) results in one of the following cases:

(a) Model (S2) is solved to optimality.
(b) Model (S2) is not solved to optimality, yet the UE, the OE, or both are improved. For cases (a) and (b), the new search direction $\boldsymbol{w^{m^*+1}}$ is either new (which is desirable) or else a repeat of $\boldsymbol{w^m}$ for some $m \leq m^*$. The latter is possible if model (S2) was previously solved not to optimality (case b) in direction $\boldsymbol{w^m}$.
(c) Model (S2) is not solved to optimality, and neither the UE nor the OE is improved. Neither polytope is improved, so solving model (M1) yields the same active UE inequality as in the previous iteration: $\boldsymbol{w^{m^*+1}} = \boldsymbol{w^{m^*}}$.

The algorithm ceases improving if the new search direction is a direction that has already been searched. Note that the algorithm can be modified to find a different search direction, or model (S2) can be solved using a higher computational resource limit when solving for the second time in a direction.

Motivating example revisited

The above algorithm was applied to the process network shown in Figure 2. The four initial search directions $\boldsymbol{w^{m=1}}$, $\boldsymbol{w^2}$, $\boldsymbol{w^3}$, and $\boldsymbol{w^4}$ are shown as arrows in Figure 8. They yield points A (or C), A, B, and C, respectively, and inequalities OE^1, OE^2, OE^3, and OE^4, respectively. Points A, B, and C are input into the Quickhull algorithm to yield inequalities UE^{AB}, UE^{BC}, and UE^{AC}. The MPD between the two initial polytopes is equal to 738; the new search direction $\boldsymbol{w^5}$ is perpendicular to U^{AB}. In the first iteration ($m = 5$), the solution of (S2) yields point A (or B), which does not inflate the UE. The bound of (S2) generates inequality OE^5, which is identical to UE^{AB}. Next, the MPD is 330 and the new search direction $\boldsymbol{w^6}$ is perpendicular to UE^{BC}. In the second iteration ($m = 6$), the solution of (S2) yields point B (or C) and inequality OE^6, which is identical to UE^{BC}. Again, the UE is not inflated. After adding OE^6, the MPD is 0, indicating convergence of the OE and UE. The algorithm terminates. Note that the coefficients of generated OE inequalities provide a history of previous searched directions. In this example, the initial UE polytope L_F ($m = 4$) is able to find the convex hull so that the UE cannot be inflated during iterations. The solution of scheduling model (S2) in each iteration is given in Table 2. Of the six total OE inequalities generated, three ($m = 2, 3, 4$) can be filtered out. The remaining three inequalities match the three UE inequalities shown in Table 3.

Table 3. UE Inequalities from Feasible Solutions (Points) in Figure 8

$l \in L_F$ ($m = 4$)	Vertices	Inequalities
1	A, C	$-P_B \leq 0$
2	A, B	$-P_A/\sqrt{2} + P_B/\sqrt{2} \leq 0$
3	B, C	$3P_A/\sqrt{10} + P_B/\sqrt{10} \leq 4176/\sqrt{10}$

Multistage example

To demonstrate the flexibility of the proposed method, we generate the PAR of a process network with many products and a multistage structure, such as found in the food and chemical industries. Process network (PN1) (Figure 9) is an example modified from Maravelias[47] with six processing units and seven final products. Detailed data for this example

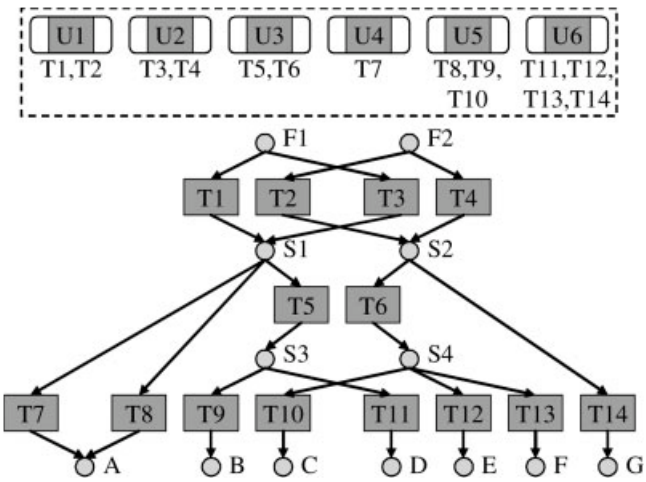

Figure 9. Process network (PN1).

Table 4. Convergence of the Algorithm for (PN1)

	MPD	UE Volume	OE Volume	Vol. Ratio (%)
Initial polytopes	1053.413	9.68E + 14	4.88E + 18	0.02
After 10 iterations	62.505	2.62E + 15	5.22E + 15	50.1
After 20 iterations	55.777	3.15E + 15	5.01E + 15	62.8
After 30 iterations	21.711	3.23E + 15	3.32E + 15	97.1
After 40 iterations	6.081	3.30E + 15	3.30E + 15	99.9

are given in Appendix A. In the first stage, orders (batches) passing through units U1 or U2 take 2 h. In the second stage, orders passing through unit U3 take 1 h. In the third stage, processing time is 2 h for tasks on unit U4 and 1 h for tasks on units U5 and U6. Product A can be produced by task T7 or T8. Products E and F are produced by different tasks but involve the same precursor on the same unit. Note that products A and G bypass the second stage.

For a 48-h scheduling horizon, a seven-dimensional PAR was generated after 40 iterations and 4000 s. This PAR is defined by 67 UE inequalities and bounded by 25 filtered OE inequalities. Furthermore, the UE/OE volume ratio is in excess of 99% as shown in Table 4. In this example, the volume ratio is high because (S2) is solved to optimality in a large number of iterations (31 out of 40). To put the final MPD (6.081) in perspective, the maximum production of any single product over 48 h is between 450 and 575 U. Every vertex of the PAR is a feasible solution/detailed schedule. Any feasible solution that is outside the PAR is guaranteed to be within the OE, therefore within 6.081 U of the PAR.

Extensions

Time-dependent functions

In production planning, it is often required to determine weekly production targets for the first 2–4 periods, while monthly production targets are sufficient for subsequent periods. Hence, we often have to consider nonuniform planning periods of fixed or varying length. To solve such problems we can modify the RHS of Eq. 18 to be time-dependent:

$$\sum_{k \in K^{\mathrm{FP}}} w_k^l P_{kt} \leq \Pi^l(H_t) \quad \forall l \in L_F \tag{29}$$

If $\Pi^l(H_t)$ is a linear or convex piece-wise linear function, then Eq. 29 can be rearranged to resemble Eq. 19, with H_t becoming an extra dimension of the PAR (Figure 10). Time-dependent PARs can be used to solve production planning problems in which the timing and duration of medium-term maintenance needs to be scheduled, or the short-term scheduling decomposes into a start-up/shut-down phase and a cycling phase.

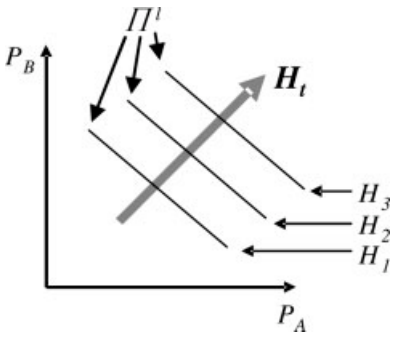

Figure 10. Development of time-dependent PARs: an inequality being inflated by H_t.

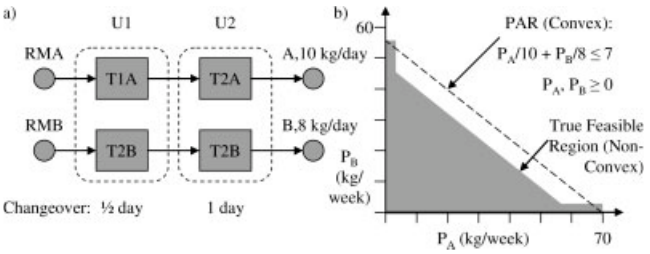

Figure 11. Process attainable region in the presence of changeovers.

Nonconvex regions and functions

The linear constraints in Eqs. 18 and 19 seek to provide a convex approximation of feasible production targets P_{kt} and a convex underestimation of production cost Cp_t, respectively. However, in facilities where the capacity of the process network depends heavily on the loading of equipment units and/or when large changeover times/costs are present, these approximations may be poor. Consider the process network in Figure 11a, in which changeover for units is significant. Units U1 and U2 can be used together to produce product A at rate 10 kg/day or product B at rate 8 kg/day. Changing products requires half-day for unit U1 and 1 day for unit U2. For a scheduling horizon of one week, the tightest convex approximation is shown by the dotted line in Figure 11b, while the true feasible region is shown by the gray nonconvex region.

These nonconvex regions and functions can be modeled as unions of convex regions using disjunctive programming.[48,49] For this process network, the attainable region can be divided into three subregions. In region I, both units begin set up for producing A and neither unit undergoes changeover. In region II, both units begin set up for producing B and neither unit undergoes changeover. In region III, both units begin set up for producing the same product and each undergoes exactly one changeover to produce the other product. Note that changeover of unit U2 is a bottleneck that holds up overall production by 1 day. A binary is introduced for each region: Z^{I}, Z^{II}, and Z^{III}. Production P_k is disaggregated by region: P_k^{I}, P_k^{II}, and P_k^{III}. Certain production amounts (e.g., $[P_{\mathrm{A}}, P_{\mathrm{B}}] = [60,0]$) can be attained by more than one region;

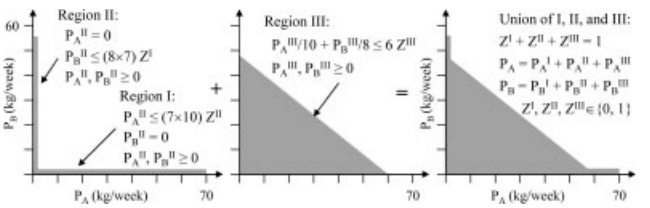

Figure 12. True feasible region as the union of convex regions.

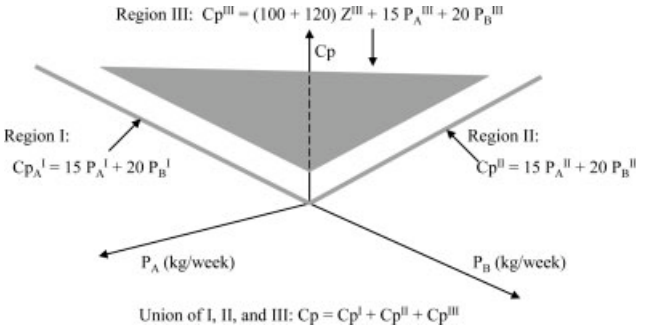

Figure 13. True feasible region including production cost.

Equations from Figure 12 are also needed but are not shown.

nevertheless, the union of all regions shown in Figure 12 correctly models the true feasible region.

Modeling of nonconvex feasible regions can be extended to production cost Cp. Suppose, for the process network shown in Figure 11a, units U1 and U2 have fixed changeover costs 100 and 120, respectively, and operating costs 15 and 20 per kg, respectively. The attainable region of the process network is shown in Figure 13, where Cp is modeled as an extra dimension. Similar to P_k, Cp is disaggregated by region. Equations from Figure 12 are also needed in Figure 13, but are not shown. Note that the projection of all regions in Figure 13 onto the P_A and P_B axes yields the unified region in Figure 12.

In this simple example, we use three auxiliary variables to capture all the necessary information. In general, there are many ways to partition the true nonconvex region into convex regions and many ways to formulate a nonconvex region using discrete variables. Hence, the development of a systematic framework for the development of nonconvex attainable regions is a topic for future research. Regardless of the specifics of the representation, however, the surrogate model is expected to be substantial simpler than the MIP scheduling models currently used in integrated formulations.

Identification of production bottlenecks

The proposed method can be used to identify limiting constraints of a process network. Consider (PN2) in Figure 14, an example modified from Kondili et al.[11] with four process-

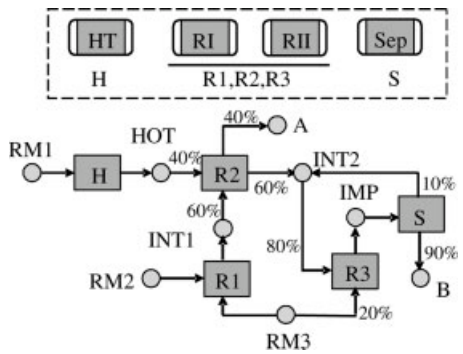

Figure 14. Process network (PN2).

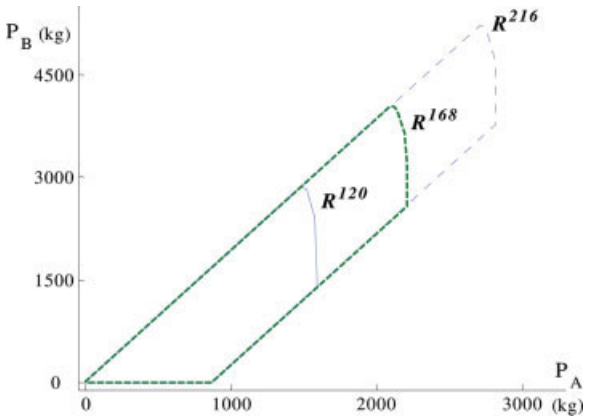

Figure 15. Effect of length of planning period.

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

ing units. Detailed data for this example are given in Appendix A.

We develop the PAR of (PN2) for planning periods (i.e. scheduling horizons) of 120, 168, and 216 h. The corresponding PARs $\mathbf{R}^{120}$, $\mathbf{R}^{168}$, and $\mathbf{R}^{216}$ are shown in Figure 15. As explained previously, the shape of the PAR reflects the characteristics of the process network. In this example, product B requires intermediate INT2, which is a byproduct of A. Hence, production of B requires production of A in ratio 1.9275:1, as reflected in the slope of the upper-left boundary of the PAR of (PN2). This hidden constraint is caused by coproduction, where one product must be produced whenever another product is produced. Coproduction is common in cutting stock problems where different sizes have to be combined to reduce waste, in multistage continuous plants where a process is connected to two or more downstream processes to match the processing rate, and in any process network with significant amounts of byproducts. Coproduction represents a structural constraint that is difficult to identify because it does not explicitly depend on production capacities/rates.

Production of B is further linked to production A via the storage capacities S_{INT2}^{MAX} and S_{IMP}^{MAX} for intermediates INT2

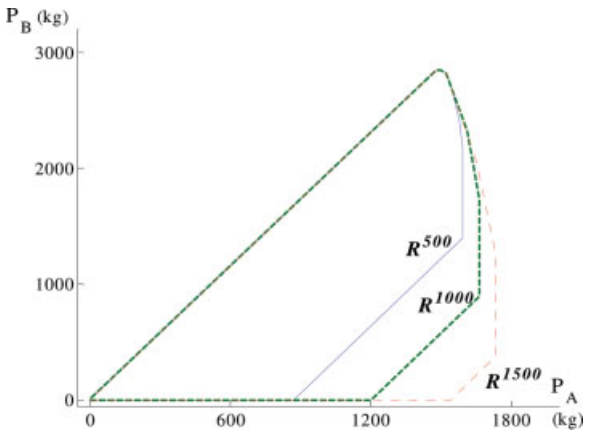

Figure 16. Effect of storage capacity.

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

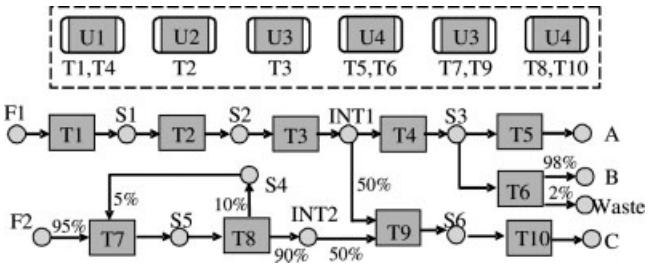

Figure 17. Process network (PN3).

and IMP, respectively. For $P_A > 866.7$ kg, product B must be produced at at least the stoichiometric ratio 1.9275 to prevent INT2 storage from overflowing. This second hidden constraint is a parallel line that intersects the P_A-axis at 866.7.

Even if the length of scheduling horizon is increased (or the processing capacities/rates of tasks are increased), the feasible production region remains limited by these two coproduction constraints. Only the upper-right front of the PAR is affected by horizon length.

To further illustrate how process features can affect the PAR of a process network, we generate the PAR of (PN2) for a planning period of 120 h with the storage capacity for INT2 increased from 500 to 1000 and 1500 kg. The corresponding regions $\mathbf{R}^{500}$, $\mathbf{R}^{1000}$, and $\mathbf{R}^{1500}$ are shown in Figure 16. As explained earlier, storage capacity S_{INT2}^{MAX} allows us to produce A without producing B. Thus, increasing S_{INT2}^{MAX} allows us to produce smaller amounts of B for the same amount of A.

Figures 15 and 16 demonstrate how subtle process features can affect what production levels are attainable. The proposed method, in addition to providing an efficient method for solving production planning problems, can also be used as a qualitative aid for identifying and analyzing the dominant characteristics and constraints of process networks.

Integration of Production Planning and Scheduling

The inequalities in Eqs. 18 and 19 provide approximations of functions $F(P_{kt})$ and $C(P_{kt})$, respectively. They can be generated off-line and only need to be generated once for each manufacturing facility. Once generated, they can be integrated with the production planning model without introducing any variables not already in the general production planning model (PP1). The proposed model (PP3) consists of Eqs. 1, 4–8, and 18–19. This new model is considerably more tractable than integrated planning–scheduling formulations such as (PP2), which uses Eqs. 9–17 to provide production feasibility and cost information that is more cheaply provided by Eqs. 18–19. It also provides a significantly better approximation of feasible production amounts than models with aggregate capacity constraints.

Table 5. Rolling Horizon Algorithm (S-PP3) for Long-Term Scheduling

0	Set θ to 0.
1	Solve (PP3) for planning periods to obtain targets $P_{k\theta+1}$, $P_{k\theta+2}$, ... , P_{kT}.
2	Increase θ by 1.
3	Solve scheduling model (S3) for period $t = \theta$ to meet target $P_{k\theta}$. Fix $P_{k\theta}$ to closest solution found.
–	If $\theta = T$, stop.
–	If the production target is met exactly ($S_{kN} = P_{k\theta}$ for every $k \in K^{FP}$), go to 2; otherwise, go to 1.

Rolling horizon algorithm for scheduling

Detailed schedules may be generated for the first few (or all) planning periods using the rolling horizon algorithm described in Table 5.

Production targets predicted by (PP3) are likely feasible. However, to ensure feasibility, detailed scheduling model (S3) can be solved for each period. Model (S3) consists of Eqs. 9–17 and 30–32. Its objective is the minimization of the one-norm difference between production targets $\overline{P_k}$ and actual production amounts for which a detailed schedule can be found.

$$\min \mathrm{Dev} = \sum_{k \in K^{FP}} (\mathrm{Dev}_k^+ + \mathrm{Dev}_k^-) + \delta(\mathrm{Cp} - \overline{\mathrm{Cp}}) \tag{30}$$

$$P_k - \overline{P_k} = \mathrm{Dev}_k^+ - \mathrm{Dev}_k^- \quad \forall k \in K^{FP} \tag{31}$$

$$\mathrm{Dev}_k^+ \geq 0 \quad \mathrm{Dev}_k^- \geq 0 \quad \forall k \in K^{FP} \tag{32}$$

where δ is a weight meant to reward total production cost Cp if it is less than $\overline{\mathrm{Cp}}$ predicted by (PP3). Parameter δ is small so minimizing $|P_k - \overline{P_k}|$ is emphasized since missing production targets cause unexpected inventory and backlog levels when (PP3) is resolved. If we are only interested in production target feasibility and not production cost, then δ is set to zero in Eq. 30.

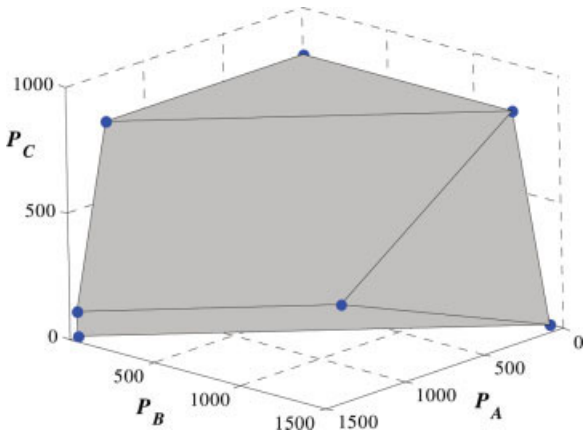

Figure 18. The PAR for (PN3) is defined by 7 feasibility cuts (4 in this view) based on 8 UE vertices (7 in this view, plus the origin).

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

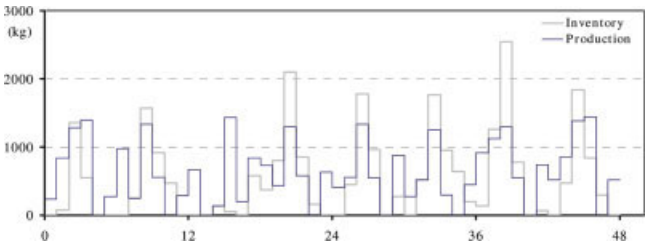

Figure 19. Production amounts and inventory levels for product A.

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

If the scheduling solution in period θ does not meet the predicted production target exactly, then planning model (PP3) is resolved with P_{kt} and Cp_t fixed for $1 \leq t \leq \theta$ and with Eqs. 18 and 19 for $t \in \{\theta + 1, \ldots, T\}$. The updated predictions for $P_{k\theta+1}$ and $Cp_{\theta+1}$ are then used as targets for the next time period, i.e. the planning horizon rolls forward one planning period. This allows for production targets in latter time periods to be updated to "make up for" production targets missed in previous time periods. We refer to the proposed rolling horizon method for scheduling as (S-PP3).

To further enhance the generation of detailed schedules, we can save and reuse schedules found during PAR generation, specifically schedules whose production targets correspond to vertices of the UE (recall PAR $\equiv$ UE). Production targets and total production cost assigned by (PP3) must lie on: (a) a vertex of the PAR, (b) a nonvertex boundary of the PAR, or (c) the interior of the PAR. In the first case, (S3) can be skipped because a saved solution is available. In the second and third cases, the closest saved solution as judged by Eq. 30 can be used to "hot start" (S3).

Furthermore, to solve challenging problems over long planning horizons, multiple PARs can be developed for the same production facility. For example, products can be grouped into families, and family-based PARs of lower dimension can be developed and used with aggregate demand forecasts. Furthermore, PARs for multiple production sites can be developed independently but used simultaneously if the production planning problem involves multiple manufacturing sites.

Example: Production planning and scheduling

To demonstrate rolling horizon method (S-PP3) we consider process network (PN3) in Figure 17, which is a modified example from Papageorgiou and Pantelides[16] with six processing units and three final products. Data are given in appendix A. Algorithm (S-PP3) is used to solve a production planning problem with 48 1-week planning periods. The objective is to meet weekly demands for products A, B, and C at minimum cost. Demand may be fulfilled late, but each tardy period is penalized by a backlog cost that is equal to five times the holding cost. Starting in the second week, demand for all products is random but follows a 6-week periodic pattern (Figure A1). Scheduling within each week is modeled using a scheduling horizon of 168 h.

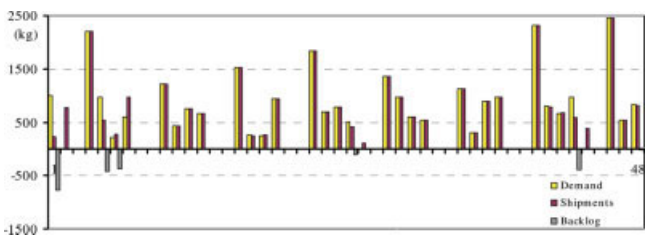

Figure 20. Demand, shipments, and backlog levels for product A.

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

Table 6. Results of Model (PP3) and Algorithm (S-PP3), with No Production Cost

	(PP3)	(S-PP3)	(PP2)
Holding cost ($)	578,074	576,914	587,368
Backlog cost ($)	157,158	164,520	154,466
Total cost ($)	735,232	741,434	741,835
CPU seconds	0.015	1,423.05	7,251.69

Two variations of this problem are solved, with and without production cost at the scheduling level as formulated by Eq. 15. All models were implemented in GAMS 22.1 and solved with CPLEX 10.0 on a Pentium 4 at 2.8 GHz running Windows XP, and PAR generation data are given in Appendix B.

Example: Solution with no production cost

First, the PAR of (PN3) is generated off-line in 4052 CPU seconds to create seven feasibility cuts (Figure 18). Next, model (PP3) is solved to optimality to provide a 48-week production plan with objective $735,232. The production amounts and inventory levels of product A are shown in Figure 19, while the demand, shipments, and backlog levels for A are shown in Figure 20. Note that production amounts and inventory levels both peak just before a large order is due, and this is because some order demands exceed the production capacity, and it is cheaper to build-up inventory than to work down backlog. Note also that shipments follow demand levels very closely and that backlog is usually satisfied within 1 week. Similar trends are observed for the other two products.

The solution of model (PP3) provides a complete production plan, i.e. production targets, inventory levels and shipments. Furthermore, we can use the rolling horizon algorithm (S-PP3) to generate detailed schedules. The final results of both are summarized in Table 6. The final solution of

Table 7. Computational Statistics of Algorithm (S-PP3), 48 Planning Time Periods

Model	Times	Total CPU
(PP3), using OE inequalities (Bound)	1	0.150
(PP3), using UE inequalities (PAR)	1	0.150
Production targets at PAR vertices	3	–
Solution of scheduling model (S3)	45	–
Periods where targets exactly met	37	462.111
Periods where targets inexactly met	8	960.761
Solution of (PP3) to update targets	8	0.152
Total	–	1,423.054

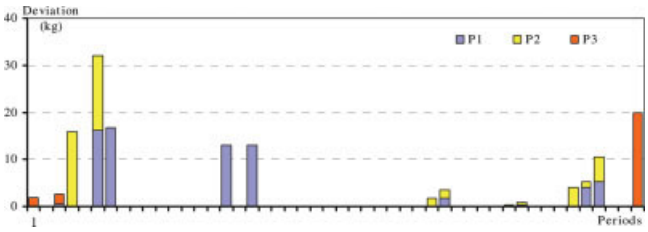

Figure 21. Absolute deviation of production amounts between (PP3) and (S-PP3).

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

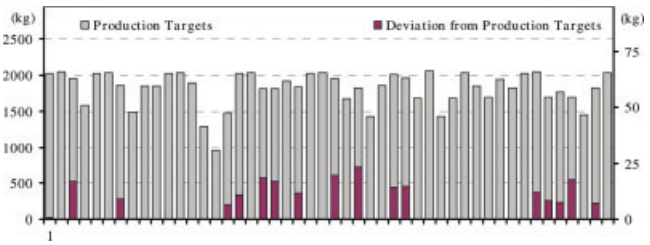

Figure 22. Production targets and deviation of production amounts of Product A.

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

(S-PP3) is \$741,434, which is within 1% of (PP3), \$735,232. A lower bound of \$686,777, within 7%, is found in 0.015 CPU seconds by solving (PP3) with UE inequalities replaced by OE inequalities, since the latter provide a strict overapproximation of the exact attainable region. Hence, the solution provided by (PP3) is provably tight and can be used to construct detailed schedules with almost no degeneration. By comparison, solution of the fullspace model (PP2) after 2 h yields a feasible solution with objective equal to \$741,835 with a lower bound of \$688,810.

Computational statistics for algorithm (S-PP3) are given in Table 7. Of the 48 planning time periods, three time periods had production targets coinciding with a saved solution from PAR generation; (S3) does not need to be run, because detailed schedules are available. For 37 time periods, (S3) is able to identify a detailed schedule for exactly fulfilling the production target (462 s total). For the remaining eight time periods, an exact match could not be found in 120 CPU s (961 s total). For these eight time periods, (PP3) was rerun (0.152 s total) to update downstream production targets.

The average total ($P_A + P_B + P_C$) production is ~1820 kg per planning time period. The average absolute deviation (shown in Figure 21) between (PP3) and production amounts found by (S3) is less than 3.0 kg or 0.17%. The average absolute deviation between updated production targets and production amounts found by (S3) is 1.3 kg or 0.07%. This demonstrates our method's ability to effectively locate production targets that are provably feasible. Even though a minority of planning time periods have nonzero deviation, the amount of that deviation is remarkably small. Finally, note that detailed schedules for all 48 periods are obtained in less than 30 min.

Example: Solution with production cost

Here, we solve a variation of the above problem in which there is a fixed and variable production cost for each scheduling batch. Equation 19 for (PN3) is generated off-line in 14,915 CPU s to create 79 optimality cuts (Eq. 19). The production planning problem is first solved using model (PP3). Using algorithm (S-PP3) to ensure feasibility, the objective is degraded from \$1,167,094 to \$1,190,475. A lower bound of \$1,086,963, within 7%, is found in 0.046 CPU s solving (PP3) with UE inequalities replaced by OE inequalities. Results are summarized in Table 8. Note that model (PP3) provides a solution that is within 2% of the detailed solution obtained by (S-PP3), which is again very close. Solution of the (PP2) after 2 h yields a feasible solution of \$1,184,543 with a best bound of \$1,092,743.

Figure 22 shows updated production targets predicted by iterative solution of (PP3) and the absolute deviation of the production amounts found by (S3). Note that the sum of production targets for all three products in each period is between 960 and 2060 kg (left *y*-axis), while the total deviation lies between 0 and 23 kg (right *y*-axis). The average deviation is 4.5 kg, 0.25% of the production. Similar to before, the deviation between production targets predicted by our method and what is provably feasible is very small.

Figure 23 shows updated production cost predictions and the positive deviation of total production cost found by (S3). Production cost predictions are between \$5400 and \$10,800 (*y*-axis on the left), while positive deviations are between 0 and \$207.4 (*y*-axis on the right).

These two examples demonstrate the ability of our method to compactly yet effectively convey what production amounts P_k and total production cost Cp are feasible. Our method (PP3) can very quickly locate a near-feasible solution of high quality. Additionally, to ensure feasibility, the proposed rolling horizon algorithm (S-PP3) may be used to translate the

Table 8. Results of Model (PP3) and Algorithm (S-PP3) with Production Cost

	(PP3)	(S-PP3)	(PP2)
Holding cost (\$)	574,756	568,345	580,180
Backlog cost (\$)	183,933	210,221	191,641
Production cost (\$)	408,405	411,910	412,722
Total cost (\$)	1,167,094	1,190,475	1,184,543
CPU seconds	0.062	5,406.09	7,282.98

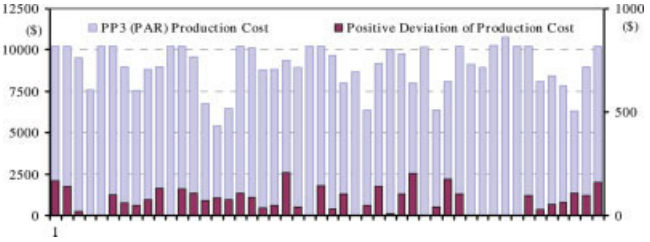

Figure 23. Predicted production cost and positive deviation from prediction.

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

(PP3) solution into detailed schedules, with almost no deviation in production amounts and production cost.

Conclusions

In this article, we presented a framework that enables us to effectively (a) determine the set of attainable production levels (targets) in a multiproduct process network and (b) calculate production cost as a function of production levels. The framework involves the development of two sets of constraints that provide a convex approximation of the set of feasible production amounts and an underestimating function of the production cost. They include only planning variables P_{kt} and can thus be efficiently integrated with production-planning formulations. The method is general because it can be used with any scheduling formulation to develop the attainable region of any type of manufacturing facility. In this article, we used STN-based subproblems (models (S2) & (S3)), but any MIP model that describes scheduling decisions in a given facility could be used instead. In principle, subproblems do not even need to be formulated as MIPs. Therefore, the method can be used for the solution of problems in all different types of process networks (e.g., multistage, multipurpose, etc.).

We also showed that the proposed framework can be extended to handle nonconvex feasible regions and cost functions via the introduction of binary variables. In addition, construction of approximating functions can be used to identify and analyze the dominant characteristics of a process network. The proposed framework can also be used for developing time-dependent approximating functions, which should be particularly useful for applications involving long-term maintenance or cyclic scheduling.

Finally, it was shown that the production planning formulation can be integrated with scheduling models in a rolling horizon approach to provide detailed schedules for every planning period, with almost no deviation in predicted production amounts and only a small deterioration of the production-planning objective function.

Notation

Summary of Optimization Models/Algorithms

PP1 = general production planning (Eqs. 1–8)
PP2 = integrated planning-scheduling: (PP1) + (S1) + (17) (Eqs. 1, 4–8, 9–17)
PP3 = proposed production planning formulation: (PP1) + PAR (Eqs. 1, 4–8, 18, 19)
S1 = scheduling: used for integration with (PP1) (Eqs. 9–16).
S2 = scheduling: used to obtain a feasible point for the UE and an inequality for the OE (Eqs. 9–17, 20)
S3 = scheduling: used to obtain detailed schedules in (S-PP3) (Eqs. 9–17, 30–32)
M1 = used to determine MPD and a new direction for (S2) (Eqs. 23–28)
S-PP3 = rolling horizon algorithm for detailed scheduling (Table 5)

Indices/sets

$i \in I$ = processing task
$j \in J$ = equipment unit
$k \in K$ = chemical (product, state)
$l \in L_F/L_O$ = feasibility/optimality cuts; used to index PARs
m = iteration in the proposed algorithm; index of the inequalities in Eqs. 21
$n \in \{0, 1, \ldots, N\}$ = scheduling (small bucket) time point/period
$t \in \{0, 1, \ldots, T\}$ = planning (big bucket) time point/period

Subsets

I^B/I^C = set of batch/continuous tasks
$I(j)$ = set of tasks that can be carried out in unit j
$I(k)/O(k)$ = set of tasks consuming/producing chemical k
$K^{RM}/K^{INT}/K^{FP}$ = set of raw materials/intermediates/final products

Parameters

Dem_{kt} = demand for final product $k \in K^{FP}$ at time point t
h_k = unit holding cost for product k
H = length of planning horizon
H_t = length of planning period t
M = large number in big-M constraints
r_i^{MIN}/r_j^{MAX} = minimum/maximum processing rate of continuous task $i \in I^C$
S_k^{MAX} = storage capacity for chemical k
u_k = unit backlog cost for final product $k \in K^{FP}$
$V_{ij}^{MIN}/V_{ij}^{MAX}$ = minimum/maximum batch size of batch task $i \in I^B$ on unit $j \in J$
w_k^l = coefficient for product $k \in K^{FP}$ in inequality $l \in L_F \cup L_O$
w_C^l = coefficient for production cost in inequality $l \in L_O$
α_i = fixed cost for beginning batch task $i \in I^B$ or processing continuous task $i \in I^C$
β_i^B = variable cost for processing batch task $i \in I^B$
β_i^C = variable cost for processing continuous task $i \in I^C$
δ = coefficient in Eq. 30
Δt = length of small-bucket (scheduling) period
π_k = price of final product $k \in K^{FP}$ if sales are allowed
Π^l/Π^m = RHS of UE/OE inequality
ρ_{ik} = stoichiometric coefficient of chemical k in task i
τ_i = processing time of batch task $i \in I^B$

Planning variables

Ch_t = holding cost in period t
Cu_t = backlog cost in period t
Cp_t = production cost in period t
CT = total cost
D_{kt} = shipment of final product $k \in K^{FP}$ in period t
I_{kt} = inventory level of product k in period t
P_{kt} = production target for final product $k \in K^{FP}$ in period t
U_{kt} = outstanding (backlogged) demand of final product $k \in K^{FP}$ in period t

Scheduling variables

W_{ijn} (binary) = 1 if batch (continuous) task $i \in I^B(I^C)$ starts (is being processed) at point n
B_{ijn}^B = batch size of batch task $i \in I^B$ that starts at time point n
B_{ijn}^C = amount of continuous task $i \in I^C$ being processed during period n
S_{kn} = inventory level of chemical k at time point n

(M1) variables

MPD = maximum perpendicular distance
slack^l = if Z^l=1 and the 2-norm of $\boldsymbol{w}^l$ is 1, then slack^l is distance by which P_k violates UE^l
Z^l (binary) = indicates active UE inequality ($Z^l = 1$)

Acknowledgments

The authors acknowledge financial support from the Graduate School and the College of Engineering of the University of Wisconsin, Madison.

Literature Cited

1. U.S. Department of Commerce. Bureau of Economic Analysis. Gross domestic product by industry, 2003–2005 (revised). U.S. Government Printing Office: Washington, DC: December 11, 2006.
2. U.S. Patent and Trademark Office. Technology assessment and forecast report. Office of Electronic Information Products Division – Patent Technology Monitoring Branch: July 2005. The Office of Electronic Information Products is based in Alexandria, VA, but publishes online.
3. Stephanopoulos G. Invention and innovation in a product-centered chemical industry (Webcast lecture). Computing and Systems Technology (CAST), 2004. Available at www.castdiv.org/WebCAST.htm.
4. Chopey NP. Outlook for 2006: challenges plus changes. *Chem Eng.* 2006;1:22–23.
5. Grossmann IE, Westerberg AW. Research challenges in process systems engineering. *AIChE J.* 2000;46:1700–1703.
6. Grossmann IE. Enterprise-wide optimization: a new frontier in process systems engineering. *AIChE J.* 2005;51:1846–1857.
7. Tayur S. What is missing to enable optimization of inventory deployment and supply chain? Plenary Presentation in FOCAPO 2003, Coral Springs, FL, January 12–15, 2003.
8. Shapiro JF. Challenges of strategic supply chain planning and modeling. *Comput Chem Eng.* 2004;28:855–861.
9. Meyr H, Wagner M, Rohde J. Structure of advanced planning systems. In: Stadtler H, Kilger C (eds.), Supply Chain Management and Advanced Planning – Concepts, Models, Software, and Case Studies. Berlin: 99–104, 2002.
10. Shobrys DE, White DC. Planning, scheduling, and control: why cannot they work together. *Comput Chem Eng.* 2002;26:149–160.
11. Kondili E, Pantelides CC, Sargent RWH. A general algorithm for short-term scheduling of batch operations. I. MILP formulation. *Comput Chem Eng.* 1993;17:211–227.
12. Subrahmanyam S, Pekny JF, Reklaitis GV. Decomposition approaches to batch plant design and planning. *Ind Eng Chem Res.* 1996;35:1866–1876.
13. Graves SC. Using Lagrangean techniques to solve hierarchal production planning problems. *Manage Sci.* 1982;28:260–275.
14. Birewar DB, Grossmann IE. Simultaneous production planning and scheduling in multiproduct batch plants. *Ind Eng Chem Res.* 1990;29:570–580.
15. Wellons MC, Reklaitis GV. Scheduling of multipurpose batch chemical plants. II. Multiple-product campaign formation and production planning. *Ind Chem Eng Res.* 1991;30:688–705.
16. Papageorgiou LG, Pantelides CC. Optimal campaign planning/scheduling of multipurpose batch/semicontinuous plants. II. A mathematical decomposition approach. *Ind Chem Eng.* 1996;35:510–529.
17. Karimi IA, McDonald CM. Planning and scheduling of parallel semicontinuous processes. II. Short-term scheduling. *Ind Eng Chem Res.* 1997;36:2701–2714.
18. Zhu XX, Majozi T. Novel continuous time MILP formulation for multipurpose batch plants. II. Integrated planning and scheduling. *Ind Chem Eng Res.* 2001;40:5621–5634.
19. Romero J, Badell M, Bagajewicz M, Puigjaner L. Integrating budgeting models into scheduling and planning models for the chemical batch industry. *Ind Eng Chem Res.* 2003;42:6125–6134.
20. Guillén G, Badell M, Espuña A, Puigjaner L. Simultaneous optimization of process operations and financial decisions to enhance the integrated planning/scheduling of chemical supply chains. *Comput Chem Eng.* 2006;30:421–436.
21. Erdink-Dogan ME, Grossmann IE. A decomposition method for the simultaneous planning and scheduling of single-stage continuous multiproduct plants. *Ind Eng Chem Res.* 2006;45:299–315.
22. Wilkinson SJ, Shah N, Pantelides CC. Aggregate modeling of multipurpose plant operation. *Comput Chem Eng.* 1995;19:S583–S588.
23. Dimitriadis AD, Shah N, Pantelides CC. RTN-based rolling horizon algorithms for medium term scheduling of multipurpose plants. *Comput Chem Eng.* 1997;21:S1061–S1066.
24. Wan X, Pekny JF, Reklaitis GV. Simulation-based optimization with surrogate models—application to supply chain management. *Comput Chem Eng.* 2005;29:1317–1328.
25. McKay KN, Wiers VCS. Integrated decision support for planning, scheduling, and dispatch in a focused factory. *Comput Ind.* 2003;50:5–14.
26. Lasschuit W, Thijssen N. Supporting supply chain and planning decisions in the oil and chemical industry. *Comput Chem Eng.* 2004;28:863–870.
27. Berning G, Brandenburg M, Gursov K, Kussi JS, Mehta V, Tolle FJ. Integrating collaborative planning and supply chain optimization for the chemical process industry (I)—methodology. *Comput Chem Eng.* 2004;28:913–927.
28. Kallrath J. Combined strategic and operational planning: an MILP success story in industry. *OR Spectr.* 2002;24:315–341.
29. Iyer RR, Grossmann IE. A bilevel decomposition algorithm for long-range planning and process networks. *Ind Eng Chem Res.* 1998;37:474–481.
30. Shah N. Process industry supply chains: advances and challenges. *Comput Chem Eng.* 2005;29:1225–1235.
31. Kallrath J. Planning and scheduling in the process industry. *OR Spectr.* 2002;24:219–250.
32. Pinto JM, Joly M, Moro LFL. Planning and scheduling models for refinery operations. *Comput Chem Eng.* 2000;24:2259–2276.
33. Bodington CE. *Planning, Scheduling and Control: Integration in the Process Industries*. New York: McGraw-Hill, 1995.
34. Horn F. *Third European Symposium on Chemical Reaction Engineering*. London: Pergamon Press, 1964.
35. Glassier D, Hildebrandt D, Crowe C. A geometric approach to steady flow reactors: the attainable region and optimization in concentration space. *Ind Eng Chem Res.* 1987;26:1803–1810.
36. Hildebrandt D, Glassier D. The attainable region and optimal reaction structures. *Comput Chem Eng.* 1990;45:2161–2168.
37. Nemhauser GL, Wolsey LA. *Integer and Combinatorial Optimization.* New York: Willey, 1989.
38. Geoffrion AM. Generalized Benders decomposition. *J Optim Theor Appl.* 1972;10:237–260.
39. Benders JF. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik.* 1962;4:238–252.
40. Köppe M, Weismantel R. Cutting planes from a mixed-integer Farkas lemma. *OR Lett.* 2004;32:207–211.
41. Sherali HD, Fraticelli, BMP. A modification of Benders' decomposition algorithm for discrete subproblems: an approach for stochastic problems with integer recourse. *J Global Optim.* 2002;22319–342.
42. Hooker JN, Ottosson G. Logic-based Bender's decomposition. *Math Prog.* 2003;96:33–60.
43. Chu Y, Xia Q. Generating Benders cuts for a general class of integer programming problems. *Lect Notes Computr Sci.* 2004;3011:127–141.
44. Director SW, Hatchel GD. The simplicial approximation approach to design centering. *IEEE Trans Circuits Syst.* 1997;24:363–372.
45. Goyal V, Ierapetritou MG. Determination of operability limits using simplicial approximation. *AIChE J.* 2002;48:2902–2909.
46. Barber CB, Dobkin DP, Huhdanpaa H. The Quickhull algorithm for convex hulls. ACM Trans. *Math Softw.* 1996;22:469–483.
47. Maravelias CT. Mixed-time representation for state-task network models. *Ind Eng Chem Res.* 2005;44:9129–9145.
48. Türkay M, Grossmann IE. Disjunctive programming techniques for the optimization of process systems with discontinuous investment costs-multiple size regions. *Ind Eng Chem Res.* 1996;35:2611–2623.
49. Balas E. Disjunctive programming and hierarchy of relaxations for discrete optimization problems. SIAM Alg. *Discrete Math.* 1985;35:466–486.

Appendix A: Example Data

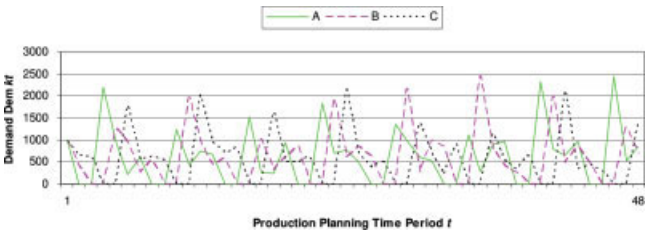

Figure A1. Demand Dem_{kt} for the production planning and scheduling example.

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

Table A1. Data for Figure 2

		RM	INT	A	B
	$S_k^{INITIAL}$	∞	0	0	0
	S_k^{MAX}	∞	0	∞	∞
ρ_{ik}^* (kg)	TA1	−1	1		
	TA2		−1	1	
	TA3		−1		1

*Task i belongs to $O(k)/I(k)$ if ρ_{ik} is positive/negative.

Table A2. Data for Figure 2

	$r_{ij}^{MAX}/r_{ij}^{MIN}$ (kg/wk)		
	U1	U2	U3
TA1	2100/1400		
TA2		1400/1050	
TA3			1050/1050

Table A3. Data for (PN1) in Figure 9

		F1	F2	S1	S2	S3	S4	A	B	C	D	E	F	G
	$S_k^{INITIAL}$	∞	∞	0	0	0	0	0	0	0	0	0	0	0
	S_k^{MAX}	∞	∞	30	30	30	30	∞	∞	∞	∞	∞	∞	∞
ρ_{ik}^* (kg)	T1	−1		1										
	T2		−1		1									
	T3	−1		1										
	T4		−1		1									
	T5			−1		1								
	T6				−1		1							
	T7			−1				1						
	T*			−1				1						
	T9					−1			1					
	T10						−1			1				
	T11					−1					1			
	T12						−1					1		
	T13						−1						1	
	T14				−1									1

*Task i belongs to $O(k)/I(k)$ if ρ_{ik} is positive/negative.

Table A4. Data for (PN1) in Figure 9

	Time (h)	$V_{ij}^{MAX}/V_{ij}^{MIN}$ (kg/h)					
	τ_i	U1	U2	U3	U4	U5	U6
T1	2	10/0					
T2	2	10/0					
T3	2		15/0				
T4	2		15/0				
T5	1			10/0			
T6	1			10/0			
T7	2				15/0		
T*	1					10/0	
T9	1					10/0	
T10	1					10/0	
T11	1					10/0	
T12	1						10/0
T13	1						10/0
T14	1						10/0

Table A5. Data for (PN2) in Figure 14

		RM1	RM2	RM3	HOT	INT1	INT2	IMP	A	B
	S_k^{INITIAL}	∞	∞	∞	0	0	0	0	0	0
	S_k^{MAX}	∞	∞	∞	1000	500	0	1000	∞	∞
$\rho_{ik}{}^*$ (kg)	H	−1			1					
	R1		−0.5	−0.5			1			
	R2				−0.4	0.6	−0.6		0.4	
	R3		−0.2			−0.8		1		
	S					0.1		−1		0.9

*Task i belongs to $O(k)/I(k)$ if ρ_{ik} is positive/negative.

Table A6. Data for (PN2) in Figure 14

	Time (h)	Production Cost		$V_{ij}^{\text{MAX}}/V_{ij}^{\text{MIN}}$ (kg/h)			
	τ_i	α_i (\$)	β_i^{B} (\$/kg)	HT	RI	RII	Sep
H	1	11	0.55	50/0			
R1	2	12	0.60		40/0	25/0	
R2	2	13	0.65		40/0	25/0	
R3	1	14	0.70		40/0	25/0	
S	2	15	0.75				100/0

Table A7. Data for (PN3) in Figure 17

		F1	F2	S1	S2	S3	S4	S5	S6	INT1	INT2	A	B	WS	C
	S_k^{INITIAL}	∞	∞	0	0	0	50	0	0	0	0	0	0	0	0
	S_k^{MAX}	∞	∞	100	100	100	100	100	100	100	100	∞	∞	∞	∞
$\rho_{ik}{}^*$ (kg)	T1	−1		1											
	T2			−1	1										
	T3				−1					1					
	T4					1				−1					
	T5					−1						1			
	T6					−1							0.98	0.02	
	T7		−0.95				−0.05	1							
	T8						0.1	−1			0.9				
	T9								1	−0.5	−0.5				
	T10								−1						1

*Task i belongs to $O(k)/I(k)$ if ρ_{ik} is positive/negative.

Table A8. Data for (PN3) in Figure 17

	Time (h)	Production Cost		$V_{ij}^{\text{MAX}}/V_{ij}^{\text{MIN}}$ (kg/h)					
	τ_i	α_i (\$)	β_i^{B} (\$/kg)	U1	U2	U3	U4	U5	U6
T1	2	11	0.55	50/12.5					
T2	1	12	0.60		80/20				
T3	1	13	0.65			60/15			
T4	2	14	0.70	50/12.5					
T5	2	15	0.75				80/20		
T6	2	16	0.80				80/20		
T7	4	17	0.85					30/7.5	
T8	2	18	0.90						40/10
T9	2	19	0.95					30/7.5	
T10	3	20	1.00						40/10

Table A9. Data for the Production Planning and Scheduling Example

	A	B	C
Initial inventory (kg)	0	0	0
Inventory cost h_k (\$/kg wk)	13.3	16.7	20
Backlog cost u_k (\$/kg wk)	67.7	83.3	100

Appendix B: Model and Computational Statistics of Examples

Table B1. Computational Statistics for PAR Generation

	Figure 2		(PN1)	(PN2)					(PN3)	
Horizon (h)	168	168	48	120	168	216	120	120	120	120
Notes		LP rela.					$S_{INT2}^{MAX} * 2$	$S_{INT2}^{MAX} * 3$		+Cost
Initial directions	4	4	14	4	4	4	4	4	6	8
CPU (S2)	0.31	0.31	2.947	902.15	900.59	902.22	902.94	900.41	1351.20	1801.90
Initial MPD	738	738	1053	1317	1865	2407	1317	1316	1295	2083
UE volume	726,276	726,276	9.68E + 014	1,217,959	1,663,102	3,018,658	1,681,619	2,143,027	2.78E + 08	7.36E + 08
OE volume	1,452,552	1,452,552	4.88E + 018	4,536,844	8,896,366	15,072,445	4,737,214	4,933,068	1.75E + 09	1.75E + 09
Initial vol. ratio (%)	50	50	0.02	27	19	20	35	43	16	42
Iterations	2	2	40	9	7	7	8	7	6	29
CPU (S2)	0.20	0.20	4,065.65	3154.25	3151.06	2251.48	3152.16	2805.82	2,700.53	13,110.83
CPU (M1)	0.02	0.02	16.64	0.15	0.15	0.14	1.65	0.15	0.14	2.03
Final MPD	0	0	6	27	85	177	51	46	18	78
UE volume	726,276	1,392,029	3.30E + 015	1,875,746	2,839,125	4,960,681	2,292,840	2,573,263	7.36E + 08	4.20E + 12
OE volume	726,276	1,392,029	3.30E + 015	1,914,608	2,980,057	5,211,157	2,368,453	2,652,842	7.59E + 08	4.63E + 12
Final vol. ratio (%)	100	100	>99	98	95	95	97	97	97	91
Unique UE vertices	3	5	41	9	7	7	9	8	8	30
Nonredundant OE inequalities	3	5	25	12	11	9	10	10	6	35
Nonredundant UE inequalities	3	5	67	9	7	7	9	8	7	79

Runs for model (S2) that were not solved to optimality were terminated after 450 CPU seconds. The computation time of the Quickhull algorithm is negligible. For two-dimensional polytopes (Figure 2, (PN2)), the number of UE vertices and UE inequalities should be equal.

Manuscript received July 20, 2006, and revision received Feb. 17, 2007.