AIChE

General Framework and Modeling Approach Classification for Chemical Production Scheduling

Christos T. Maravelias
Dept. of Chemical and Biological Engineering, University of Wisconsin, Madison,
1415 Engineering Dr., Madison, WI 53706

DOI 10.1002/aic.13801
Published online April 12, 2012 in Wiley Online Library (wileyonlinelibrary.com).

Despite the increased number of publications in the area during the last two decades, there is no unified notation and systematic framework for chemical production scheduling. In this article, we first develop a framework for the description of scheduling problems in the chemical industries. While building upon ideas used in discrete manufacturing, the proposed framework accounts for features such as material handling restrictions which are critical in chemical production scheduling. Second, we present a classification of the various modeling approaches that have been presented in the process systems engineering literature. Our classification is broader than previous schemes because it accounts for more attributes, and it also offers a broader discussion of the modeling of time. We believe that our analysis will enhance the understanding of chemical production scheduling and lead to further advances in the area. © 2012 American Institute of Chemical Engineers *AIChE J,* 58: 1812–1828, 2012
Keywords: chemical production scheduling, production environment, mixed-integer, programming

Introduction

Scheduling finds its application in a variety of fields: shipping, railways, manufacturing, government organizations, educational institutions, sports, and so forth. In the manufacturing sector, scheduling has been practiced since the early twentieth century, but the first systematic scheduling methods were published in the early 1950s.[1,2] Consequently, a significant body of theoretical results (e.g., regarding complexity hierarchy) as well as a broadly accepted notation and a general problem classification have been developed for discrete manufacturing scheduling.[3,4]

In the process industries, scheduling appears in a wide range of applications: from the oil industry (e.g., transportation and unloading of crude oil from pipelines) to the pharmaceutical and specialty chemical sectors (e.g., batch scheduling for optimal utilization of shared multiproduct facilities). Most of the early scheduling papers in the process systems engineering (PSE) literature, which appeared in the late 1970s and early 1980s, considered batch problems somewhat similar to discrete manufacturing ones. Specifically, they considered problems where batches have to be processed in consecutive stages (with no intermediate mixing or splitting) in the same way discrete jobs require a number of processing steps or operations. The scope of chemical production scheduling was expanded in the early 1990s to include process facilities of arbitrary structure with no material handling (i.e., mixing and splitting) restrictions, additional resource constraints and various processing restrictions. A number of modeling and solution methods have been proposed since then to address an ever-widening set of problems and applications. As a result, chemical production scheduling has become an important research subarea of process operations.[5–9]

Despite the large volume of papers in the field, however, there is no unified notation and framework for chemical production scheduling. This is due to some ambiguity in the terms and concepts used to describe scheduling approaches, as well as the lack of a general problem and model classification methodology. To give just two examples of the former: the term *multipurpose* has been used to describe in some cases general facilities where batch mixing and splitting is allowed, and in some other cases facilities with product-specific sequence of operations and no batch mixing/splitting. Also, the terms *order* and *batch* are often used interchangeably in the context of multistage facilities although their meaning is different.

Accordingly, the goal of this article is twofold. First, it presents a framework that attempts to remove some of the ambiguity in the literature. Second, it presents a general classification of modeling approaches that considers multiple modeling attributes simultaneously, rather than hierarchically.

Our discussion focuses on processes converting a fixed set of input materials into a fixed set of output materials of constant quality. In other words, we assume that the conversion coefficients of all tasks are fixed. Thus, we do not consider blending problems. Also, we do not consider problems where product quality changes dynamically (e.g., polymer grades). Furthermore, the emphasis in the modeling approach classification is on mixed-integer programming (MIP) models. Nevertheless, the main insights of this article are applicable to all modeling/solution paradigms.

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

The article is structured as follows. In the next section, we review some fundamental concepts, place scheduling in the context of supply chain management, present a high-level problem statement, and discuss the major optimization decisions. In "Chemical Production Environments," we develop our framework, where special emphasis is given to the production environment. In "Modeling Approach Classification," we present a general model classification schema and briefly comment on the relationship between problem classes and modeling approaches.

Background

We start with a brief review of scheduling in the PSE literature. Next, we review some fundamental concepts and present the traditional notation used in discrete manufacturing and services scheduling. This is important because, as we will see, the notation for chemical production scheduling is substantially different even though the problems are in some cases very similar. Our discussion borrows elements from Pinedo,[10] but offers a broader perspective as well as some critical insights for our discussion in subsequent sections. Furthermore, we discuss scheduling in the context of supply chain management. Specifically, we describe the interactions of scheduling with other planning functions and how these interactions determine the class of problem. Finally, we present a high-level problem statement and introduce the major scheduling decisions. Throughout the article, we will use lowercase Latin characters for indices, uppercase Latin bold letters for sets, uppercase Latin bold-italic for unit and production environments, lowercase Greek letters for parameters, and uppercase Latin letters for variables.

Scheduling in the PSE literature

Interestingly, one of the most popular early scheduling methods was developed in the late 1950s what later became known as the critical path method (CPM).[11] CPM, which is a popular project scheduling method, was used for construction scheduling at DuPont. However, chemical production scheduling was identified as an important research area only in the late 1970s.[12–14] The early work in the area focused on sequential processes which, as we will see in "Basic insight," have similarities with discrete manufacturing facilities.[15–19] The first methods to address problems in general chemical facilities, which we will later term *network* facilities, were developed in the early 1990s.[20–22] Since then, the number of publications in the area has increased dramatically. For example, in 2010, more than 35 papers were published in the PSE literature with the word "Scheduling" in their title (20 in *Industrial and Engineering Chemistry Research*, 11 in *Computers and Chemical Engineering*, three in the *AIChE Journal*, and two in *Chemical Engineering Science*). Furthermore, many researchers have been working on the integration of scheduling with other planning functions and problems; e.g., integration of production planning and scheduling,[23,5,7] integration of scheduling and control,[24–26] and simultaneous batch process design and scheduling.[27]

In general, scheduling methods have been classified in terms of process network structure and processing characteristics.[6,7,28] In short, approaches to problems in sequential facilities rely on the concept of precedences between batches, whereas approaches to general problems were based upon the concepts of state-task network (STN)[20] or resource-task network (RTN).[22] However, the last few years have seen a number of approaches that combine different modeling elements. For example: (i) RTN-type models, which have been traditionally used for general facilities, were used to address problems in sequential facilities[29]; (ii) RTN models, which were typically combined with common grid time representations, were used with unit-specific time representations[30]; and (iii) batching and scheduling decisions are considered simultaneously in multistage facilities.[31,32] The list of innovative combinations of modeling methods from different *schools of thought* is expected to grow in the following years as researchers develop more sophisticated methods to address new classes of problems and larger instances. This will inevitably lead to a diversification of approaches that will render the existing classification schemata inadequate. One of the goals of this article is to address this limitation.

Preliminary concepts

First, we note that the term scheduling was originally used to describe only the allocation of tasks to resources over time; that is, it considers only the timing of tasks. Thus, some of the first books on the subject treated sequencing and scheduling as two separate functions.[33,34] In contract, in the PSE literature, scheduling refers to a much wider problem: it encompasses the determination of the number of tasks, the assignment of tasks to resources, and the sequencing and/or timing of tasks.

In discrete manufacturing, the entities to be scheduled are *jobs*, $i \in \mathbf{I}$, and the shared resources are machines, $j \in \mathbf{J}$. A job may have to be allocated to a single step (operation) or it may require a number of steps, $k \in \mathbf{K}$. Also, it is most often assumed that a job is not split or that two jobs are not mixed into one. Problems are generally posed in the following *machine environments* (see Figure 1):

- *Single machine* (***1***): All jobs have to be processed on the same machine; this is a special case of all other machine environments.
- *Machines in parallel* (***P***): Each job i can be processed on exactly one of m parallel machines; there are three classes of environments with parallel machines, discussed later.
- *Flow-shop* (***F***): There are m machines and each job has to be processed on all m machines; all jobs follow exactly the same route. Equivalently, each job requires m steps, where each step can be performed on only one machine.
- *Job-shop* (***J***): There are m machines; each job has a predetermined route, $\mathbf{R}_i$, to follow, where $\mathbf{R}_i$ is an ordered set of machines, which may include all or a subset of $\mathbf{J}$, and it may also include the same machine more than once (recirculation).
- *Open-shop* (***O***): There are m machines and each job has to be processed on a subset, $\mathbf{J}_i$, but there are no restrictions in the routing of each job (the scheduler decides the route); that is, $\mathbf{J}_i$ is not ordered.
- *Flexible flow-shop* (***FF***): This is a generalization that combines the ***P*** and ***F*** environments. Each job has to visit a set of steps, $k \in \mathbf{K}$, in a predetermined and common among all jobs sequence, where each step consists of a set of parallel machines $j \in \mathbf{J}_k$. Note that the routing is in terms of steps, not machines; and the assignment of jobs to machines is a scheduler decision.
- *Flexible job-shop* (***FJ***): This is a generalization of the ***P*** and ***J*** environments. Each job has to go through a number of steps, where each step can be performed by multiple machines in parallel. Each job has to visit a different subset

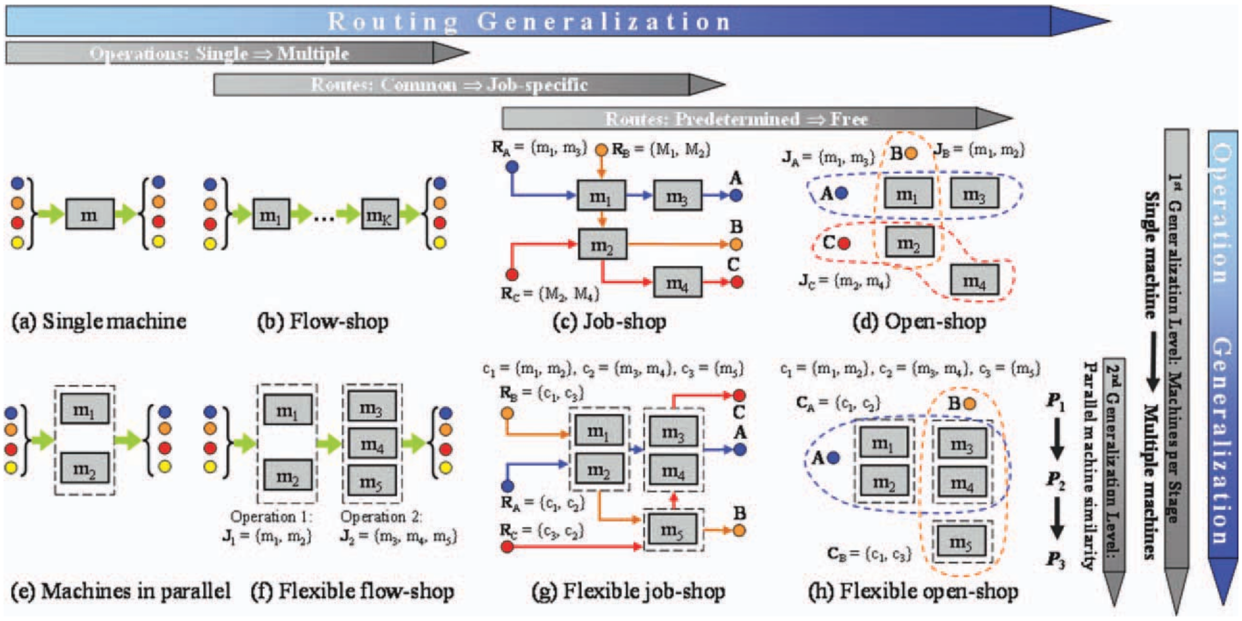

Figure 1. Machine environments in discrete manufacturing scheduling.

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

of steps in a job-specific routing. A machine may belong to multiple steps and/or different steps for each job. As in $\boldsymbol{J}$, the route of a job may include the same step more than once (recirculation).

• *Flexible open-shop* (***FO***): This is the generalization of the ***P*** and ***O*** environments.

An important observation is that there are two major ways in which machine environments are generalized. First, we have the routing of jobs. Going from ***1*** (which has no routing) to ***F***, we add multiple steps (and therefore routing), but all routings are (i) predetermined and (ii) identical among all jobs. Going from ***F*** to ***J***, we keep predetermined routings, but we relax the restriction of identical routings. Finally, in ***O*** we relax both restrictions. This generalization is shown as horizontal differentiation in Figure 1.

Second, we have the machines comprising the steps of the environment. In the simplest case each step consists of a single machine, which leads to environments ***1***, ***F***, ***J***, and ***O***. The first generalization is the introduction of parallel machines for each step, which results in the flexible counterparts of the previous four environments: ***P***, ***FF***, ***FJ***, and ***FO***. At a second level, there are three types of machines in parallel:

(a) Parallel identical machines ($\boldsymbol{P_1}$): the processing time of each job i is the same, τ_i, in all machines.

(b) Different parallel machines with specified speeds ($\boldsymbol{P_2}$): if υ_j is the speed of machine j, then, the processing time of job i in machine j is τ_i/υ_j; that is, the speed (effectiveness) of machines is relatively the same for all jobs.

(c) Unrelated machines ($\boldsymbol{P_3}$): the processing time of job i in machine j is τ_{ij}.

Obviously, $\boldsymbol{P_3}$ is a generalization of $\boldsymbol{P_2}$ which is a generalization of $\boldsymbol{P_1}$. We also note that in multistep environments (e.g., ***FF*** and ***FJ***) the parallel machines of each step may be of different type (e.g., step 1 in a ***FF*** can be of type $\boldsymbol{P_1}$ while step 2 is of type $\boldsymbol{P_3}$). The generalization of environments in terms of the machines comprising a step is shown as vertical differentiation in Figure 1.

Another important dimension is the suitability of a particular machine to perform one or more operations of a specific job. As this aspect is also important in chemical production scheduling, we will discuss it in detail in the next section. For now, we note that the typical assumption in ***F*** and ***FF*** is that each machine can perform a single operation. In ***F***, this means that there are as many machines as steps; and in ***FF***, it means that $\mathbf{J}_k \cap \mathbf{J}_{k'} = \emptyset$ for $k \neq k'$, and $\bigcup_{k \epsilon \mathbf{K}} \mathbf{J}_k = \mathbf{J}$, that is, $(\mathbf{J}_k,\ k \ \epsilon \ \mathbf{K})$ is a partition of $\mathbf{J}$. The same assumption is most often made in ***J***, but not all jobs have to go through all machines. In ***FJ***, the simplest case arises when machines belong to work centers, $c \ \epsilon \ \mathbf{C}$, and the routings of the jobs are in terms of work centers; that is, the routing, $\mathbf{R}_i$, of job i is an ordered set of centers. This case is shown in Figure 1g. However, in ***FJ*** more complex situations may arise because jobs have different routings, which means that the steps should be determined in terms of jobs. In some case, buffers between operations are used.

A scheduling problem is traditionally described by a triplet $\alpha|\beta|\gamma$, where α denotes the machine environment, β describes details of processing characteristics and constraints, and γ gives the objective function. Field β may include multiple entries, such as changeovers, storage/waiting constraints, precedence constraints, preemptions, machine-job suitability, recirculation, and so forth.

We note that there are obvious similarities between discrete manufacturing and some types of chemical production. Specifically, in chemical production a batch can be viewed as a job that has to be processed in several stages (e.g., reaction → separation → drying → packing) without splitting into multiple batches or mixing with other batches. However, as we will see in the next section, chemical production facilities can be in some ways more complex than the machine environments presented in this section.

Scheduling in the context of supply chain management

Production scheduling is only one of the many planning functions in manufacturing supply chains, which has three

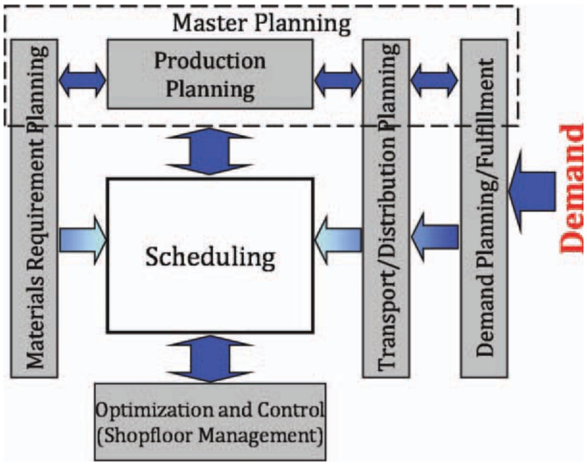

Figure 2. Interactions of scheduling with other planning functions.

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

major implications.[7,35,36] First, the interaction of scheduling with other planning functions, and production planning in particular, determines what type of decisions are made by the scheduler. Second, market considerations (including demand planning) combined with capacity constraints determine the production goal and the type of scheduling problem. Third, input parameters to scheduling (e.g., order data) are determined by other functions. A simplified diagram of the interactions of scheduling within the supply chain planning matrix is shown in Figure 2.

Vertical Integration. While there is no unique distinction between planning and scheduling decisions, it is generally accepted that planning provides scheduling with targets (e.g., orders with due dates) or fixes some decisions (e.g., number of batches). In other words, the types of decisions to be made at the scheduling level depend on the type of production planning problem; e.g., if only production targets are determined at the planning level, then batching, assignment and sequencing decisions have to be made at the scheduling level. Furthermore, production planning determines resource availability through, for example, unit maintenance planning. In short, the production planning function determines inputs to scheduling as well as the class of scheduling problem. In turn, scheduling interacts with optimization and control. It provides a schedule, which may trigger subsequent setpoint optimization and/or control, and receives information about the execution status of the scheduled tasks and the availability of resources which are used in rescheduling. A more in-depth discussion is presented in Harjunkoski et al[25].

Horizontal Integration. The volume and variability of demand often determines the production pattern in a facility. The production of high-volume products (or intermediates) with relatively constant demand typically induces periodic production and therefore *cyclic scheduling*. For high-volume products with variable demand but with good demand forecasts, short-term scheduling with a stock target (target inventory position) which should be maintained (*push* or *make-to-stock* policy) is required. Products with irregular and/or small demand (coming directly from customers or the adjacent production line) are produced to meet specific orders (*pull* or *make-to-order* policy), which typically means that a short-term schedule should be generated frequently.[37] In all cases, demand planning indirectly affects (through distribution and production planning) production scheduling. At the other end, procurement planning determines the availability of feedstocks or the release date of orders, which are also inputs to scheduling.

Market Environment and Master Planning. Market considerations and capacity constraints determine the overarching production goal; e.g., if production capacity is barely sufficient to meet the demand, then the goal is the maximization of throughput or the minimization of backlogs. However, this goal may not necessarily translate to a similar scheduling objective function, because production amounts can be fixed at the production planning level. We also note that the scheduling objective can change due to demand fluctuations or contract changes.

Remarks. In summary, the interactions of production scheduling with the other planning functions determine: (i) the type of scheduling problem to be solved (cyclic vs. short-term), (ii) the types of decisions to be made at the scheduling level, (iii) the scheduling objective, and (iv) input parameters. The presentation in this article focuses on short-term scheduling, but most of the concepts and ideas are applicable to cyclic scheduling as well. In terms of scheduling decisions, we will consider the most broad case. We assume that production planning provides only production targets (orders), so batching (number and size of batches), resource-task assignments, and task sequencing and timing are all made at the scheduling level. If some of these decisions are predetermined, this simply results in a special case of scheduling. The proposed framework and model classification are independent of the scheduling objective function. Finally, we will assume that shipments of raw materials and/or order release times (determined from requirements planning), shipments of products and/or orders with due dates (determined from production and distribution planning), and availability of resources (determined from production planning and optimization/control) are input parameters. The scope of our framework is shown in Figure 3.

Problem statement and scheduling decisions

The general chemical production scheduling problem can be posed as follows. We are given:

(a) Production facility data; e.g., processing and storage unit capacities, unit connectivity.

(b) Production recipes; e.g., mixing rules, processing times/rates, utility requirements.

(c) Equipment unit – task compatibility.

(d) Production costs; e.g., raw materials, utilities, changeover.

(e) Material availability; e.g., deliveries (amount and date) of raw materials.

(f) Resource availability; e.g., maintenance schedule, resource allocation from planning.

(g) Production targets or orders with due dates.

Item (a) above describes the scheduling hardware, item (b) describes how this hardware is going to be used, and item (c) is defined by the interaction of (a) and (b). Unless there is a retrofit or capacity expansion project or a major change in the product portfolio, items (a)–(c) can be thought to remain the same in all scheduling instances in a given facility. The same is true for some production costs (e.g., changeovers). However, costs such as raw materials and utilities (e.g., electricity) can vary. Finally, items (e)–(g) are

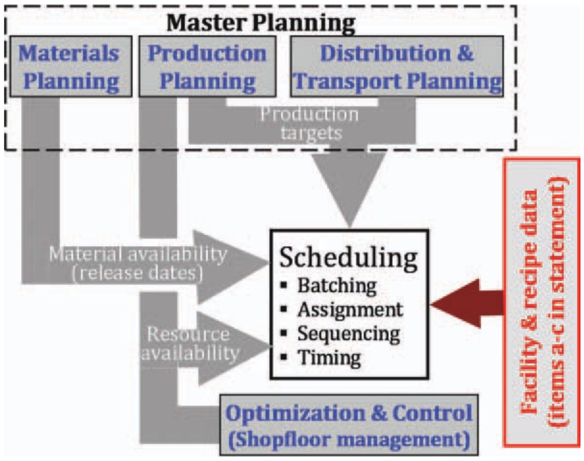

Figure 3. Information flow to scheduling and definition of inputs and scheduling decisions.

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

inputs to scheduling which are typically outputs of other planning functions and therefore are expected to change over time. Fixed inputs (i.e., items a–c) are depicted in Figure 3 via the red arrow, while variable inputs (items d–g) are shown as gray arrows (see Novas and Henning[38]).

Our goal is to find a least cost schedule that meets production targets (or orders) while satisfying resource constraints. For given production targets, alternative objective functions are the minimization of tardiness or lateness, which can be viewed as minimization of backlog cost; or the minimization of earliness, which can be viewed as minimization of inventory cost. If production beyond the given targets/orders is allowed or desired, then the maximization of profit or throughput can also be considered.

In the general problem, we consider in this article, we seek to optimize our objective by making four types of decisions (see Figure 4):

(a) Selection and sizing of tasks/batches to be carried out (batching).

(b) Assignment of tasks to processing/storage units or general resources.

(c) Sequencing of tasks on processing/storage units.

(d) Timing of tasks.

In the next two sections, we will further discuss how these decisions shape the proposed framework and model classification. Until then, we remark on the following. First, while there is a subtle difference in the way tasks are defined, both batch and continuous scheduling problems involve essentially the same type of decisions.

Second, when two tasks, i and i', are executed on a resource j that can perform at most one task at a time (unary resource), then they cannot be simultaneously processed. This means that any feasible solution should satisfy

$$\{s_{i'} \geq s_i + \tau_i\} \vee \{s_i \geq s_{i'} + \tau_{i'}\} \quad \forall i, i' \in \mathbf{I}_j \tag{1}$$

where $\mathbf{I}_j$ is the set of tasks to be scheduled on resource j. In some approaches, the condition in Eq. 1 is satisfied through sequencing (variables and constraints), while in others through the timing of tasks; that is, the mapping of tasks on a time reference grid. Eq. 1 does not have to be necessarily satisfied by tasks scheduled on resources that can process more than one tasks simultaneously.

Third, there is an inherent hierarchy in the aforementioned decisions. Specifically, a decision-maker has to first decide how many tasks have to be carried out (batching); the selected tasks have then to be assigned to units (assignment); next, tasks assigned to the same unit have to be sequenced; and last, starting times have to be determined. Finally, we note that timing in the aforementioned hierarchy is equivalent to scheduling in the early literature (see ‘‘Preliminary concepts’’).

Chemical Production Environments

In the previous section, we saw that the machine environment plays an important role in defining the discrete manufacturing problem classes. In this section, we develop a similar framework for chemical production scheduling. In ‘‘Basic insights,’’ we show that it is the constraints regarding the handling of the (typically fluid) materials, rather than the structure of the manufacturing facility, that determine the type of production environment. Next, we present what we consider to be the three broad types of chemical processing, and then we discuss other processing characteristics and constraints. Finally, we discuss how chemical production scheduling problem classes can be defined. In the remainder of the article, we will use the term ‘‘production environment’’ instead of ‘‘machine environment.’’

Basic insights

A key aspect in the basic framework we presented in ‘‘Preliminary concepts,’’ was that the tasks to be scheduled correspond to specific jobs. For example, in a ***FF*** with three

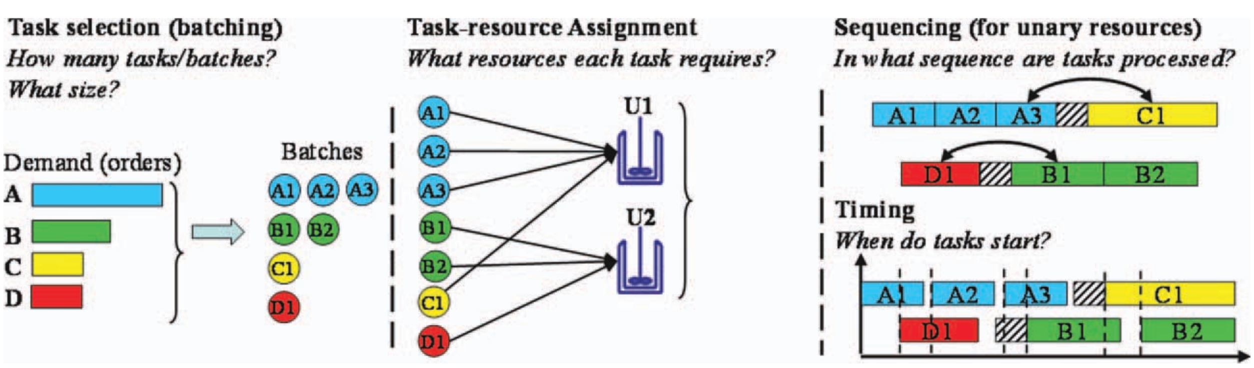

Figure 4. Major decisions in chemical production scheduling.

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

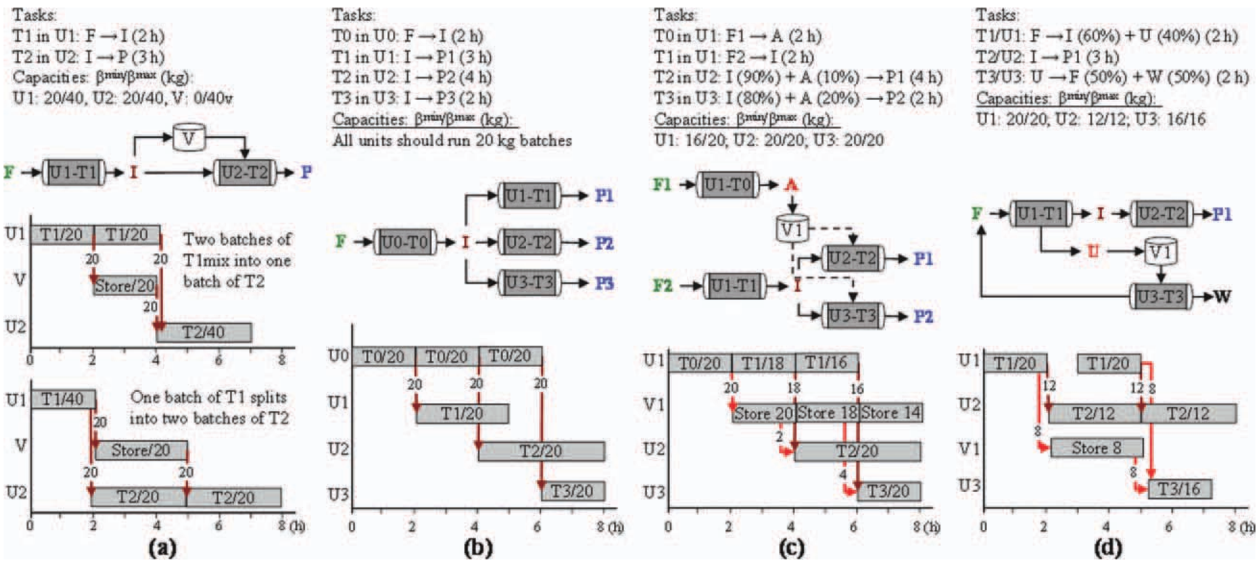

Figure 5. Examples of production environments.

(a) Network processing in a facility with seemingly sequential unit structure. (b) Sequential type of processing in facility with an arborescence-like unit structure. (c) Sequential processing (no splitting/mixing of I) combined with material mixing. (d) Sequential processing (no splitting/mixing of I) combined with production of multiple materials. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com]

jobs, $\mathbf{I} = \{A, B, C\}$ and two operations, $\mathbf{K} = \{1, 2\}$, each one consisting of two machines, $\mathbf{J}_1 = \{M1, M2\}$ and $\mathbf{J}_2 = \{M3, M4\}$, there are six tasks to be scheduled: A-1 (job A in operation 1), A-2, B-1, B-2, C-1, and C-2. This type of sequential processing is found in chemical production scheduling problems where the identity of a batch has to be maintained through a series of operations and some additional restrictions are present. These problems can be classified and addressed for the most part using the concepts presented in "Preliminary concepts." However, in problems without these restrictions, this sequential structure is not present and the notion of a batch going through multiple operations is not defined. Therefore, a different representation is needed.

We will illustrate the basic concepts using the examples shown in Figure 5. Before we do so, we note the following in relation to the traditional notation discussed in "Preliminary concepts":

(a) We will use the term unit, instead of machine, to describe a resource that can perform at most one task at a time (unary resource). Units can be further grouped into processing units and storage units (vessels).

(b) In our discussion of sequential processes, we will use the term stage, instead of operation, to denote a distinct processing step in a facility (e.g., fermentation, centrifugation, etc).

(c) We will use the term batch, instead of job, to denote the entity that has to be processed in different stages.

The first crucial insight is that the type of production environment is not determined by the structure of the facility (i.e., the units and their interconnections). Figure 5a shows a facility where a single product is produced following a perfectly linear pattern, where each stage consists of a single dedicated unit and each task consumes and produces a single material. Nevertheless, the concept of a batch is not defined if there are no mixing and splitting restrictions, as illustrated by the two Gantt charts shown in Figure 5a. To further illustrate the point, we consider the facility in Figure 5b, where a single unit is used to convert a raw material to a single intermediate which is then converted to three different products in three different units. The key restriction here is that the intermediate has to be transferred into a single unit to be converted to a final product; that is, a batch of I cannot be mixed with other bathes nor split toward multiple batches in units U1, U2, and U3. Thus, this is a sequential process, which interestingly can be viewed as a flexible flowshop with two operations (one machine in the first and three machines in the second) and three jobs (assuming one batch per product). These two examples show that it is the material handling restrictions (mixing and splitting of batches) rather than the structure of the facility that determines what type of processing we have, and therefore, what types of representations and models are necessary. This issue, which was first raised in Ref. [39], is further discussed in Appendix A.

Second, the type of processing is not necessarily determined by the number of materials used as input or output. Consider the process shown in Figure 5c, where two final products, P1 and P2, are produced by mixing intermediate I and additive A. The key in this process is that a batch of I has to be consumed by a single batch of T2 (to produce P1) or a single batch of T3 (to produce P2); it cannot be mixed with other batches of I nor split towards many batches of T2 and/or T3. This implies that we have sequential type of processing although a task consumes two materials. However, a batch of additive A, which is produced in the same unit as intermediate I, can be stored and used in multiple batches of T2 and/or T3. This implies that additive A should be treated as a material used in network-like processing.

Alternatively, consider the process shown in Figure 5d, where task T1 is a reaction resulting in a pure intermediate I, to be converted into final product P1 by task T2, and a mix U which consists of unreacted feed F and waste W. The amount of intermediate I produced by one batch of T1 has to be consumed by exactly one batch of T2. Neither mixing (with other batches of T1) nor splitting (toward multiple batches of T2) is allowed, which implies sequential processing although there are two output materials. At the same time, U from multiple batches of T1 can be stored and when enough material is accumulated sent to separation T3 which yields pure F and waste stream W. In this case, a batch of T1 yields two output materials with different material restrictions.

These examples show that it is the material handling constraints that determine the production environment. They also show that the major difference between chemical production and discrete manufacturing scheduling is that tasks consume and produce fluids which can be mixed or split. Interestingly, material handling constraints (no mixing/splitting) that remove this *fluidity* make chemical production similar to discrete manufacturing. Not surprisingly, it is these constraints that lead to sequential processing, which as we will see in the next subsection can be viewed through the lenses of the framework presented in "Preliminary concepts."

We close with two final remarks. First, we note that most chemical facilities consist of multiple types of processes. For example, it is common to have sequential processes (e.g., fermentation) followed by network processes (e.g., mixing with different additives), followed by continuous processing (e.g. drying or packing). Second, it is common to have materials that have to be produced in a single batch, but can then be used in multiple lots, which means that mixing (blending) is not allowed but splitting is. As a facility may include different types of subsystems, it is more appropriate to talk about types of processing rather than facilities of a single type of environment.[39] Finally, based on the insights in this subsection, we propose three types of processing: sequential (Figure 5b), network (Figure 5a), and hybrid (Figures 5c,d).

Sequential processing

A process is sequential if it produces/consumes a single material and batch mixing and splitting are not allowed for both the input and output materials. A process can also be considered sequential if it consumes materials that can be coming from multiple batches (batch mixing) or produces materials that can be used in multiple downstream batches (batch splitting), but these materials do not have to be accounted for; e.g., they are always available if they are inputs or can always be disposed/stored if they are outputs (see discussion in "Hybrid processing"). The key issue here is that these processes can be modeled via batches (jobs) and stages (operations), using the framework discussed in "Preliminary concepts." Open-shops are not relevant because chemical transformations have to be carried out in a given sequence. Hence, we will focus our discussion on two multiproduct environments that are similar to the ***FF*** and ***FJ*** environments.

The first is the so-called *multistage* facility, where each batch of each product has to be processed in a number of stages, each stage consists of one or more parallel processing units, and a unit belongs to a single stage. A batch cannot be mixed or split and all products go through the stages in the same sequence, unless a product is not processed in a given stage. The scheduling of these facilities is similar to the scheduling of ***FF***s. The second is what in this paper we call multipurpose facility. It is a facility where batches cannot be mixed or split and have to go again through multiple stages, but the sequence of stages is product-specific. Also, a processing unit may belong to different stages depending on the product, and/or a unit may belong to multiple stages.

First, we remark that the general problem is posed in terms of facility (e.g., unit capacities) and product (e.g., recipes) data, as well as raw material and resource availability, and product demand (see problem statement). It is not expressed in terms of batches. If the batching problem is solved independently, then scheduling can be expressed in terms of batches. In this case, and if there are no utility and storage constraints, then problems in multistage and multipurpose facilities are practically equivalent to ***FF***s and ***FJ***s, respectively. The majority of approaches to these two problems in the process engineering literature have considered this narrow case. The introduction of storage and/or utility considerations preserves the batch/stage-based representation, but introduces additional features and constraints. However, if batching decisions are considered at the scheduling level, then the problem cannot be viewed as a flexible flowshop or jobshop because there is no fixed number of jobs. Maravelias and coworkers have addressed the general problem in sequential facilities.[31,32,40]

Second, we build upon the ideas presented in "Preliminary concepts" to represent the general scheduling problem in sequential production environments. Specifically, in addition to routing restrictions (horizontal differentiation in Figure 1) and different levels of operation complexity (vertical differentiation), we consider the suitability of a unit to perform a task of a given product as a third dimension.

We start with the multistage problem. The simplest case arises when all products go through all the stages and all units are compatible with all products (see Figure 6a). In this case, we only need to define the set of stages, $k \in \mathbf{K}$, and the units in each stage, $j \in \mathbf{J}_k$. The first generalization results when a product, $i \in \mathbf{I}$, does not have to be processed in one or more stages (see Figure 6b). In this case, which can also be viewed as multipurpose, we have to define product-specific routings, $\mathbf{R}_i \subseteq \mathbf{K}$, to correctly enforce sequencing. Note that, in the absence of preemption, this case is not equivalent to the previous one with some processing times being equal to zero. The second generalization is obtained from product-unit compatibility constraints (Figure 6c). To represent this, we introduce subsets $\mathbf{J}_{k,i}$ which include the units in stage k that are suitable for product i.

In multipurpose facilities, we have product-dependent routings, $\mathbf{R}_i$. The simplest case arises when units belong to the same work center (Figure 6d). If work centers are viewed as independent stages, then, this case is modeled by defining routings $\mathbf{R}_i$ and $\mathbf{J}_k$ (Figure 1g shows the same case with routings defined in terms of centers). A first extension arises when we have routings with reentries; that is, units belong to multiple stages for the same product. If the reentering product undergoes the same type of operation, then we simply add the same stage more than once in the routing of the product; that is, $\mathbf{R}_i$ is an ordered multiset (Figure 6e). If the re-entering product undergoes a different operation, then we add a new stage in the routing $\mathbf{R}_i$, and represent processing times in terms of stages. To illustrate, we consider an environment with two units, U1 and U2, and product P which has to go through three operations O1 in U1 $\rightarrow$ O2 in U2 $\rightarrow$ O3 in U1. This problem can be modeled by defining routing $\mathbf{R}_\mathrm{P} = \{1, 2, 3\}$ with $\mathbf{J}_{\mathrm{P},1} = \{\mathrm{U1}\}$, $\mathbf{J}_{\mathrm{P},2} = \{\mathrm{U2}\}$, and $\mathbf{J}_{\mathrm{P},3} = \{\mathrm{U1}\}$, and the associated processing times $\tau_{\mathrm{P},1}$, $\tau_{\mathrm{P},2}$, and $\tau_{\mathrm{P},3}$. The most general case results when units belong to different stages depending on the product they process (Figure 6f). This case can be represented using subsets $\mathbf{J}_{i,k}$, where stage numbers depend on products. Note that all

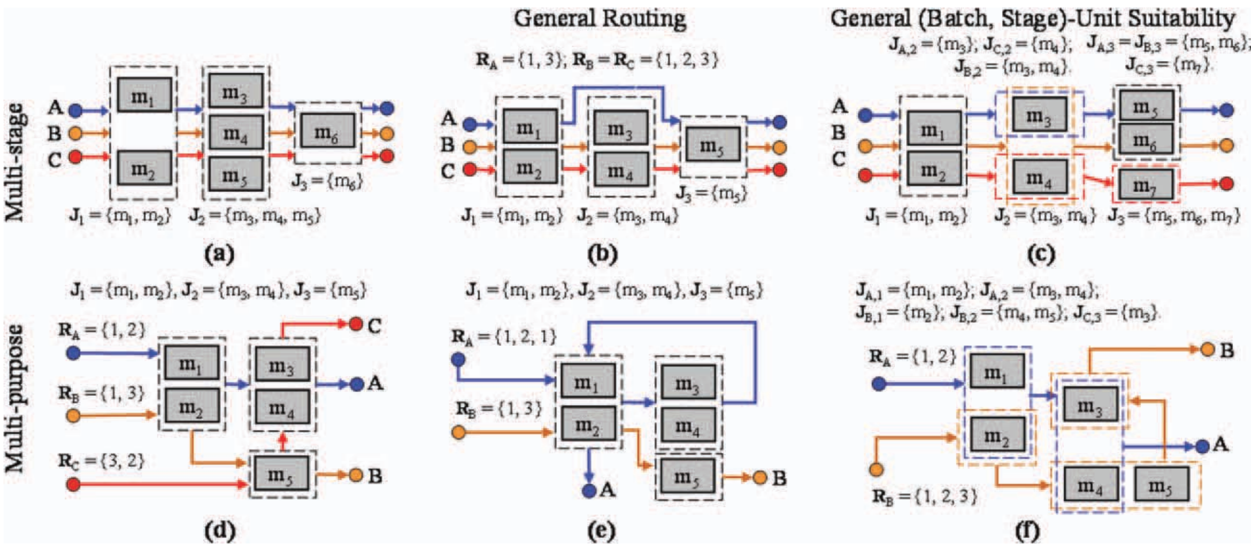

Figure 6. Sequential production environments.

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

sequential facilities can be viewed as special cases of multipurpose with reentries, and unit suitability constraints that depend on product and stage.

To summarize, the structure of any sequential facility can be stated in terms of product routings, $\mathbf{R}_i$, and product/stage-unit suitability information, $\mathbf{J}_{i,k}$. If the batching problem is solved independently, then we can simply replace products with batches; that is, routings and unit suitability are defined for batches (which means that multiple batches may have identical $\mathbf{R}_i$ and $\mathbf{J}_{i,k}$).

Network processing

Network processing arises when there are no restrictions in the way all input and output materials are handled; that is, multiple batches of the same task can be mixed or material produced by a single batch can be consumed by multiple batches. As illustrated in ‘‘Basic insights’’ network processing can arise in facilities of almost any structure. In the absence of other processing restrictions (e.g., storage, unit connectivity, transfer equipment, etc), network processing basically means that materials can flow freely from storage vessels (suitable for their storage) to processing units where they are consumed, and from processing units where they are produced to storage vessels.

First, we note that the notion of a batch of a product going through different stages is irrelevant in networks because there is no requirement for a batch to maintain its identity. Similarly, processing stages are not typically used because products can be produced in different ways which further implies that batching decisions have to be made. Clearly, since the number of tasks to be scheduled is variable, the framework in presented in the previous subsection, which relies on fixed products (or batches) and stages, is inadequate to address these problems.

Second, the types of tasks are typically defined in terms of input and output materials. In other words, a task is identified by the materials it consumes and produces and the corresponding conversion coefficients, rather than a product and a stage (as in sequential processes). Conceptually, tasks can be defined using a matrix where tasks correspond to columns, materials to rows, and the matrix elements are the corresponding coefficients. It is typically assumed that a task has a unique set of conversion coefficients. The two most widely used representations of network processes, the STN[20,21] and RTN,[22] rely on the consumption/production of materials (modeled either as states in STN or resources in RTN) by tasks. STN and RTN type approaches have been used to address problems with a wide range of additional characteristics, such as utilities, material transfer, and storage. However, these characteristics are not specific to network processing.

Hybrid processing

We use the term *hybrid* to describe processing that is not sequential nor network. For example, a process is hybrid if (i) a task consumes multiple materials or produces multiple materials and some of them have mixing/splitting restrictions; or (ii) a proper subset of materials produced or consumed by a task have mixing or splitting restrictions. In ‘‘Basic insight,’’ we considered two simple examples of such processes in Figures 5c,d. The example in Figure 5c is representative of problems where a task consumes multiple materials (i.e., we have material mixing), but mixing of batches of the same material is not allowed for at least one input material. Although not discussed in the literature, this type of processing is not uncommon in the process industries. For example, in the production of emulsifiers in the food industry, an additive is added to a batch before spray tower processing. We remark here the fine distinction between mixing of batches (of the same material) and mixing of materials. The former refers to the case where the output of two batches of the same task is mixed, which if allowed leads to network processing even if a task produces a single material and this material is consumed by a single task (see top Gantt chart in Figure 5a). The latter refers to the case where a task requires two inputs which therefore have to be mixed. The difference between the two is not fully understood in the literature, where it is often incorrectly assumed that network processing is caused by material mixing.

The example in Figure 5d illustrates situations where the main product of a task should be processed, without being mixed or split, by a downstream task, but a byproduct also has to be handled. For example, centrifugation of a batch coming from fermentation results in a liquid concentrate that has to be processed (without mixing or splitting) further, and a byproduct aqueous stream that has to be treated. Also, in fermentation processes (e.g., beer production) it is common to remove yeast before the end of the batch, which is going to be used in multiple subsequent fermentations. In this case, we note the difference between batch splitting and a batch producing multiple materials. The former arises when the output of the same batch is consumed in multiple subsequent batches. This leads to network processing even if only one material is produced (see bottom Gantt chart in Figure 5a). The latter situation arises when a task produces multiple outputs. As we illustrated in Figure 5d, production of multiple materials can be combined with no mixing/splitting restrictions.

The key insight here is that there exist processes where batch identity has to be tracked and preserved for some materials (e.g., intermediate to be processed further downstream), while other materials (additives, promoters, byproducts, etc.) can be handled more freely. We propose the term hybrid processing to describe this type of environment. We note that the use of word hybrid in this article is not the same as in Sundaramoorthy and Maravelias.[39] Here, we use it to describe a general type of processing which is independent of the representation or modeling of the process. Sundaramoorthy and Maravelias used it to define a type of material in their representation. Note that in hybrid processes, as well as in most sequential processes, the size of the batch changes as it moves along the various stages.

Finally, we note that in some cases, hybrid processing can be treated as sequential processing. The example in Figure 5c can be modeled as a sequential process if the additive will always be available (e.g., if produced in a dedicated unit with excess capacity) and thus does need to be modeled. The example in Figure 5d can be simplified by ignoring the byproduct if it does not require any further treatment or the resources for this treatment will always be available. While similar simplifications can be applied to many problems of this type, in the general case we need methods that preserve batch integrity while monitoring inventory levels. Interestingly, this general class of problems has received no attention in the literature and cannot be addressed by existing methods. Some special cases can be modeled using the UOPSS framework proposed by Kelly and Mann,[41] while the modeling approach of Sundaramoorthy and Maravelias[39] can potentially be extended to account for these types of processing restrictions.

Other processing characteristics and constraints

Chemical production facilities often exhibit special processing characteristics, such as auxiliary equipment, utilities, labor, unit setups, and changeovers between batches, storage and transfer constraints, and so forth. It is important to note that all processing constraints can, in principle, be present in any type of facility (sequential, network, or hybrid), although most of them have been studied in the context of network problems only. A thorough discussion can be found in Mendez et al.[6] Here we briefly review the major characteristics.

General Resource Constraints. Our discussion thus far has focused on task-unit assignments and task–task sequencing decisions because it is assumed that the major type of shared resources are the processing units. However, chemical production often requires a wide range of resources such as auxiliary units (e.g., piping and storage vessels) and utilities [e.g., steam, water, cleaning-in-place (CIP)]. Resources can be discrete (e.g. labor) or continuous (e.g. steam). We note that in RTN, materials are also viewed as resources consumed (inputs) and produced (outputs) by tasks.

Switchovers. There are two types of switchovers: sequence-independent, henceforth referred to as *setups*; and sequence-dependent, henceforth referred to as *changeovers*. In batch processing, they are typically activities that have to be carried out before a processing activity starts (e.g., unit cleaning). In continuous processes, they typically represent transitions between steady states (e.g., grade transition in polymer manufacturing).[42] They lead to down-time and sometimes to costs. Switchovers may require shared resources, such as CIP, which means that they are resource-constrained as are the processing tasks.[43]

Material Handling and Storage. As we have already seen, material handling constraints are very important because they lead to different types of problems. It is important to make a distinction here between what, in the scheduling literature, is called ''storage constraints'' and what, in this article, we term ''material handling constraints.'' The former refer to (i) storage capacity constraints (number and size of storage vessels), and (ii) constraints on the duration a material can be stored (in a processing unit or storage vessel). The combination of these two aspects is often referred to as storage policy (see Ref. [31]). The latter refers to an attribute of a material, which is typically specified by the production recipe or quality control requirements. Material handling restrictions are thought to be independent of the specific storage and transfer units of a facility.

Material Transfer. Transfer operations play an important role in the process industries. In a few applications, such as pipeline and crude oil scheduling, the major operations to be scheduled are transfer operations rather than production ones. In other cases, where transfer operations are highly constrained (e.g., transfers using cranes in metallurgical facilities), they should be scheduled simultaneously with production. Finally, most problems involve charging and withdrawing operations which, if resource-constrained, should be modeled because they may lead to connectivity restrictions. Furthermore, transfer operations are linked with storage levels; e.g., a withdrawing operation cannot start unless the inventory is above a threshold level.[41,44] Finally, we note that material storage and transfer constraints when coupled with unit capacity constraints may induce a type of processing.

Problem classes

In ''Preliminary concepts,'' we saw that a scheduling problem is traditionally defined in terms of a triplet $\alpha/\beta/\gamma$. Following the same convention, we can define a class of chemical production scheduling problem in terms of triplet $\alpha/\beta/\gamma$ where (see Figure 7):

- α denotes the production (rather than machine) environment. The main production environments are sequential-multistage (***SS***), sequential-multipurpose (***SP***), network (***N***), and hybrid (***H***). Sequential production environments can be further divided into the ones shown in Figure 6. As different types of processing can be present in the same facility, the choices of production environment also include combinations of the three processing types; that is, α may include multiple entries.
- β denotes the processing restrictions and characteristics. Some are similar to those in discrete manufacturing, e.g. setups, changeovers, release/due times; while some are specific

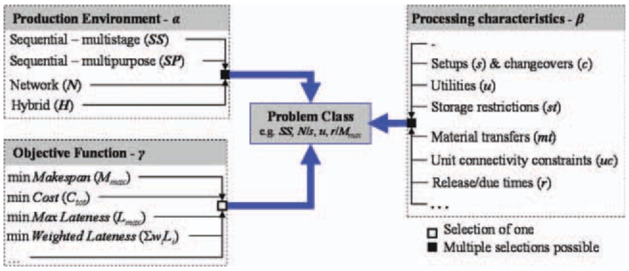

Figure 7. Problem classes in chemical production scheduling.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com] Partial list of processing characteristics and restrictions (β), and objective functions (γ) are shown.

to chemical production, e.g., storage constraints, material transfers, and so forth. Moreover, multiple processing characteristics and restrictions can be present simultaneously; that is, β may include multiple entries (see Figure 3 in Mendez et al.[6]).

- γ denotes the objective function; e.g., makespan minimization ($M_{\max}$), production cost minimization (C_{tot}), weighted lateness minimization ($\Sigma\ w_iL_i$), and so forth.

Modeling Approach Classification

The goal of this section is to present a classification of modeling approaches. Before we delve into the details of the proposed schema, we briefly discuss three closely related concepts, which, in this article, we term problem, problem representation, and modeling approach. A scheduler in a plant is faced with a problem. Problems can be posed in a myriad of different ways, depending on the production environment, the interaction of the scheduling function with the other planning functions, the tools, and even the culture of the company. To address it, the problem has to be converted into a standardized form that can be used in software tools and understood by the people who will be using the solution. This is what we term ''problem representation.'' Although it often depends on the problem and the method that will be used to address it, the problem representation is, in principle, independent of those two. Also, the problem representation can be simpler than the actual problem because noncritical activities or nonbinding constraints are often ignored (e.g., small setup times, auxiliary materials that can be assumed to be always available, etc). The development of a general problem representation for chemical production scheduling remains an open question, primarily due to the lack of unified notation (see Appendix B). A first attempt to address this challenge can be found in Novas and Henning.[38] Also, Hai et al.[45] propose an ontology-based approach for process modeling which can be used toward the development of a general problem representation. For a given a problem representation, one can use different modeling approaches. For example, problems in a multistage process represented using the concepts of products, stages, and units presented in ''Sequential processing,'' can be addressed using different modeling approaches as we will see in ''Modeling of time.'' Finally, there is a subtle distinction, which we will not discuss here in detail, between models and formulations. Specifically, multiple formulations can be developed for the same model. For example, there are different ways in which assignment constraints can be expressed in STN models; e.g., Eq. 1 in Ref. [20] vs. Eq. 9 in Ref. [21]. The focus of this section is on modeling approaches, and MIP-based approaches in particular.

The proposed classification is based on three attributes: major scheduling decisions, modeling elements, and modeling of time. The first two, which are intimately interrelated, are discussed in ''Scheduling Decisions and Modeling Element,'' while the third one in ''Modeling of time.'' In ''Modeling approach space,'' we present the proposed classification schema, and in ''Relationship between problem classes and modeling approaches,'' we briefly discuss the relationship between problem classes and modeling approaches.

Scheduling Decisions and Modeling Elements

Our discussion of optimization decisions here focuses on the decisions that are necessary to model the allocation of processing units, assumed to be the primary resources, to tasks. Furthermore, we note that the type of decisions made at the scheduling level are often determined externally. For example, as discussed in ''Preliminary concepts,'' some of the decisions we consider to be scheduling decisions can be made at the production planning level and therefore are inputs to the scheduling function. However, the decision-maker often has the flexibility to use a range of solution approaches, each with different sets of decisions. For instance, for a problem that involves all levels of decisions, one could use heuristic methods to determine the number and size of batches followed by a scheduling approach that involves only assignment and sequencing decisions.

At the one extreme, we have approaches that consider only sequencing and timing of tasks. Single-unit problems with changeovers, which often lead to the famous travelling salesman problem, are such examples. In the middle of the spectrum, we have approaches that are concerned with task-unit assignment and task sequencing. In ''Sequential processing,'' we saw that most approaches to problems in sequential environments are concerned with these two decisions. At the other extreme, we have methods that account for batching, assignment, and sequencing/timing decisions. Material-based approaches for problems in network environments are such examples. Recent approaches for the scheduling of sequential processes also fall in this group.[46,47,31]

The type of decisions considered often determines the second major modeling attribute which is the selection of the entity modeled to enforce resource constraints. If batching decisions are fixed, then the resulting problems are modeled using what we term *batch-based* approaches, while *material-based* approaches are most often used to address problems where batching decisions are considered (see Sundaramoorthy and Maravelias[39] for a more thorough discussion).

In batch-based approaches with fixed number of batches, variables are defined for batches, assignment constraints are expressed for every batch in every stage, and sequencing constraints are expressed for every pair of batches as well as for the same batch between two consecutive stages. In other words, the modeling of the scheduling problem is based on the notion of a batch; material types and amounts are not defined. Interestingly, RTN-based approaches that were recently proposed to address such problems treat batches (and not materials) as resources that are consumed and produced at different stages; that is, they also rely on the modeling of batches.[29]

In material-based approaches, the number and size of batches are optimized. Furthermore, sequencing is accomplished through material balances: a task can start only if the intermediate materials consumed by it are available, which means that the tasks producing these materials should have already been performed (assuming that we start from zero intermediate inventories). We note that the sequencing through materials is advantageous because the notion of precedence cannot be defined in network processes due to the existence of initial intermediate inventories and recycle streams. We also point out that tasks are still modeled (using binary variables); linked to batchsizes (typically using variable bound constraints); and, used to enforce other types of restrictions.

A high-level classification of existing approaches in terms of scheduling decisions and modeling elements is shown in Figure 8. We close with three remarks. First, we note that our discussion in this subsection focused on the elements used to enforce resource constraints on processing units (unary resources). Of course, there is a wide range of other resources (e.g., utilities), but all these entities can in principle be combined with the two primary elements, materials and batches, discussed here. Second, the use of batches or materials is not exclusive. Maravelias and coworkers presented models which, while not including material balances, implicitly accounted for materials through batching decisions and batch size constraints.[46,32,31] Third, we note that what we presented in this subsection is only a first attempt to classify existing modeling approaches in terms of decisions and major modeling elements. As new scheduling methods are likely to emerge, we will have to expand the scope of these attributes.

Scheduling Decisions

Modeling Elements	Sequencing/timing	Unit-batch assignment Sequencing/timing	Batching (no & size) Unit-batch assignment Sequencing/timing
Tasks	Sequential •Single-unit	Sequential •Single-stage •Multi-stage	
Materials			1. Network 2. Network & sequential processes
Tasks & materials			Sequential with batching decisions

Figure 8. Classification of existing modeling approaches in terms of modeling elements and scheduling decisions.

Production environments addressed using each combination of scheduling decisions and modeling elements are also given.

Modeling of time

The modeling attribute that has received the most attention in the PSE literature is arguably the so-called time representation, which is, for the most part, understood as a selection between discrete and continuous time representation. In this article, we offer a broader discussion of the issue of modeling of time, of which the distinction between discrete and continuous representation is only one facet. In particular, we propose that there are four components (or levels) in the modeling of time:

(1) Selection between precedence-based and time-grid-based approach.

(2) Selection of type of precedence relations (local vs. global) or type of time grid approach (unit-specific vs. common).

(3) Specific assumptions.

(4) Selection between discrete and continuous time.

Precedence vs. Time Grid. In general, there are two ways to sequence and/or time tasks. The first is through the enforcement of binary precedence relationships between tasks executed on the same unit. Regardless of the specifics of the formulation, this approach requires the expression of constraints between pairs of tasks. In MIP formulations, this is accomplished via the introduction of sequencing binaries for pairs of batches, and the expression of sequencing constraints. Variables representing time may be used, but they represent times specific to a batch (e.g., starting times in a stage) and appear, again, in constraints involving pairs of batches or stages of the same batch; they are not related to any external time frame. Also, if the number of batches, and therefore the tasks to be scheduled, is known then the size of the formulation is fixed.

The second approach uses one (or more) time grid onto which events (e.g., the beginning of a task) are mapped. This time frame is independent of the specific tasks scheduled to take place; it has a given number of (fixed or variable) time points which do not necessarily correspond to task-specific time variables, which are defined separately and matched with points of the time grid. Also, what makes this approach different from the previous one is that (some attributes of) the time grid(s), which determine the size of the formulation, have to be defined by the modeler before optimization. For example, the modeler has to decide how many time points the time grid will have.

Interestingly, the majority of methods developed to address problems in sequential processing adopt a precedence-based approach, while all approaches to network processes adopt a time grid approach. A third option, which will not be included in our discussion because it is primarily used in production planning, is a lot-sizing-type of representation. This representation is often used when scheduling of sequential processes is integrated with production planning, and there are no sequence-dependent changeovers.[48–50] It is also used, in conjunction with one of the previous representations, in decomposition methods where an approximation or relaxation of the scheduling problem is considered.[51]

Type of Precedence/Time-Grid and Specific Assumptions. The selection of a precedence-based or time-grid-based approach is the first level of time modeling. Once this decision is made, the next level concerns the manner in which sequencing constraints are expressed in the former, and time matching between the time grid and the tasks is performed in the latter.

(1) Precedence-based modeling. In general, there are two broad approaches: one where precedence relationships are established between all pairs of tasks assigned to the same unit (global precedence), and one where they are established only between pairs of tasks processed consecutively on the same unit (local precedence). The basic difference between the two approaches is illustrated in Figure 9a. Given the type of precedence, the user can then decide to use different types of timing variables (e.g., start time vs. end time) and different assumptions (see Figure 9b). Most assumptions can be used within both global and local approaches although the corresponding MIP constraints can be different.

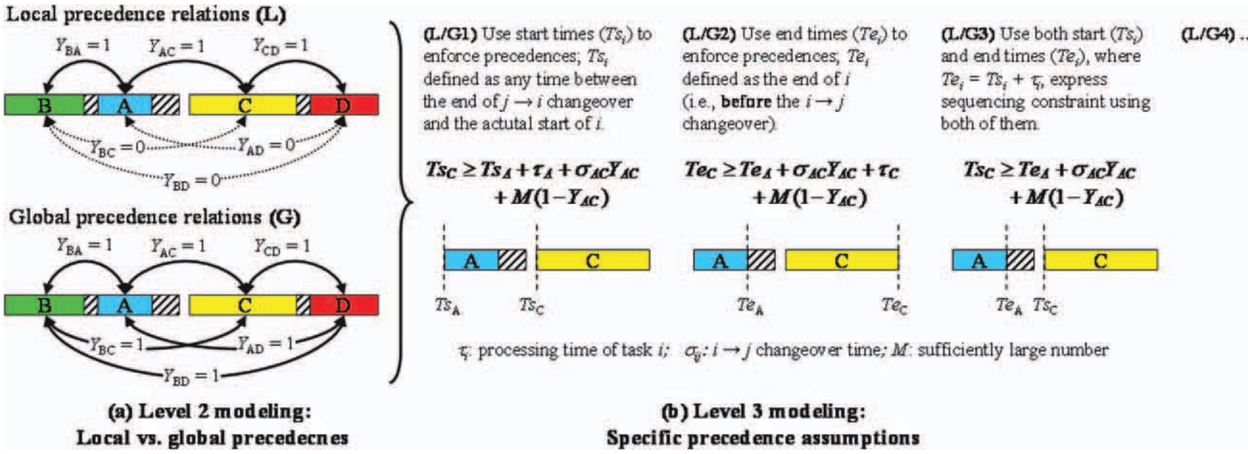

Figure 9. Illustration of the two major precedence-based approaches and corresponding assumptions.

For simplicity, assumptions in (b) are shown to be common between local and global approaches. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com]

(2) Time-grid-based modeling. The generation of models using time grids has been traditionally viewed in terms of: (i) the selection of discrete or continuous time representation, (ii) the selection of a common or (multiple) unit-specific time grids, and (iii) time grid assumptions. Here, we suggest that the selection between discrete and continuous time is relevant for all types of modeling approaches, so we discuss it separately in the next paragraph. Also, items (ii) and (iii) have been thought to be relevant only to continuous-time formulations, but in principle they can be studied in the context of discrete-time models as well. In fact, the mixed-time representation of Maravelias[52] can be viewed as a discrete-time model (since the grid is fixed) with different assumptions regarding the duration of processing times. The difference between common and unit-specific approaches is shown in Figure 10a. In addition to the grid type selection (i.e., the second level), there are multiple ways to model the relationship between the time grid(s) and the (beginning and end of) tasks, which is illustrated in Figure 10b. The specifics of the various combinations of assumptions proposed in the literature are not discussed here.

Discrete vs. Continuous Representation. Time representation has been studied only in relation to approaches that use time grids, because precedence-based approaches have been thought to be, by default, continuous-time approaches. However, any precedence-based approach (where time variables are continuous) can be converted into a discrete-time approach by simply adding integrality constraints on the time variables. In fact, constraint programming models constructed using modeling environments (e.g., OPL Studio®) are often precedence-based models combined with discrete-time representation. More generally, any approach is a discrete-time approach if the timing variables are integers, which means that any time-grid-based MIP formulation with time matching constraints becomes a discrete-time approach (albeit rather inefficient) if timing variables are constrained to be integers. In other words, the selection of the time grid and the assumptions regarding the grid (modeling levels 2

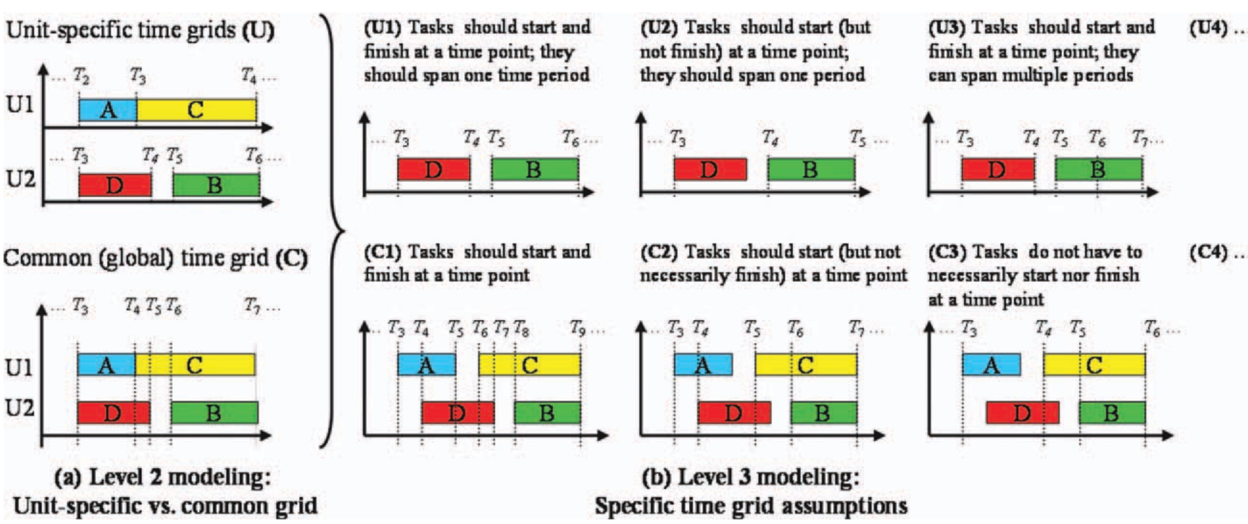

Figure 10. Illustration of second (unit-specific vs. common time grid) and third (specific assumptions) levels of modeling of time in time-grid-based approaches.

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

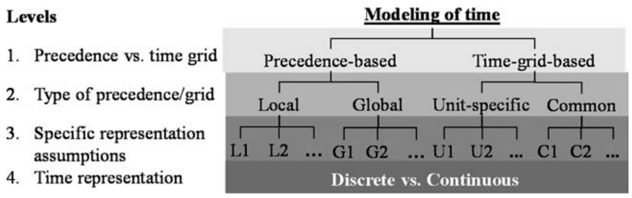

Figure 11. Components (levels) for the modeling of time.

and 3) are decoupled from the discrete vs. continuous selection. Of course, from a computational point of view, one would like to exploit knowledge about the nature of time variables to develop a more effective formulation. Finally, we note that, although viewed from a different angle, the notion that a discrete-time model can be obtained as a special case of a continuous-time model was first discussed in Maravelias and Grossmann.[53]

Hierarchy of Time Modeling. Based on the four components discussed above, we can construct the hierarchy shown in Figure 11, which includes a wide range of models. Of course, some of these models are expected to be less efficient than others, so they have not been and are unlikely to be studied in the future. However, this treatment allow us to better understand what decisions go into the modeling of time.

Modeling approach space

The proposed classification can be viewed as follows: the *universe* of modeling approaches to chemical production scheduling is a three-dimensional (3-D) space where each dimension corresponds to a key attribute:

(a) *Optimization decisions*: the major decisions are batching, assignment, and sequencing/timing; all approaches involve a subset of these three.

(b) *Major modeling elements*: these typically include batches (or orders), materials, or both batches and materials.

(c) *Modeling of time*: this includes the four levels discussed above.

The coordinates of an approach are basically its classification. They also yield the modeling techniques that were employed. For example, approaches with variable number of tasks include batching decisions which means that materials, and their corresponding amounts, have to be modeled; and conversely, models with fixed number of batches, require no batching decisions and thus materials should not necessarily be included. A graphic illustration of the proposed classification is shown in Figure 12, where the discrete-time STN model[20] is positioned in the aforementioned 3-D space.

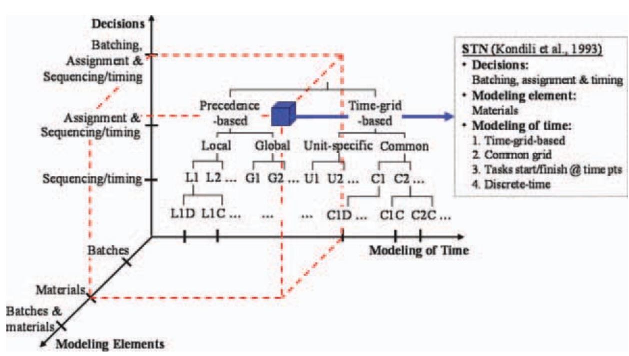

Figure 12. Graphic illustration of the classification of modeling approaches.

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

Relationship between problem classes and modeling approaches

The selection of modeling elements, optimization decisions, and modeling of time determine the modeling approach. Thus, they are attributes of the model not necessarily defined by the specific facility or problem being modeled. Given a modeling approach, the actual model should be suitable for the given production environment and account for specific process characteristics and constraints. In other words, the scheduling model is determined by the (user selected) modeling approach and the (externally specified) problem class.

The two aspects are, naturally, interrelated. Most notably, the selection of the modeling approach often depends on the problem at hand. However, a problem can be modeled using multiple modeling approaches; for example, problems in a given sequential facility with changeovers can be addressed using different types of batch-based models with different types of precedence variables and constraints, while a problem in a network facility can be addressed using a wide range of models. At the same time, a modeling approach can be used to address different problems in different types of facilities. For example, the STN model of Kondili et al.[20] was extended to account for semibatch processes[21]; RTN and STN type models can be amended to account for material transfers[54,55]; the STN-type model of Ierapetritou and Floudas[56] was expanded to account for continuous and semicontinuous processes.[57]

Conclusions

In this article, we first presented a framework for chemical production scheduling. The notion of production environment is introduced to allow us to define the various classes of problems. We identify three main types of chemical production environments: sequential, network, and hybrid. Methods to address problems in the first and second environments have been available since the late 1970s and early 1990s, respectively. However, there are no methods available for problems in hybrid environment. It is also suggested that chemical production scheduling problem classes can be defined in terms of production environment(s), processing characteristics and constraints, and objective function.

Our analysis also offered a number of novel insights:

(a) The production environment, and therefore the problem class, depends primarily on the way materials are handled, rather than on the structure of the facility (as in discrete manufacturing).

(b) Sequential processing is in some ways similar to discrete manufacturing, whereas network and hybrid processing have different requirements and therefore require different methods.

(c) There are two subtypes of sequential production environments: multistage and multipurpose. Problems in both of them can be expressed in terms of products (or batches), $i \in \mathbf{I}$, stages, $k \in \mathbf{K}$, units, $j \in \mathbf{J}$, product routings, $\mathbf{R}_i$, and unit-(product/stage) compatibility information expressed by sets $\mathbf{J}_{i,k}$.

(d) Material mixing and batch mixing are two different concepts. The former describes the consumption of multiple materials by one task; and it can be present in sequential processing. The latter describes the mixing of two or more batches of the same material (produced by the same or different tasks); and it implies network processing.

(e) Batch splitting is not equivalent to a task having more than one outputs. The former describes the situation where the material coming from the same batch is used as input in multiple downstream tasks; it implies network processing. The latter simply means that a task produces multiple materials, some of which may be subject to no splitting/mixing restrictions (i.e., sequential processing).

(f) Facilities typically consist of subsets of units/tasks of different processing types. This means that the overwhelming majority of existing methods, which were developed to address problems in a single environment, cannot address these problems. A first attempt to address problems in facilities that combine sequential and network environments was proposed by Sundaramoorthy and Maravelias.[39]

Second, we developed a systematic classification of modeling approaches, which considers three attributes: scheduling decisions, modeling elements, and modeling of time. The development of the proposed classification resulted in the following critical insights:

(a) Most approaches to sequential problems rely on the modeling of batches, thus called batch-based. In most of them it is assumed that the batching problem has already been solved, which implies that no batching decisions are made and thus discrete batches need to be assigned to units and sequenced. If it is also assumed that there are no storage and utility constraints, then these problems become similar to discrete manufacturing problems.

(b) Approaches to network problems rely on the modeling of materials through material balances, and they involve all types of decisions (batching, assignment, and timing). They are termed material-based.

(c) The modeling of time involves four levels. The selection of discrete or continuous representation is only one of them.

(d) At the first level, we decide between a precedence-based and time-grid-based approach. The former is used exclusively in batch-based approaches for problems in sequential environments, while the latter is used in both batch-based and material-based approaches.

(e) The second and third levels concern the selection of type of precedence-based (global vs. local precedence) and time-grid-based (common vs. unit specific grid) approach, and the specific assumptions.

(f) The selection between discrete- and continuous-time is the fourth level, and it is independent from the other three levels. For example, all precedence-based approaches, which have been thought to be inherently continuous-time, can become discrete by simply introducing integrality constraints on the timing variables. The same is true for any time-grid-based approach.

The major contribution of this article, we believe, is the identification of the limitations of the existing notation, the clarification of some common misconceptions, and the introduction of some new ideas (e.g., hybrid processing) that would allow us to systematize our knowledge in this area. It is also a first attempt to critically review our past efforts in developing modeling and solution methods, understand what problems they address, what are their limitations, and how they can be improved. The development of the general framework and modeling approach classification, while more general and systematic than all previous attempts, were just the means to this end. We hope that this article will foster further research in the area.

Acknowledgments

The author like to acknowledge financial support from the National Science Foundation under grant CBET-1066206. The author would also like to thank Mr Jeff Kelly, Prof. Gabriela Henning, and Dr. Danielle Zygnier for fruitful discussions in the area of scheduling.

Notation

Machine environments

- $\boldsymbol{1}$ = single machine
- $\boldsymbol{P}$ = machines in parallel
- $\boldsymbol{F}$ = flow-shop
- $\boldsymbol{J}$ = job-shop
- $\boldsymbol{O}$ = open-shop
- $\boldsymbol{FF}$ = flexible flow-shop
- $\boldsymbol{FJ}$ = flexible job-shop
- $\boldsymbol{FO}$ = flexible open-shop

Production environments

- $\boldsymbol{SS}$ = sequential multistage
- $\boldsymbol{SP}$ = sequential multipurpose
- $\boldsymbol{N}$ = network
- $\boldsymbol{H}$ = hybrid

Indices and sets

- $i \in \mathbf{I}$ = job/batch/product
- $j \in \mathbf{J}$ = machine/unit
- $k \in \mathbf{K}$ = operation(step)/stage

Subsets

- $\mathbf{I}_j$ = tasks can be executed on unit j
- $\mathbf{J}_k$ = machine/unit
- $\mathbf{J}_{i,k}$ = units suitable to process product (batch) i in stage k
- $\mathbf{R}_i$ = routing of product (batch) i

Greek letters

- α = unit/production environment
- β = processing characteristics and constraints
- γ = objective function

Literature Cited

1. Salveson ME. On a quantitative method in production planning and scheduling. *Econometrica*. 1952;20:554–590.
2. Johnson SM. Optimal two- and three-stage production schedules with setup times included. *Nav Res Logist Q*. 1954;1:61–68.
3. Pinedo M. *Scheduling : Theory, Algorithms, and Systems*, 3rd. Springer, New York; London, 2008.
4. Pinedo,M.L., *Planning and Scheduling in Manufacturing and Services*. New York: Springer, 2009.
5. Kallrath J. Planning and Scheduling in the Process Industry. *OR Spectrum* 2002;24:219–250.
6. Mendez CA, Cerda J, Grossmann IE, Harjunkoski I, Fahl M. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput Chem Eng*. 2006;30:913–946.
7. Maravelias CT, Sung C. Integration of production planning and scheduling: overview, challenges and opportunities. *Comput Chem Eng*. 2009;33:1919–1930.
8. Grossmann I. Enterprise-wide optimization: a new frontier in process systems engineering. *AIChe J*. 2005;51:1846–1857.
9. Grossmann IE, Biegler LT. Part Ii. Future perspective on optimization. *Comput Chem Eng*. 2004;28:1193–1218.
10. Pinedo M. *Scheduling: Theory, Algorithms, and Systems*. Englewood Cliffs, NJ: Prentice Hall, 1995.
11. Kelley JE. Critical-path planning and scheduling: mathematical basis. *Oper Res*. 1961;9:296–320.
12. Reklaitis GV. Review of scheduling of process operations. *AIChE Symp. Ser*. 1982;78:119–133.

13. Mauderli A, Rippin DWT. Production planning and scheduling for multipurpose batch chemical-plants. *Comput Chem Eng*. 1979;3:199–206.
14. Mauderli A, Rippin DWT. Scheduling production in multipurpose batch plants - the batchman program. *Chem Eng Prog*. 1980;76:37–45.
15. Egli UM, Rippin DWT. Short-term scheduling for multiproduct batch chemical plants. *Comput Chem Eng*. 1986;10:303–325.
16. Wellons MC, Reklaitis GV. Optimal schedule generation for a single-product production line .1. Problem formulation. *Comput Chem Eng*. 1989;13:201–212.
17. Ku HM, Karimi IA. Scheduling in serial multiproduct batch processes with finite interstage storage - a mixed integer linear program formulation. *Ind Eng Chem Res*. 1988;27:1840–1848.
18. Cerda J, Vicente M, Gutierrez JM, Esplugas S, Mata J. A New methodology for the optimal design and production schedule of multipurpose batch plants. *Ind Eng Chem Res*. 1989;28:988–998.
19. Vaselenak JA, Grossmann IE, Westerberg AW. An embedding formulation for the optimal scheduling and design of multipurpose batch plants. *Ind Eng Chem Res*. 1987;26:139–148.
20. Kondili E, Pantelides CC, Sargent RWH. A general algorithm for short-term scheduling of batch-operations. 1. Milp formulation. *Comput. Chem Eng*. 1993;17:211–227.
21. Shah N, Pantelides CC, Sargent RWH. A general algorithm for short-term scheduling of batch-operations. 2. Computational issues. *Comput Chem Eng*. 1993;17:229–244.
22. Pantelides CC.Unified frameworks for the optimal process planning and scheduling. *Second Conference on Foundations of Computer Aided Operations*, 12, 1994.
23. Subrahmanyam S, Bassett MH, Pekny JF, Reklaitis GV. Issues in solving large-scale planning, design and scheduling problems in batch chemical plants. *Comput Chem Eng*. 1995;19:S577–S582.
24. Shobrys DE, White DC. Planning, scheduling and control systems: why can they not work together. *Comput Chem Eng*. 2000;24:163–173.
25. Harjunkoski I, Nystrom R, Horch A. Integration of scheduling and control-theory or practice? *Comput Chem Eng*. 2009;33:1909–1918.
26. Munoz E, Capon-Garcia E, Moreno-Benito M, Espuna A, Puigjaner L. Scheduling and control decision-making under an integrated information environment. *Comput Chem Eng*. 2011;35:774–786.
27. Birewar DB, Grossmann IE. Incorporating scheduling in the optimal design of multuproduct batch plants. *Comput Chem Eng*. 1989;13:141–161.
28. Pinto JM, Grossmann IE. Assignment and sequencing models for the scheduling of process systems. *Ann Oper Res*. 1998;81:433–466.
29. Castro PM, Grossmann IE, Novais AQ. Two new continuous-time models for the scheduling of multistage batch plants with sequence dependent changeovers. *Ind Eng Chem Res*. 2006;45:6210–6226.
30. Shaik MA, Floudas CA. Unit-specific event-based continuous-time approach for short-term scheduling of batch plants using rtn framework. *Comput Chem Eng*. 2008;32:260–274.
31. Sundaramoorthy A, Maravelias CT. Modeling of storage in batching and scheduling of multistage processes. *Ind Eng Chem Res*. 2008;47:6648–6660.
32. Sundaramoorthy A, Maravelias CT. Simultaneous batching and scheduling in multistage multiproduct processes. *Ind Eng Chem Res*. 2008;47:1546–1555.
33. Baker KR. *Introduction to Sequencing and Scheduling*. New York: Wiley, 1974.
34. French S. *Sequencing and Scheduling : An Introduction to the Mathematics of the Job-Shop*. Horwood E; New York: Wiley, Chichester, 1982.
35. Stadtler H. Supply chain management and advanced planning - basics, overview and challenges. *Eur J Oper Res*. 2005;163:575–588.
36. Papageorgiou LG. Supply chain optimisation for the process industries: advances and opportunities. *Comput Chem Eng*. 2009;33:1931–1938.
37. Pochet Y, Wolsey LA. *Production Planning by Mixed Integer Programming*. New York; Berlin: Springer, 2006.
38. Novas JM, Henning GP. Reactive scheduling framework based on domain knowledge and constraint programming. *Comput Chem Eng*. 2010;34:2129–2148.
39. Sundaramoorthy A, Maravelias CT. A general framework for process scheduling. *AIChe J*. 2011;57:695–710.
40. Sundaramoorthy A, Maravelias CT, Prasad P. Scheduling of multistage batch processes under utility constraints. *Ind Eng Chem Res*. 2009;48:6050–6058.
41. Kelly JD, Mann JL. Crude oil blend scheduling optimization: an application with multimillion dollar benefits. 2. The ability to schedule the crude oil blendshop more effectively provides substantial downstream benefits. *Hydrocarbon Process*. 2003;82: 72–79.
42. Terrazas-Moreno S, Flores-Tlacuahuac A, Grossmann IE. Simultaneous cyclic scheduling and optimal control of polymerization reactors. *AIChe J*. 2007;53:2301–2315.
43. Gimenez DM, Henning GP, Maravelias CT. A novel network-based continuous-time representation for process scheduling. II. General framework. *Comput Chem Eng*. 2009;33:1644–1660.
44. Prasad P, Maravelias CT, Kelly J. Optimization of aluminum smelter casthouse operations. *Ind Eng Chem Res*. 2006;45:7603–7617.
45. Hai R, TheiÃŸen M, Marquardt W. An ontology based approach for operational process modeling. *Adv Eng Inform*. 2011;25:748–759.
46. Prasad P, Maravelias CT. Batch selection, assignment and sequencing in multi-stage multi-product processes. *Comput Chem Eng*. 2008;32:1106–1119.
47. Ferris M, Maravelias CT, Sundaramoorthy A. Simultaneous batching and scheduling using dynamic decomposition on a grid. *INFORMS J Comput*. 2009;21:398–410.
48. Sung C, Maravelias CT. A mixed-integer programming formulation for the general capacitated lot-sizing problem. *Comput Chem Eng*. 2008;32:244–259.
49. Liu SS, Pinto JM, Papageorgiou LG. Single-stage scheduling of multiproduct batch plants: an edible-oil deodorizer case study. *Ind Eng Chem Res*. 2010;49:8657–8669.
50. Kopanos GM, Puigjaner L, Maravelias CT. Production planning and scheduling of parallel continuous processes with product families. *Ind Eng Chem Res*. 2011;50:1369–1378.
51. Castro PM, Erdirik-Dogan M, Grossmann IE. Simultaneous batching and scheduling of single stage batch plants with parallel units. *AIChe J*. 2008;54:183–193.
52. Maravelias CT. Mixed-time representation for state-task network models. *Ind Eng Chem Res*. 2005;44:9129–9145.
53. Maravelias CT, Grossmann IE. On the relation of continuous- and discrete-time state-task network formulations. *AIChe J*. 2006;52:843–849.
54. Barbosa-Povoa APFD, Pantelides CC. Design of multipurpose production facilities: a rtn decomposition-based algorithm. *Comput Chem Eng* 1999;23:S7–S10.
55. Gimenez DM, Henning GP, Maravelias CT. A Novel network-based continuous-time representation for process scheduling. I. Main concepts and mathematical formulation. *Comput Chem Eng*. 2009;33:1511–1528.
56. Ierapetritou MG, Floudas CA. Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. *Ind Eng Chem Res*. 1998;37:4341–4359.
57. Ierapetritou MG, Floudas CA. Effective continuous-time formulation for short-term scheduling. 2. Continuous and semicontinuous processes. *Ind Eng Chem Res*. 1998;37:4360–4374.

Appendix A: Facility Structure vs. Type of Processing

To illustrate that the facility does not necessarily determine the type of processing, we revisit the example shown in Figure 5b. The facility at hand consists of one reactor, U0, in the first stage (for task T1), and reactors U1, U2, and U3, in the second stage (dedicated to tasks, T1, T2, and T3, respectively) as shown in Figure A1. The facility has also dedicated storage vessels, for feedstock F, final products, P1, P2, and P3, as well as vessel IV for intermediate I. Reactor U1 is connected to vessel IV through valve V0, and IV is connected to all second stage reactors via valves V1, V2, and V3, as shown in Figure A1.

If two or more batches of I cannot be mixed into IV (no mixing), and each batch of I has to be consumed by a single second stage task (no splitting), then the operation of the facility should essentially follow two rules: (i) open valve V0 only if tank IV is empty; and (ii) only one valve, among V1, V2, and V3, can be used to discharge IV, and when opened it has to completely empty IV. The first rule enforces the no mixing restriction, whereas the second enforces the no splitting restriction. A schedule satisfying these two rules is shown in Figure A2a.

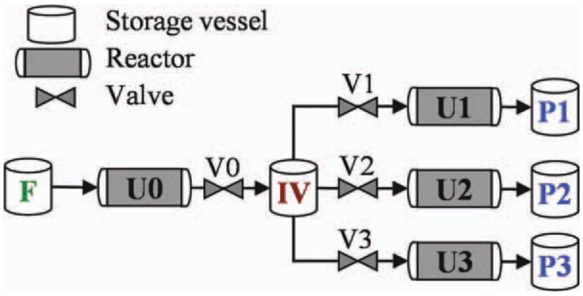

Figure A1. Processing units, storage vessels, and interconnections.

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

However, exactly the same facility can be used for other types of processing. For example, if batch mixing is allowed for I, but not batch splitting, then every feasible schedule should follow rule (ii) only (see schedule in Figure A2b). If batch mixing is not allowed, but a batch of I can be used in multiple downstream tasks, then only rule (i) should be followed (see schedules in Figures A2c,d). If none of the rules is followed, the same facility can be used to carry out network processing (see schedule in Figure A2e).

Finally, note that in the schedules shown in Figures A2b–e, tank IV can be viewed as a mixer or splitter. Despite the presence of this unit, however, different types of processing, including sequential processing, can be carried out in this facility.

Appendix B: Remarks on Existing Notation

The reader might have noticed that we used the terms *batch* and *task* almost interchangeably, although they were used in different context. This somewhat confusing use of these two terms, as well as many others, are due to the lack of a common notation for chemical production scheduling, which often leads to either multiple terms being used for the same concept or the same term being used to describe different concepts. In general, there are mismatches between: (i) the notation used in different (mostly academic) modeling approaches, and (ii) the notation used in modeling approaches and the standards used to develop software products (and used by other communities). Our discussion focuses on the first type of mismatch.

As we have discussed, existing modeling approaches can be classified into two general groups, following the dichotomy between sequential and network processes: (i) batch-based approaches, developed to address problems in the former, and (ii) material-based approaches, developed to address problems in the latter. As we discussed in "Scheduling Decisions and Modeling Elements," most batch-based approaches assume that batching decisions are made independently from assignment and sequencing decisions, which implies that the number of batches as well as their batchsizes and processing times are fixed. Thus, it is assumed that problems are defined in terms of: (i) the processing stages, (ii) the parallel units comprising each stage, and (iii) the (fixed number of) batches to be processed. Whereas multipurpose facilities may require some additional information, the main elements are the same. The major implication of this structure is that the schedule concerns the assignment of batches to units and the sequencing of batches. Conversely, material-based approaches are based on the notions of (i) tasks and (ii) materials (modeled either are states in RTN or resources in RTN). Tasks play a key role because they trigger all changes in inventory levels (through material consumption and production), and resource utilization levels.

Based on this analysis we note the following. In batch-based approaches:

(a) Tasks are not explicitly defined (recall that a task in scheduling is an activity to be scheduled.). A task can be indirectly defined as the processing of a batch at (a unit of) a given stage.

(b) Batches are defined as the entities that move intact through the various processing stages (equivalent to jobs in the OR problems). Hence, one batch is associated with multiple tasks.

Also, the terms batch and order are often used inconsistently. In some papers, the two terms are used interchangeably, assuming that each order always comprises a single batch. In other papers, orders are treated as input data (e.g., orders placed by customers with amounts and due dates), while batches are the result of the scheduling exercise. The decision-maker converts orders into batches – in some cases an order may correspond to one batch, but in the general case an order can be satisfied by multiple batches and/or a batch can be used to satisfy multiple orders.

In material-based approaches:

(a) The term ***task*** is used but in a manner which is inconsistent with traditional scheduling notation (e.g., in ***FF***, the processing of a job in an operation is a task; if we have to schedule five jobs in three operations, then there are 15 tasks to be scheduled). In most material-based approaches a task is a type of processing activity, which means that a task may be executed multiple times.

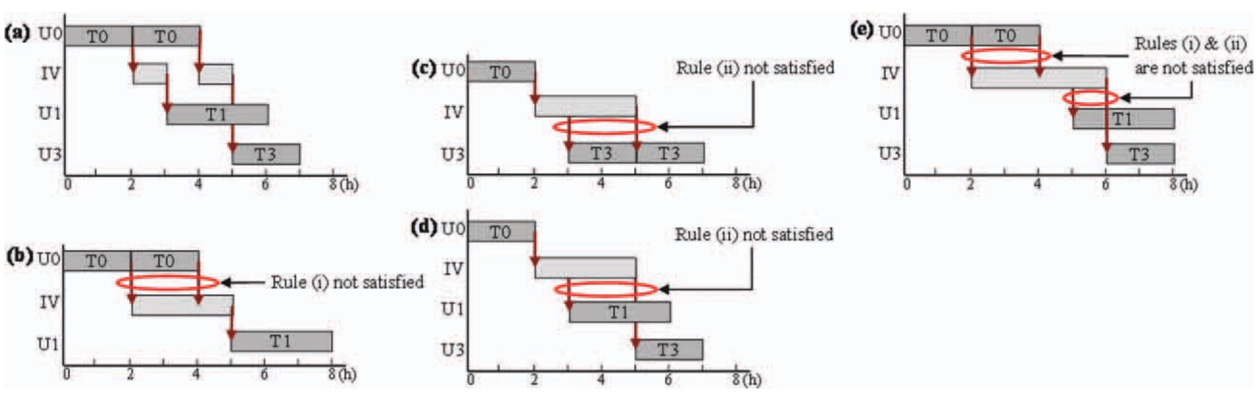

Figure A2. Different ways to operate the same facility.

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

(b) Each execution of a task i, is often referred to as a ***batch*** of task i, which is obviously a very different use of the term batch when compared to sequential processes.

In summary, the term task is not used in batch-based approaches (although it can be defined as a combination of a batch and a stage) while its use in material-based approaches is inconsistent with the traditional use of the term. The term batch is used to describe two different concepts.

The mismatches between academic approaches and standards are equally striking. For example, in ISA-88, a standard originally developed for batch process control, the terms procedure, unit procedure, operation, and phase are used to describe different levels of detail of the same task. In academic approaches, this distinction is not made and thus these terms cannot be represented in the models that have been developed.

Manuscript received Nov. 17, 2011, and revision received Feb. 14, 2012.