6
Batch and Hybrid Processes

Luis Puigjaner and Javier Romero

6.1
Introduction

Although historically, chemical engineers achieved their professional distinction with the design and operation of continuous processes [1], as we move into the new millennium, it comes somewhat as a surprise to realize that outside the petroleum and petrochemical industries, batch operation is still a common if not dominant mode of operation. Moreover, most batch processes are unlikely to be replaced by continuous processes [2, 3].

The reason is that as the production of chemicals undergoes a continuous specialization, to address the diversifying needs of the marketplace, the continuous evolution of product recipes implies a much shorter life cycle for a growing number of chemicals, than has been traditionally the case, leading to a perpetual product/process evolution [4, 5].

This situation has been matched and in part funded by a developing research interest in batch process systems engineering, batch production being the most suitable way of manufacturing the relatively large number of low-volume high-value-added products commonly found in the fine and specialty chemicals industry. Moreover, the coexistence of continuous and discrete parts in both strictly speaking batch processes and nominal continuous processes has motivated an increased research interest in further exploiting the inherent flexibility of batch procedures and the high productivity of continuous parts of the production system [6]. Thus, chemical plants constitute large hybrid systems, making it necessary to consider the continuous-discrete interactions taking place within an appropriate framework for plant and process simulation and optimization [7].

This chapter briefly discusses existing modeling frameworks for discrete/hybrid production systems embodying different approaches, before introducing a very recent framework for process recipe initialization that integrates a recipe model into the batch plant-wide model. Next, online and offline recipe adaptation from real-time plant information is presented, and finally, a model-based integrated advisory system

Computer Aided Process and Product Engineering. Edited by Luis Puigjaner and Georges Heyen
Copyright © 2006 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 3-527-30804-0

is described. This system gives online advice to operators on how to react in case of process disturbances. In this way, an enhanced overall process flexibility and productivity is achieved. Application of this promising approach is illustrated through examples of increasing complexity.

6.1.1
Plant and Process Simulation

The discrete transitions occurring in chemical processing plants have only recently been addressed in a systematic manner. Barton and Pantelides [8] did pioneering work in this area. A new formal mathematical description of the combined discrete/continuous simulation problem was introduced to enhance the understanding of the fundamental discrete changes required to model processing systems. The modeling task is decomposed into two distinct activities: modeling fundamental physical behavior, and modeling the external actions imposed on this physical system resulting from interaction of the process with its environment by disturbances, operation procedures, or other control actions.

The physical behavior of the system can be described in terms of a set of integral and partial differential and algebraic equations (IPDAE). These equations may be continuous or discontinuous. In the latter case, the discontinuous equations are modeled using state-task networks (STNs) and resource-task networks (RTNs), which are based on discrete models. Otherwise, other frameworks based on a continuous representation of time have appeared more recently (event operation network among others). The detailed description of the different representation frameworks is the topic of the next section.

6.1.2
Process Representation Frameworks

The representation of a state-task network (STN) proposed by Kondili et al. [9] was originally intended to describe complex chemical processes arising in multiproduct/multipurpose batch chemical plants. The established representation is similar to the flow sheet representation of continuous plants, but is intended to describe the process itself rather than a specific plant.

The distinctive characteristic of the STN is that it has two types of nodes; mainly, the *state* nodes, representing the feeds, intermediates and final products and the *task* nodes, representing the processing operations which transform material from input states to output states (Fig. 6.1).

This representation is free from the ambiguities associated with recipe networks where only processing operations are represented. Process equipment and its connectivity are not explicitly shown. Other available resources are not represented.

The STN representation is equally suitable for networks of all types of processing tasks, continuous, semicontinuous or batch. The rules followed in its construction are:

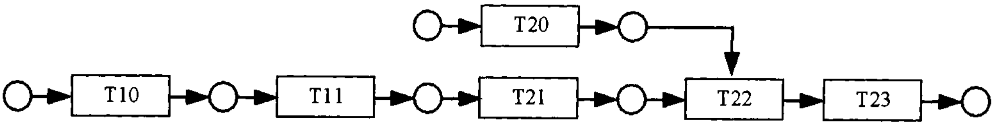


Figure 6.1 State-task network representation of chemical processes. *Circles*: state nodes; *rectangles*: task nodes

- A task has as many input (output) states as different types of input (output) material.
- Two or more streams entering the same state are necessarily of the same material. If mixing of different streams is involved in the process, then this operation should form a separate task.

The STN representation assumes that an operation consumes material from input states at a fixed ratio and produces material for the output state also at a known fixed proportion. The processing time of each operation is known *a priori* and considered to be independent of the amount of material to be processed. Otherwise, the same operation may lead to different states (products) using different processing times.

States may be associated to four main types of storage policy:

- unlimited intermediate storage
- finite intermediate storage
- no intermediate storage
- zero wait (the product is unstable).

An alternative representation; the *resource-task network* (RTN) was proposed by Pantelides [10]. In contrast to the STN approach, where a task consumes and produces materials while using equipment and utilities during its execution, in this representation, a task is assumed only to consume and produce resources. Processing items are treated as though consumed at the start of a task and produced at the end. Furthermore, processing equipment in different conditions can be treated as different resources, with different activities consuming and generating them; this enables a simple representation of changeover activities. Pantelides [10] also proposed a discrete-time scheduling formulation based on the RTN, which, due to the uniform treatment of resources, only requires the description of three types of constraint, and does not distinguish between identical equipment items. He demonstrated that the integrality gap could not be worse than the most efficient form of STN formulation, but the ability to capture additional problem features in a straightforward fashion is attractive. Subsequent research has shown that these conveniences in formulation are overshadowed by the advantages offered by the STN formulation in allowing explicit exploitation of constraint structure through algorithm engineering.

The STN and RTN representations use discrete-time models. Such models suffer from a number of inherent drawbacks [11]:

- The discretization interval must be fine enough to capture all significant events, which may result in a very large model.

- It is difficult to model operations where the processing time is dependent on the batch size.
- The modeling of continuous operations must be approximated and minimum run-lengths give rise to complicated constraints.

Therefore, attempts have been made to develop frameworks based on a continuous-time representation. Reklaitis and Mockus [12] developed a continuous-time formulation based on the STN representation. A common resource grid is need, with the timing of the grid points ("event orders" in their terminology) determined by optimization. The same authors introduced an alternative solution procedure based on Bayesian heuristics in a later work [13]. Zhang and Sargent [14] describe a continuous-time representation based on RNT representation for both batch and continuous operations. The poor relaxation performance of the continuous-time models is the main obstacle to their large scale application. To avoid this deficiency, Shilling and Pantelides [11] modify the model by Zhang and Sargent (1996). A global linearization gives rise to a mixed-integer linear programming (MILP) which is solved by a hybrid branch-and-bound procedure. Recent reviews on these approaches can be found in Shah [15] and Silver et al. [16].

A realistic and flexible description of complex recipes has been recently improved using a flexible modeling environment [17] for the scheduling of batch chemical processes. The process structure (individual tasks, entire subtrains or complex structures of manufacturing activities) and related materials (raw, intermediate or final products) is characterized by means of a processing network which describes the material balance. In the most general case, the activity carried out in each process constitutes a general activity network. Manufacturing activities are considered at three different levels of abstraction: the process level, the stage level and the operation level.

This hierarchical approach permits the consideration of material states (subject to material balance and precedence constraints) and temporal states (subject to time constraints) at different levels.

At the process level, the process and materials network (PMN) provides a general description of production structures (such as synthesis and separation processes) and materials involved, including intermediates and recycled materials. An explicit material balance is specified for each of the processes in terms of a stoichiometric-like equation relating raw materials, intermediates and final products (Fig. 6.2). Each process may represent any kind of activity necessary to transform the input materials into the derived outputs.

Between the process level and the detailed description of the activities involved at the operation level, there is the stage level. At this level, the block of operations to be executed in the same equipment is described. Hence, at the stage level each process is split into a set of the blocks (Fig. 6.3). Each stage implies the following constraints:

- The sequence of operations involved requires a set of implicit constraints (links).
- Unit assignment is defined at this level. Thus, for all the operations of the same stage, the same unit assignment must be made.

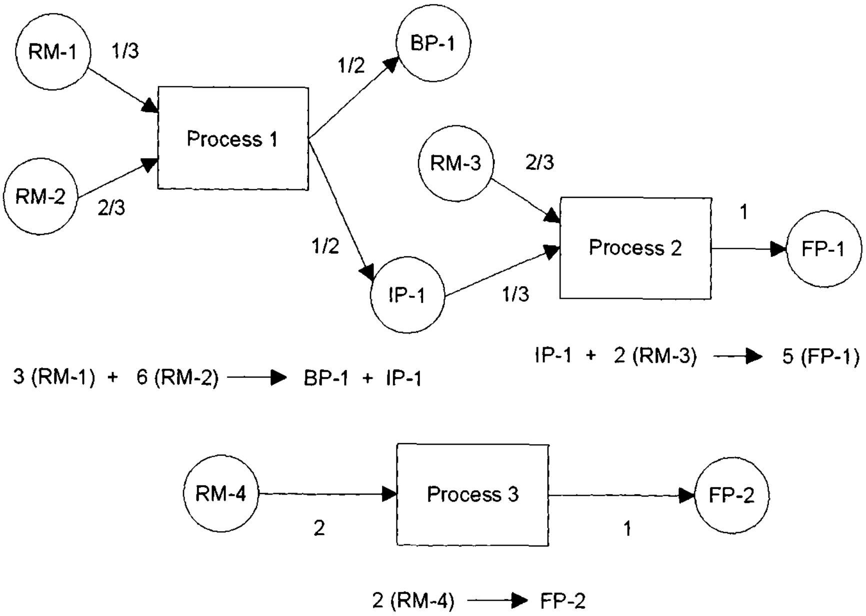


Figure 6.2 A process and materials network (PMN) describing the processing of two products. RM are row materials, IP are intermediate products, BP are by-products and FP are final products

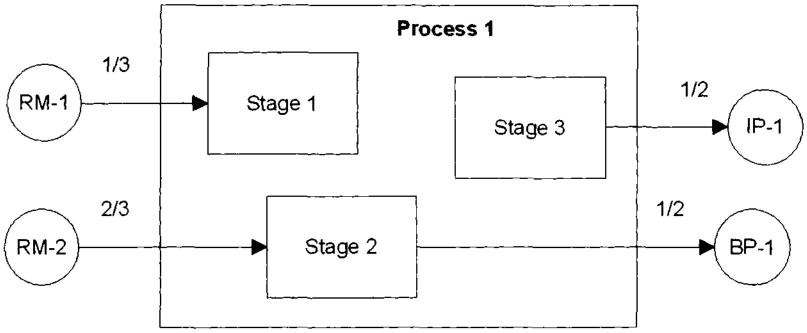


Figure 6.3 Stage level. Each stage involves different unit assignment opportunities

- A common size factor is attributed to each stage. This size factor summarizes the contribution of all the operations involved.

The operation level contains the detailed description of the activities contemplated in the network (tasks and subtasks), while implicit time constraints (links) must be also met at this level. The detailed representation of the structure of activities defining the different processes is called the event operation network (EON). It is also at this level that the general utility requirements (renewable, nonrenewable, storage) are represented.

The event operation network representation model describes the appropriate timing of process operations. A continuous-time representation of process activities is made using three basic elements: events, operations and links [18, 19].

Events designate those time instants where some change occurs. They are represented by nodes in the EON graph, and may be linked to operations or other events. Each event is associated to a time value and a lower bound.

Operations comprise those time intervals between events (Fig. 6.4). Each operation *m* is represented by a box linked with solid arrows to its associated nodes: initial *NI m* and final *NF m* nodes. Operations establish the equality links between nodes (two) in terms of the characteristic properties of each operation: the operation time, *TOP* and the waiting time *TW*. The operation time will depend on the amount of materials to be processed; the unit model and product changeover. The waiting time is the lag time between operations, which is bounded.

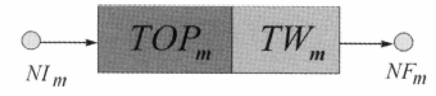


Figure 6.4 The time description for operations. *TOP* Operation time, *TW* waiting time, *NI m* initial node of operation *m*, *NF m* final node of operation *m*

Finally, links are established between events by precedence constraints. A dashed arrow represents each link K from its node of origin NO_k to its destiny node ND_k and an associated offset time ΔT_K.

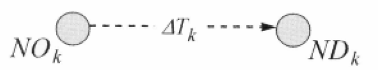


Figure 6.5 Event to event link and associated offset time representation. The *dashed arrow* represents each link K from its node of origin NO_k to its destiny node ND_k

Despite its simplicity, the EON representation is very general and flexible and it allows the handling of complex recipes (Fig. 6.6). The corresponding TOP, according to the batch size and material flow rate, also represents transfer operations between production stages. The necessary time overlapping of semicontinuous operations with batch units is also contemplated in this representation through appropriate links.

Other resources required for each operation (utilities, storage, capacity, manpower, etc.) can also be considered associated to the respective operation and timing.

Simulation of plant operation can be performed in terms of the EON representation from the following information contained in the process recipe and production structure characteristics:

- A sequence of production runs or jobs associated to a process or recipe.
- A set of assignments associated to each job and consistent with the process *p*.
- A batch size associated to each job and consistent with the process.
- A set of shifting times for all the operations involved.

These decisions may be generated automatically by using diverse procedures for the determination of an initial feasible solution. Hence, simulation may be executed by

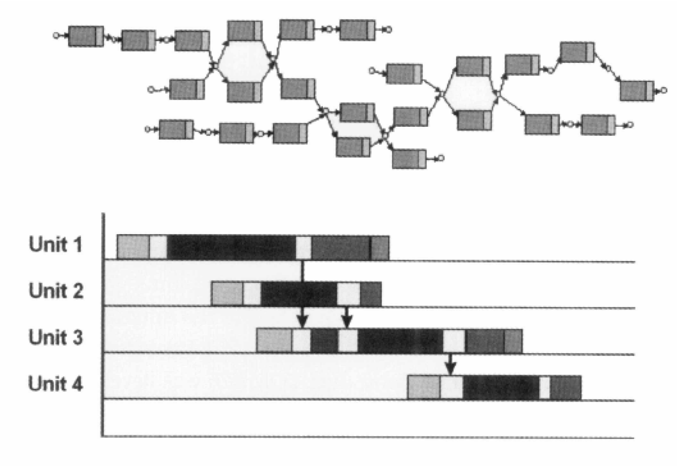


Figure 6.6 The recipe described as a structured set of operations. The event operation network (EON) representation allows the handling of complex synthesis problems. The corresponding typical Gantt chart is given below

solving the corresponding EON to determine the timing of the operations and other resources requirements.

The flexibility and potential of the EON representation has been further exploited by incorporating the flexible recipe concept, which is the subject of the next section.

6.2
The Flexible Recipe Concept

The simulation environments described in the previous section assume operating at nominal conditions following fixed recipes. Moreover, these nominal conditions are determined only once and sometimes considering only one stage of the process recipe. However, batch and hybrid manufacturing systems' optimum performance require an integrated modeling environment capable of incorporating systematic information and of adapting to changing plant scenarios. Very recently, the flexible recipe concept has been introduced as an appropriate mechanism that permits the simultaneous optimization of recipe and plant operation [20, 21].

This concept arose from the fact that batch processes normally do not operate at the plant-wide optimal nominal conditions of the fixed batch recipes, but the traditional fixed recipe does not allow for adjustment to plant resource availability or to variations in both quality of raw materials and in the actual process conditions. However, the industrial process is often subject to various disturbances and to constrained plant resources availability. Therefore, the fixed recipe is in practice approximately adapted, but in a rather unsystematic way depending on the experience and intuition of operators. As an alternative, the concept of flexible recipe operation is introduced, and a general framework is presented to systematically deal with the required adaptations at a plant-wide level.

The flexible recipe concept was considered for the first time in the context of evolutionary operation [22]. The main objective of that approach was to gain statistical insight into the problem behavior in order to gradually improve process efficiency through suggestions of minor recipe modifications in each batch-run. However, it was not until the work of Rijnsdorp appeared [20] that the concept of flexible recipes was adequately introduced. Here, the term recipe is understood in a more abstract way as referring to the selected set of adjustable elements that control the process output generating the flexible recipe. According to this concept, a flexible recipe philosophy to operate batch processes was described in the work of Verwater-Lukszo [21]. This philosophy distinguishes two main levels in the flexible recipe: (a) the recipe initialization level, where different aspects of a master flexible recipe are adjusted to actual process conditions and availability of resources at the beginning of the batch, thus giving the initialized control recipe; and (b) the recipe correction level, where the initialized control recipe is adjusted to run-time process deviations, thus generating corrected control recipes. A flexible recipe improvement system tool (called COMBO) was developed by TNO TPD (Netherlands Organization for Applied Scientific Research) for application of the flexible recipes concept in industrial practice.

However, in this approach, only one critical stage of the process is considered and hence, no interaction with plant-wide optimization is, in fact, attempted. More recently, the application of the flexible recipe concept to an entire batch train was attempted for the multistage case [23]. However, standard quality models were assumed for process operations, and hence, no insight into recipe behavior was obtained. A new framework for recipe initialization that integrates a recipe model into the batch plant-wide model has been recently introduced [24]. The aim of this approach is to optimize the entire batch process, from recipe set-point adjustment to product sequencing. For this purpose, a recipe model and a plant-wide production model are required to build the flexible recipe model. Moreover, fulfillment of present standards (ISA S88) should be a requirement for implementation in industrial practice.

6.2.1

The Flexible Recipe and the Framework of ISA S88

Batch-processes flexibility may be mainly exploited at the level of the recipe formulation. Here, the set of process parameters is adjusted to warrant process outputs as a function of uncertain process inputs. Each one of such parameters, whose value may be changed for each batch, is called a recipe item. These items can be quantitative or qualitative, time-dependent or time-independent. The equipment requirement level, as defined in ISA-S88 [25], is already a flexible category in itself. In fact, ISA-S88 defines this level as an equipment choice constraint. Finally, considering flexibility in the recipe procedure would only be contemplated when some unexpected event happens, which is out of the scope of the batch process flexibility enhancement sought here.

In a company four types of recipes are typically found:

- General recipe and site recipe; which basically describe the technique and are equipment independent.
- Master recipe, a recipe which is equipment-dependent and which provides specific and unique batch-execution information describing how a product is to be produced in a given set of process equipment.

- Control recipe, which starting as a copy of the master recipe, contains detailed information for minute-to-minute process operation of a single batch.

The flexible recipe might be derived from a master recipe and subsequently used for generating and updating a control recipe. Verwater and Keesman [26] introduced the concept of different levels between these two stages defined at ISA-S88. With these new levels a better description of the different possible functionalities of the flexible recipe is obtained:

- Master control recipe, that is, a master recipe valid for a number of batches, but adjusted to the actual conditions (actual prices or quality requirements) from which the individual control recipes per batch are derived.
- Initialized control recipe, that is, the adjustment of the still-adjustable process conditions of a master control recipe to the actual process conditions at the beginning of the batch, i.e., the adjustment of variables such as temperature, pressure, catalyst addition and processing time in the face of deviations in the initial temperature of the batch, equipment fouling, available processing time and so on.
- Corrected control recipe, the result of adjusting the initialized control recipe to process deviations during the batch.
- And finally, for monitoring and archiving purposes, it is also useful to define the accomplished control recipe.

Therefore, on the basis of this basic philosophy, a novel flexible recipe approach [24] has been recently proposed that excerpts a flexible recipe model from a total master control recipe. This model describes the whole batch process train. However, it is only concerned with the critical batch process variables. Besides, it also considers the possible interactions between different batches because of scheduling purposes.

Regarding the different levels between the master recipe and the initialized control recipe described, it can be concluded that four different flexible-recipe systems may be useful:

- A system for adjusting the master recipe to the actual prices and quality requirements, defining the master control recipe.
- A system for defining the initialized control recipe from the master control recipe as a function of the actual process conditions, availability of resources at the beginning of the batch and of the availability of the plant equipment.
- A model to generate the corrected control recipe in the face of deviations during each batch.
- A system for updating and improving the master control recipe as the database of accomplished control recipes increases. This model will also improve the preceding models.

The interaction of these systems in a real-plant environment is described in Fig. 6.7.

These systems will have to be developed in laboratory experiments, pilot plant operation, during normal production by a systematic introduction of acceptable small changes in certain inputs and parameters, or by adjusting white models and simulating them under different operating conditions.

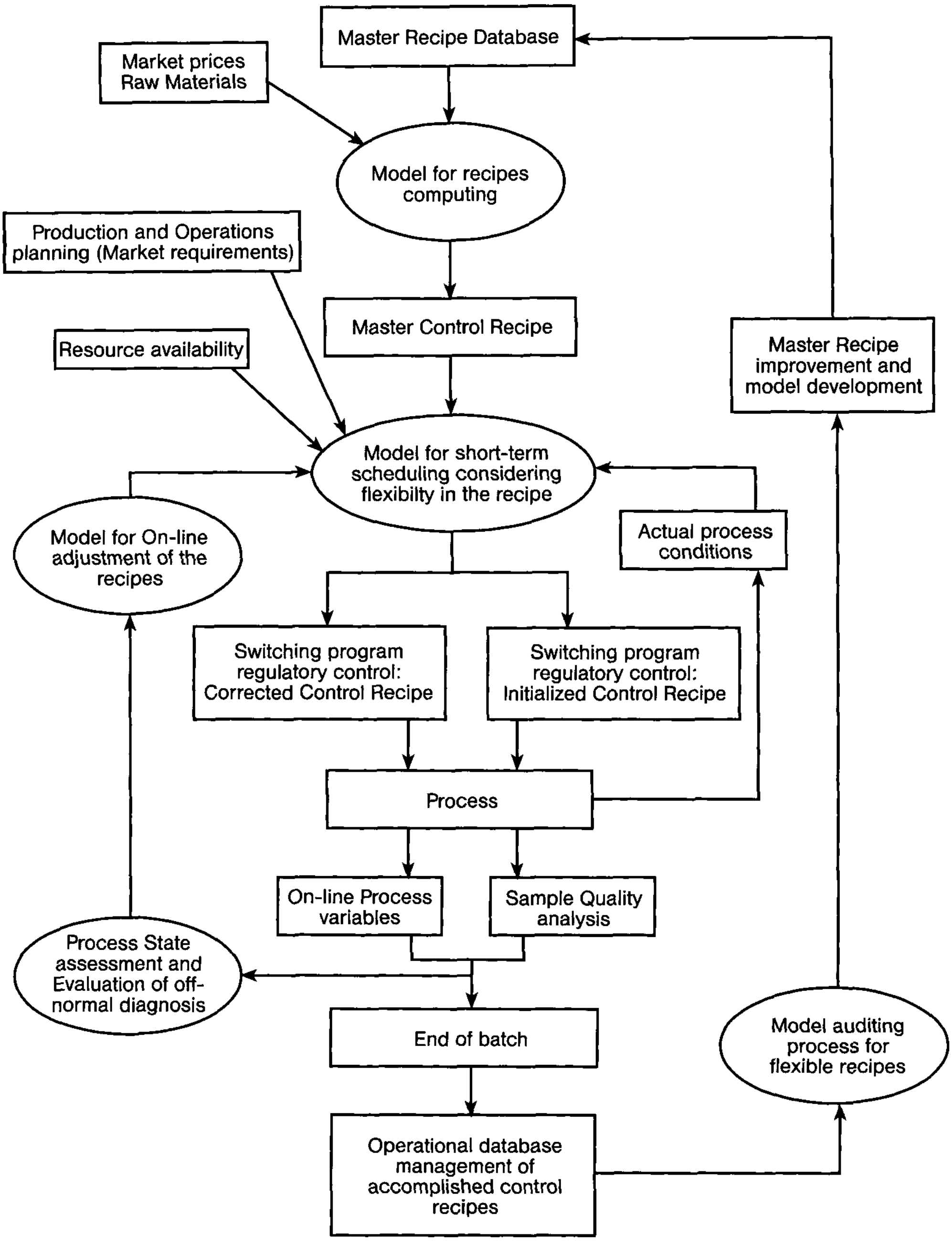


Figure 6.7 Optimal flexible recipe environment information flow proposed

6.3
The Flexible Recipe Model

The flexible recipe model is the tool that permits us to integrate a recipe optimization procedure with a batch plant optimization level. It represents the relationship that correlates a batch process output as a function of the selected input items of the recipes for different batch plant production scenarios. Therefore, it is a recipe description model that incorporates plant-wide variables. We identify four main components of the problem: quality or product specifications, process operating conditions, production costs and production due-dates.

The flexible recipe model can be applied to a variety of scenarios. For instance, during batch process operation, processing times of some tasks may vary without set-point adjustment, thus affecting the properties (quality) of the products obtained in such tasks. Then, to meet customer requirements, another batch of the same product might be able to compensate for these effects. For example, let's assume a process in which A is converted into B; one batch with low conversion of A could be compensated for by another batch with a higher conversion, assuming that these two batches are going to be mixed afterwards, so that the final product quality corresponds to the customer and legal requirements. Otherwise, the processing time might be optimized without set-point adjustment by compensating for the quality within the same batch. For instance, a batch of product A that is first heated in one piece of equipment before reacting in another. A reduction in the processing time of the first task could be offset by a higher reaction time. Moreover, the processing time could be optimized with some set-point adjustment. In this situation, the properties of intermediates produced might be altered only at the expense of a higher operation cost. For instance, the reaction time could be reduced by increasing the reaction temperature, although this recipe modification would imply a higher operation cost. Which of the above-mentioned strategies should be applied in each case will depend on the specific process and on the available knowledge of the different tasks of the process. For example, such ways of operation might not be very suitable for highly restrictive processes, such as those found in the pharmaceutical industry, but they are probably convenient to specialty batch chemical production where customer requirements are defined simply by a set of product properties and not by the specific way the product has been produced.

The preceding discussion leads to the basic concept upon which the modeling of scheduling problems considering the flexible recipe is built.

6.3.1
Proposed Concept for the Flexible Recipe Model

The flexible recipe model is regarded as a constraint on quality requirements and on production costs.

In this approach, recipe items are classified into four groups:

- The vector of process operating conditions, $\boldsymbol{poc}_i$, of stages i of a recipe. It includes parameters like temperature, pressure, type of catalyst, batch size, etc.

- The product specification vector, $\boldsymbol{ps}_i$, at the end of each process stage i of a recipe. It might include parameters like conversion of a reactant, purity or quality aspects.
- Processing time, PT_i, at each stage i of a recipe.
- Waiting time, TW_i, that is, the time between the end of a stage and the next stage start time.

Then, the product specifications vector of a batch stage will in general be a function, Ψ, of processing time, waiting time, process set-points, and product specifications at different stages i^* where the different inputs to stage i are produced. Moreover, within this model, product specifications, $\boldsymbol{ps}$, and process operation conditions, $\boldsymbol{poc}$, are subject to optimization within a flexibility region, $\boldsymbol{\alpha}$ and $\boldsymbol{\Delta}$ respectively.

A general algorithm representation of the flexible recipe model for short term scheduling is presented in Eq. (1). This model contains the nominal recipe and its capacity to accept modifications. The model adjusts the different recipe parameters for each individual batch performed in a specific production plan $\boldsymbol{\Theta}$, where $\boldsymbol{\Theta}$ is the variable that permits integrating batch process scheduling with the recipe optimization procedure.

Each specific production plan $\boldsymbol{\Theta}$ is defined when the specific orders to be delivered at a specific set of due dates, S, is specified and when the specific set of different plant resources, A, is assigned to each order. Besides this, each production plan has to meet some physical plant constraints, T, such as the multistage flowshop or job-shop batch plant topology constraints, T, operating with a set J of equipment units and a set R of process resources. Each production plan will be generated to meet the market constraints: set I of production orders in a given set DD of time horizon or due dates.

A performance criterion ϕ is also included. This criterion may vary from batch to batch and it may contain economic as well as process variables. The flexible recipe model validity constraints are considered in $\boldsymbol{\sigma}$ and $\boldsymbol{\Delta}$ regions.

$$\begin{aligned}
&\text{Optimize } \Phi(PT_i, TW_i, \boldsymbol{ps}_i, \boldsymbol{poc}_i, \Theta)\\
&\quad\text{subject to recipe constraints,}\\
&\qquad \boldsymbol{ps}_i = \Psi(PT_i, TW_i, ps_i, \boldsymbol{poc}_i),\\
&\qquad \boldsymbol{ps}_i \subset \sigma,\\
&\qquad \boldsymbol{poc}_i \subset \Delta,\\
&\quad\text{subject to production environment constraints,}\\
&\qquad \Theta(\mathcal{S}, \mathcal{A}) \subset \Omega(\mathcal{T}, \mathcal{J}, \mathcal{R}, \mathcal{I}, \mathcal{DD})
\end{aligned} \tag{1}$$

The model may interact with the short-term scheduling level either offline or online.

6.4
Flexible Recipe Model for Recipe Initialization

At the start of a batch the initial conditions may differ from those prescribed by the master recipe, even to the extent of making successful completion unlikely. Examples are deviations in catalyst activity, available heat, raw material quality and equipment fouling, among others. In such cases, the flexible recipe concept makes it possible to alter the still-adjustable process conditions, so as to ensure the most successful completion of the run. Otherwise, because of scheduling requirements, it may be worthwhile modifying the processing time of a stage of the master recipe, by modifying some operating conditions, so as to debottleneck some piece of equipment or accomplish some product due-dates. The procedure of generating the initialized control recipe from the master recipe is called recipe initialization. This procedure also implies the need to specify in which specific equipment unit and in which product sequence will each stage of the recipe be carried out.

In general, the objective is to generate the best control recipe for different production scenarios. Specifically, the proposed framework adjusts the different parameters of a master control recipe model to deviations in prices or quality of delivered raw materials and in expected initial process conditions. For instance, in such a case where available steam pressure is lower than the nominal value at the beginning of a batch, recipe items will have to be initially adapted to this fact. Another aim of this framework is to adjust the different recipe items to the availability of plant resources and equipment units.

The inputs of the problem are the production master recipe for each product, that is, the different components that define each recipe, the available equipment units for each task, the list of common utilities, the market requirements expressed as specific amounts of products to be delivered at given instants, and others. The algorithm has to determine the optimal sequence of the tasks to be performed in each unit, the values of the different parameters that specify each recipe, that is, the *initialized control recipe* and the use of utilities as a function of time.

Specifically, the optimal schedule in each case is efficiently reached using the S-graph approach [27]. This approach implies a branch-and-bound algorithm. This algorithm proceeds from a root node corresponding to the nominal master control recipe. From this root, partial schedules (nodes of the tree) are built adding schedule-arcs to the preceding nodes. At each node, a flexible recipe model is solved to calculate a relaxation of the algorithm. The solution of this model at the end of a leaf gives the optimal timing, considering the flexible recipe, of the schedule associated to that leaf. The optimal schedule corresponds to the leaf with best objective function value. Hence, a model for schedule timing integrated with a flexible recipe is necessary. The proposed model is linear, simply to permit a rapid convergence of the algorithm.

6.4.1
Flexible Recipe Model for Schedule Timing

In addition to timing restrictions, two sorts of flexible recipe constraint have to be considered: product specifications and process operating conditions and their consequences on the production cost. The product specifications vector, $\boldsymbol{ps}$, is a function, Ψ, of processing time, waiting time, process set-points, and other product specifications. The model adjusts these recipe parameters for each individual batch performed in a specific production plan, $\boldsymbol{\Theta}$, this plan being a function of orders to be satisfied, S, and of plant resources, A. Each production environment also has to meet some physical plant constraints, Ω, such as the plant topology, T, operating with a set J of equipment units and a set R of production resources. Each production plan is generated to meet the market constraints (set of production orders in a given set DD of time horizons or due dates). A performance criterion ϕ is also included.

Hence, two sort of flexible recipe constraints have to be considered to define the flexible recipe model Ψ: product specifications (quality of the final products) and process operation conditions (set points) and their consequences on the production cost.

6.4.2
Quality and Production Cost Model

Product specifications, $\boldsymbol{ps}_i$, might depend on processing time, waiting time, process operation conditions and product specifications at different stages i^* where different inputs to stage i are processed. At the first stage of a batch, i^* will represent the raw materials. It will also be assumed that, within a time interval, a linear model can be adjusted to predict small deviations from process specifications, $\boldsymbol{\delta}\mathrm{ps}_i$, as a function of small deviations from the nominal values of PT_i, TW_i, $\boldsymbol{poc}_i$ and $\boldsymbol{ps}_{i^*}$ (Eq. (2)).

$$\delta \boldsymbol{ps}_i = \boldsymbol{a}_i \delta PT_i + b_i TW_i + \sum_{i^*} \boldsymbol{C}_{i,i^*} \delta \boldsymbol{ps}_{i^*} + \mathbf{d}_i \delta \boldsymbol{poc}_i \qquad (2)$$

where $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$, are the vectors that linearly correlate the effect of processing and waiting times of stage i on product specifications. $\boldsymbol{C}_{i,i^*}$ is the matrix that linearly correlates the effect of the different product specification inputs to stage i from stage i^* on product specifications, and $\boldsymbol{d}_i$ the vector that correlates the effect of small deviations in process operation values on the product specifications.

For instance, consider the production of one batch of product A. The stage i of this process consists in heating A in equipment unit 1. Stage $i + 1$ constitutes the reaction of A to give B in equipment unit 2. The main important product specification at stage $i = 1$ is the temperature reached in unit 1 and at the second stage, the conversion of reactant A and the temperature at the end of this stage. Therefore, the vector $\boldsymbol{ps}_1$ will only contain one element (temperature at the end of the stage 1). The vector $\boldsymbol{ps}_2$ will have two elements, conversion of reactant and temperature. The vector $\boldsymbol{a}_1$ will consequently contain one element that will correlate the effect of small deviations in processing time of stage 1 on the temperature reached at stage 1. Similarly, $\boldsymbol{a}_2$ will have two elements, and each element will correlate the effect of processing time on each relevant product specification j, $\boldsymbol{ps}_{j,\,2}$. If waiting time has no effect on product specifications, the vector $\boldsymbol{b}_i$ is null. Otherwise, product specifications at stage 2 will clearly be affected by product specifications at stage 1. So, the matrix $\boldsymbol{C}_{2,\,i^*}$ will be $\{1 \times 2\}$. Its elements correlate the effect of small deviations in the temperature reached at stage 1 on the conversion and temperature at the end of stage 2.

Final products must meet some quality (product specifications) requirements. The model also considers the possibility of mixing different batches of the same product, produced within a fixed horizon, to be sold or used together. Therefore, the properties of the last task of each batch, or, in the case of some batches being mixed, the properties of the final products mixed, must meet such requirements, $\delta \boldsymbol{ps}_p^o$. That is, only deviations up to a point will be permitted (Eq. (3)).

$$\sum_m B_m \delta \boldsymbol{ps}_m \leq \delta \boldsymbol{ps}_p^o \sum_m B_m \quad \forall p, \forall m \tag{3}$$

where B_m is the batch size of product p at stage m, and m belongs to the set of last recipe stages of product p batches that are mixed.

Process operation modification can have an influence on the operation cost. This fact is also considered in the flexible recipe model. Thus, within a time interval, the set-point modification is assumed to have a linear dependence with batch-stage cost (Eq. (4)).

$$\delta Cost_i = \boldsymbol{f}_i \delta \boldsymbol{poc}_i \tag{4}$$

6.4.3
Flexibility Regions

In Eq. (5), $\boldsymbol{\Delta}$ and $\boldsymbol{\sigma}$ define the flexibility regions for $\boldsymbol{poc}_i$ and $\boldsymbol{ps}_i$ respectively. The width of these regions will basically depend on the accuracy of the model presented in the previous section. That is, the regions are defined in which the model deviates from reality by only a predetermined percentage value, ε. Assuming linearity, each of these regions can be described by a set of R^n hyper planes (Eq. (5)) where n will be number of variables considered or degree of flexibility of the batch process considered.

$$\boldsymbol{L}_i \delta \boldsymbol{poc}_i + \boldsymbol{l}'_i \delta \boldsymbol{PT}_i + \boldsymbol{l}''_i \delta \boldsymbol{TW}_i \leq \boldsymbol{M}_i \quad \forall i \tag{5}$$

where $\boldsymbol{L}_i$, $\boldsymbol{l}'_i$ and $\boldsymbol{l}''_i$ are the matrices that define the hyper planes bounding ($\boldsymbol{M}_i$) the process flexibility to be considered within the linear model.

6.4.4
Integration with the Scheduling Tool

Within the S-graph framework, a partial schedule is obtained at each node of the branch-and-bound algorithm. That is, at each node some equipment units may be already scheduled and some others not. The problem is relaxed by solving the linear flexible recipe model. Therefore, if a node has a relaxation higher than the best bound, the branch corresponding to that node is cut. Figure 6.8 shows the Linear Progamming (LP) model to be solved at each node of the branch-and-bound algorithm procedure where the objective function contemplates a trade off between production makespan and production costs. Thus, the recipe is optimized as well as the timing of the partial schedule. Here TI_i and TF_i are the starting and ending times of task i respectively, S_i is the set of states that task i generates and S_i^* the set of states that feed task i^*.

Timing of the schedule constraints,

$$TI_i \geq 0$$
$$TF_i = TI_i + PT_i + TW_i \quad \forall i$$
$$TI_i = TF_i \quad \forall i, i'/\exists s \in \left\{ \overline{S}_i \cap S_{i*} \right\}$$
$$TW_i \leq TW_i^{\max} \quad \forall i$$
$$MS \leq TF_i \quad \forall i$$

Flexible recipe model,

$$\boldsymbol{\delta ps}_i = \boldsymbol{a}_i \delta \boldsymbol{PT}_i + \boldsymbol{b}_i TW_i + \sum_{i^*} \mathbf{C}_{i,i^*} \boldsymbol{\delta ps}_{i^*} + \boldsymbol{d}_i \boldsymbol{\delta poc}_i$$

Flexibility region,

$$\boldsymbol{L\delta poc}_i + \boldsymbol{l}_i' \boldsymbol{\delta PT}_i + \boldsymbol{l}_i'' \delta TW_i \leq \boldsymbol{M}_i \quad \forall i$$

Performance criterion,

$$\sum_m B_m \delta \boldsymbol{ps}_m \leq \delta \boldsymbol{ps}_p^0 \sum_m B_m \quad \forall p, \ \forall m$$
$$\delta Cost_i = \boldsymbol{f}_i \boldsymbol{\delta poc}_i$$
$$\min \left(MS \cdot F^* + \sum_i \delta Cost_i \right)$$

Figure 6.8 Formulation for recipe initialization and multipurpose batch process schedule timing

6.4.5
Motivating Example

The proposed framework for recipe initialization integrated to production scheduling has been tested in the batchwise production of benzyl alcohol from the reduction of benzaldehyde through a crossed Cannizarro reaction. This reaction has been extensively studied by Keesman [28]. In that work, an input-output kind of black box model is developed in order to describe the behavior of the reaction phase of the recipe. The model predicts the reaction yield, $ps_{i,1}$, as a function of the reaction temperature, $poc_{i,1}$, reaction time, PT_i, amount of catalyst, $poc_{i,2}$ and amount of one reactant in excess, $poc_{i,3}$. Then, the model is used to optimize different recipe components analyzing the effects of model accuracy on the results. However, in that work only one batch phase of the recipe was considered. In the following study, the whole batch recipe train and a production environment are considered in order to fully exploit the potential of a more realistic batch process scenario.

The flexible recipe model, Ψ, for this reaction phase and given the linearity required by the model proposed in Section 6.4.2, becomes,

$$\boldsymbol{\delta ps}_{i,1} = 4\delta PT_i + (4.4, 95, 95)\begin{pmatrix}\delta poc_{i,1}\\ \delta poc_{i,2}\\ \delta poc_{i,3}\end{pmatrix} \quad \forall i \in \{\text{Reaction phase}\} \tag{6}$$

The coefficients of Eq. (6) are the linear coefficients of the Keesman quadratic model. The flexibility of this batch stage, contained in $\boldsymbol{\Delta}$ and $\boldsymbol{\sigma}$ regions according to Eq. (5), is defined by the set of cutting planes (Eq. (7)) that bounds the deviation of $\sigma ps_{i,1}$ predicted by Eq. (6) and that predicted by the quadratic model.

For simplicity, it has been assumed that the hypervolume of $\boldsymbol{R}^4$ containing $\boldsymbol{\Delta}$ and $\boldsymbol{\sigma}$ is a hypercube. Equation 7 represents the hypercube of maximum volume that bounds the flexibility region with a tolerance of less than 1.5 % for the reactant conversion.

$$\begin{pmatrix} 1 & 0 & 0 & 0\\ -1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & -1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & -1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix}\boldsymbol{\delta poc}_{i,1}\\ \boldsymbol{\delta poc}_{i,2}\\ \boldsymbol{\delta poc}_{i,3}\\ \boldsymbol{\delta poc}_{i,4}\end{pmatrix} \le \begin{pmatrix} 0.5\,^\circ\mathrm{C}\\ 0.7\,^\circ\mathrm{C}\\ 8.5\,\mathrm{g}\\ 27\,\mathrm{g}\\ 7.5\,\mathrm{g}\\ 90\,\mathrm{g}\\ 0.1\,\mathrm{h}\\ 0.3\,\mathrm{h} \end{pmatrix} \tag{7}$$

This reaction stage has been incorporated in the whole recipe. It is assumed that a preparation stage performed in equipment unit U1 and two separation stages carried out in equipment units U3 and U4 are also necessary to produce the alcohol. The reaction stage takes place in equipment unit U2. Reaction temperature at the second stage, $\delta poc_{i,1}$, depends on the temperature reached at the first one, $\delta ps_{i',2}$, as follows:

$$\delta poc_{i,1} = \delta ps_{i',2} \tag{8}$$

where i' corresponds to any preparation stage and i to any reaction stage of the alcohol recipe. The temperature reached at the preparation stage depends on the processing time according to:

$$\boldsymbol{\delta ps}_{i',2} = 10\delta PT_{i'} \tag{9}$$

This recipe has been introduced into the production scenario given in Table 6.1. P1 represents the production of benzyl alcohol. The rest of products P2, P3, and P4 share equipment units and resources with product P1.

Table 6.1 Batch production environment

Products (N)	Equipment unit Processing Time (h)				Number of batches
P1	U1	U2	U3	U4	3
	0.5	1.75	2.0	0.5	
P2	U1	U3	U4	U6	1
	1.0	2.0	1.5	1.0	
P3	U7	U4	U6	U5	2
	2.0	1.0	1.0	1.0	
P4	U2	U3	U7	U5	1
	1.5	1.0	2.0	1.5	

Figure 6.9 shows the Gantt charts corresponding to the optimum production scheduling for the proposed case study when the fixed recipe at nominal operation conditions is contemplated and when recipe adaptation is considered. The resultant production makespan is 10.75 h for the fixed recipe environment. When the proposed flexible recipe framework is considered, the production makespan diminishes to 10.45 h (2.8% makespan reduction). Also, a different sequence of batches is obtained when it is imposed that the mixing of the three batches of alcohol has to meet the nominal reaction yield ($\delta ps_p{}^o = 0$). The optimal solution is obtained in 25.5 CPU seconds using an AMD-K7 Athlon 1 GHz.

The resultant process operating conditions of the three alcohol batches for the flexible recipe scenario are summarized in Table 6.2.

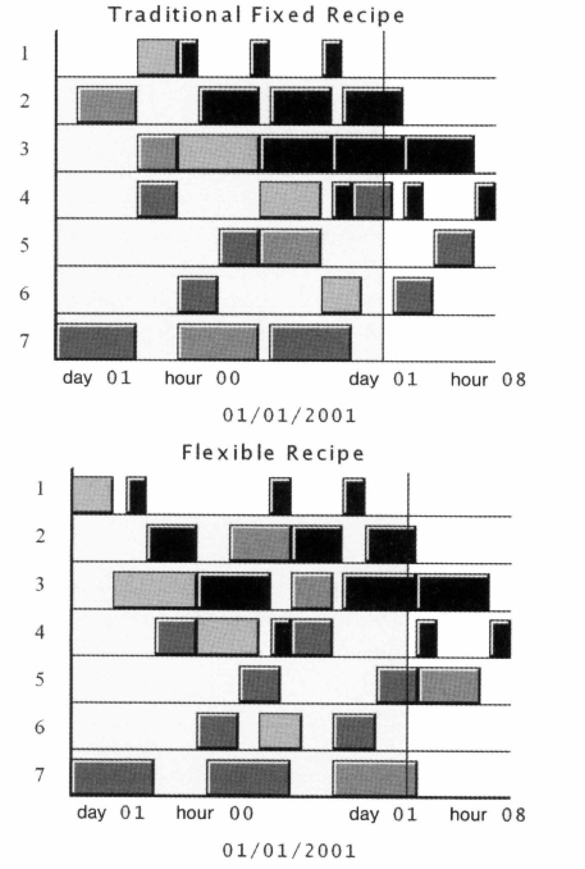


Figure 6.9 Optimal Gantt chart of batch production environment of Table 6.1 when considering the fixed recipe and the recipe adaptation respectively. The case study recipe is represented in **black**

Table 6.2 Formulation for recipe initialization and multipurpose batch process schedule timing

Batch	Temperature (°C)	Processing time (h)	Amount of KOH (g)	Amount of H_2CO (g)	Conversion (%)
1st	64.5	1.2	500	425	75
2nd	64.5	1.2	500	425	75
3rd	63.8	1.2	500	425	72

To see the effect of initial process deviations on the recipe, Keesman [28] limited the reaction temperature to 63 °C ($\delta poc_{i,1} = -1$). After optimizing the reaction stage alone, it is found that the reaction time has to be extended to 1.76 h so that the total amount of KOH reaches 528 g and the amount of formaldehyde goes to 475 g in order to keep the intended reaction yield. For these new nominal conditions, the resultant production makespan for the scenario described in Table 6.1 is 11.03 h, which means a reduction in productivity of 5.5 %. Otherwise, a better process performance can be achieved by applying the flexible recipe model to optimize the entire batch plant. The linear flexible recipe, Ψ, and the model validity constraints for these new nominal conditions are shown in Eqs. 10 and 11, respectively.

$$\boldsymbol{\delta ps}_{i,1} = 3.75\delta PT_i + (101, 112.5)\begin{pmatrix}\delta poc_{i,2}\\ \delta poc_{i,3}\end{pmatrix} \quad \forall i \in \{\text{Reaction phase}\} \tag{10}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0\\ -1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & -1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & -1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta poc}_{i,1}\\ \boldsymbol{\delta poc}_{i,2}\\ \boldsymbol{\delta poc}_{i,3}\\ \boldsymbol{\delta PT}_i \end{pmatrix} \le \begin{pmatrix} -1\,^{\circ}\mathrm{C}\\ 1\,^{\circ}\mathrm{C}\\ 12\,\mathrm{g}\\ 23\,\mathrm{g}\\ 13\,\mathrm{g}\\ 28\,\mathrm{g}\\ 0.57\,\mathrm{h}\\ 0.4\,\mathrm{h} \end{pmatrix} \tag{11}$$

Now, the optimal production makespan becomes 10.61 h. Therefore, using the proposed framework, limiting the reaction temperature to 63 °C, only implies a 1.5 % reduction in process productivity. The new process conditions for the different batches of the alcohol production appear in Table 6.3.

Table 6.3 Optimal process operation conditions for three batches of alcohol after limiting reaction temperature

Batch	Temperature (°C)	Processing time (h)	Amount of KOH (g)	Amount of H_2CO (g)	Conversion (%)
1st	63	1.55	512	438	78.3
2nd	63	1.36	512	438	71.9
3rd	63	1.36	512	438	71.9

Notice that in this case study the cost of modifying different process variables has been considered negligible. Usually, nominal values should correspond to an economic optimum. Thus, altering such nominal conditions should result in overrunning this economic optimum in spite of an eventual increase in plant productivity. Obviously, a more realistic scenario should also consider the costs associated to deviations in process operation conditions from nominal values.

6.5
Flexible Recipe Model for Recipe Correction

The recipe initialization is performed at the beginning of the batch phase, taking into account known initial deviations. But other run-time deviations may arise. However, under certain circumstances it is possible to compensate for the effects of these unknown disturbances during the batch run, provided that continuous or discrete measurements are available.

The flexible recipe model is the relationship that correlates a batch process output as a function of the selected input items of the recipe. This model is regarded as a constraint on quality requirements and on production cost. Figure 6.7 shows the environment proposed here for real-time recipe correction.

While a batch process takes place, different online continuous process variables and discrete variables values, sampled at different times, are taken. From this information, a process state assessment is performed. This assessment gives information about how the batch process is being carried out to the flexible recipe model for recipe correction. The time at which process state assessment is performed, and so at which actions take place, might be different from the moment at which a deviation is detected. Interaction or integration of the flexible recipe model with production scheduling algorithms is necessary to account for the ultimate effect of recipe correction on overall plant capacity.

Three different kinds of models are identified:

- A prediction model that estimates the continuous and discrete sampled (at the sampling time) product specification variables, as a function of the actual control recipe that has already been established by the offline initialization tool. Then, the process state assessment consists of the evaluation of the batch-process run. The predicted product specification i, $\boldsymbol{pvs}_i^w$, expected by the offline recipe initialization model, is compared with the actual variable observed at the wth process statement, $\boldsymbol{ps}_i^w$. If this deviation observed is greater than a fixed permitted error, ε, some actions will be taken in order to offset this perturbation.
- A correction model for control recipe adjustments, which describes the ultimate effect of the values measured at the time of the process state assessment as well as of those run-time corrections made during the remainder of the processing time.
- A rescheduling strategy to adjust the actual schedule to the recipe modifications.

6.5.1
Rescheduling Strategy, Ω

The output of the flexible recipe model for recipe correction might give variations in processing time or resource consumption, which would make the existing plant resources schedule suboptimal or even infeasible. Therefore, in order to accommodate for these deviations in the actual plant schedule, a rescheduling strategy is to be used. There are two basic alternatives to update a schedule when it becomes obsolete: to generate a new schedule or to alter the initial schedule to adapt it to the new conditions. The first alternative might in principle be better for maintaining optimal solutions, but these solutions are rarely achievable in practice and require prohibitive computation times.

Hence, a retiming strategy is integrated into the flexible recipe model for recipe correction. At each deviation detected, optimization is required to find the best corrected control process recipe. From this, it is proposed to solve the LP shown in Eq. (12) along with a linear representation of the recipe correction model, to adjust the plant schedule to each recipe correction. In case of dealing with a multipurpose plant, it might happen that a given schedule becomes infeasible because of process disturbances. In such a situation further actions should be taken, like changing the order sequence or canceling a running batch:

$$\begin{aligned}
&\min\left(\Phi(PT_i, TW_i, \boldsymbol{ps}_i, \boldsymbol{poc}_i, \boldsymbol{\Theta})\right) \text{ subject to,}\\
&TI_{i,j} \geq 0 \quad \forall i,j\\
&TF_{i,j} = TI_{i,j} + PT_i + TW_{i,j} \forall i,j\\
&TI_{i,j} = TF_{i',j} \forall j, i, i' / \exists s \in \left\{\overline{S}_i \cap S_{i'}\right\}\\
&TI_{i,j} \geq TF_{i,j-1} \forall i,j\\
&TW_{i,j} \leq TW_i^{\max} \forall i,j\\
&\text{correction flexible recipe model constraints}
\end{aligned} \tag{12}$$

where $TI_{i,j}$, $TF_{,j}$, PT_i and $TW_{i,j}$ are the initial, ending, processing and waiting times of each stage i of a batch corresponding to the specific sequence j of the schedule. The sequence is assumed to be fixed. S_i is the set of stages that feed stage i. S_i' is the set of stages fed by stage i'. ϕ is the performance criterion of the flexible recipe model.

6.5.2
Batch Correction Procedure

Within each batch-run, the algorithm of Fig. 6.10 is applied. This algorithm first predicts the expected deviations in process variables from the nominal values as a function of the corrections already taken. Then, the process state assessment verifies if there exist significant discrepancies between the observed variables and the predicted. If so, it freezes process variables of all batch-stages already performed and of the batch-stages that are currently being performed and are not the actual batch-stage

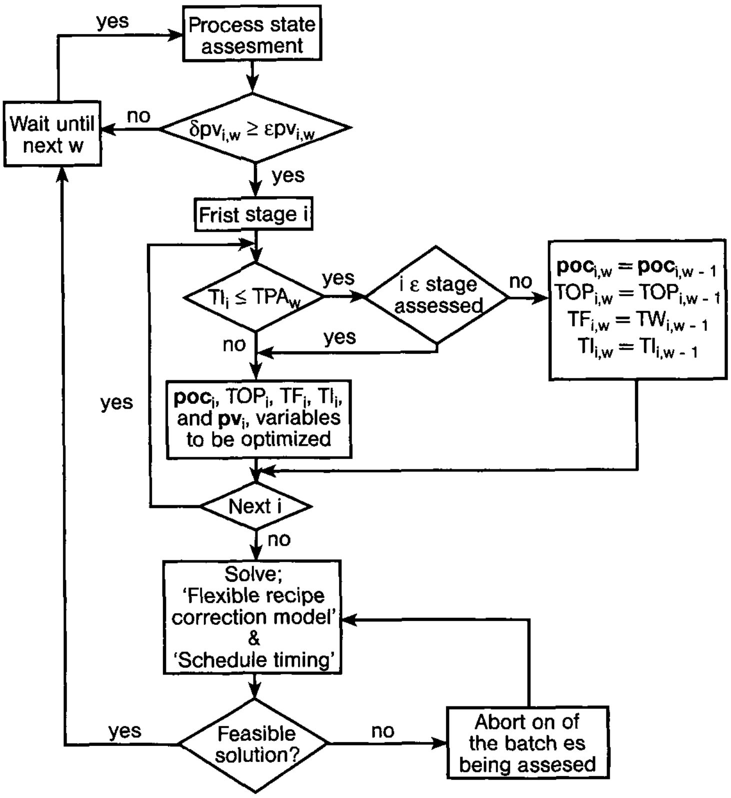


Figure 6.10 Batch correction procedure algorithm

of assessment and reoptimizes the actual recipe taking into account the effect on the schedule timing.

6.5.3
Application: Model-Based Advisory System for Recipe Correction and Scheduling

In this section, an integrated model-based advisory system is designed to give online support for batch process operation. This application integrates recipe modifications as well as modifications in the actual plant schedule-timing, thus breaking the traditional approach to disassociate recipe correction from plant-wide adjustment. It is based on the flexible recipe concept.

The application envisaged gives advice to plant operators and schedulers on how to react in the face of disturbances, so that different kinds of actions can be supported, for instance:

- correct recipe parameters to offset compensating disturbances to meet product specifications at the expense of processing time;

- allow batches to end on time, that is, finishing the batch below expected product quality;
- reduce processing times in order to fit a rush order;
- modify process operating conditions and processing time to partly compensate for disturbances, finishing the batch below product specifications, but not as low as if no action would have been taken.

Different elements form the integrated model advisory system presented here; The recipe adaptation set, where recipe flexibility is defined, the plant adaptation set, where plant schedule adaptation is included using different kinds of rescheduling alternatives, and finally, an integrated criterion, where recipe items modifications, product due-dates accomplishment and product specification deviations costs are included.

In this application, the flexible recipe concept is included in the recipe adaptation set. This set consists of a statistical process model optimized from historical process data and relevant model constraints. Variables in the process model may be classified into those that appear perturbed (***P***), those that may be tackled to compensate disturbances (***M***) and those that define the output of the batch process in terms of quality or yield (**O**). Hence, the flexible recipe model (Ψ), included at the recipe adaptation set is as follows,

$$\begin{aligned} &\mathbf{O} = \Psi(PT, \mathbf{M}, \mathbf{P}) \\ &\mathbf{O} \subset \sigma \\ &\{\mathbf{M}, PT\} \subset \delta \end{aligned} \tag{13}$$

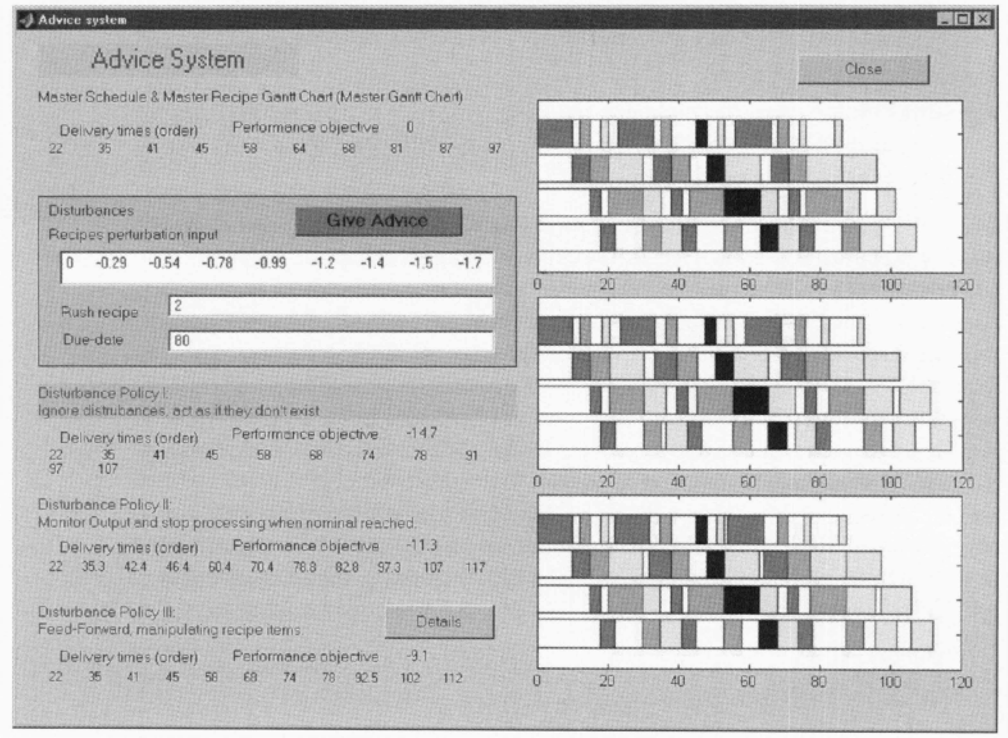
where $\boldsymbol{\delta}$ is the flexibility region for process operating conditions and $\boldsymbol{\sigma}$ is the flexibility region for product specifications.

The plant adaptation set, the other key concept of the advisory system, describes the plant resources management, including the relevant equipment information for scheduling, and defines penalties for due-date violations for the accepted orders. In this application, sequence of products is predefined and is assumed not to change. Hence, the plant adaptation set is described by Eq. (13).

When a deviation between the expected and the actual behavior during processing is observed, some advice on how to react is requested. The application presented here considers two different kind of perturbations: process disturbances on some input variable of the batch recipe stages, and rush orders, a new order to be satisfied at a specific due-date and to be fitted in the actual production plan. Figure 6.11 shows this advice system mechanism window.

The application being presented here has been simplified to just consider a linear flexible recipe model at the recipe adaptation set. Then, an LP formulation is used to calculate process adaptation effects.

As soon as a deviation is detected, the control recipe can be readjusted, and this readjustment may have an impact on the whole plant operation. A number of scenarios for recipe readjustment are considered, for instance;



Advice System

Master Schedule & Master Recipe Gantt Chart (Master Gantt Chart)

Delivery times (order) Performance objective 0
22 35 41 45 58 64 68 81 87 97

Disturbances
Recipes perturbation input
0 -0.29 -0.54 -0.78 -0.99 -1.2 -1.4 -1.5 -1.7
Rush recipe 2
Due-date 80

Disturbance Policy I:
Ignore distrubances, act as if they don't exist

Delivery times (order) Performance objective -14.7
22 35 41 45 58 68 74 78 91
97 107

Disturbance Policy II:
Monitor Output and stop processing when nominal reached.

Delivery times (order) Performance objective -11.3
22 35.3 42.4 46.4 60.4 70.4 78.8 82.8 97.3 107 117

Disturbance Policy III:
Feed-Forward, manipulating recipe items.

Delivery times (order) Performance objective -9.1
22 35 41 45 58 68 74 78 92.5 102 112

Figure 6.11 Integrated batch operation advice system results.

- The perturbation may be compensated for within the same batch stage, tackling manipulating variables with a consequence on operation cost.
- Other batches may have to be corrected, for instance reducing their processing time, to accommodate the impact of correcting a disturbance on a specific batch-stage (i.e., on its processing time).
- The timing of the schedule may have to change, imposing a delay on product delivery.

An integrated optimization criterion computing consequences of different scenarios coordinates the recipe adaptation set actions with the plant adaptation set ones in order to maximize overall batch plant performance. This architecture has been implemented in MATLAB 6.0.

When disturbances are encountered, the recipe adaptation set may decide to vary some processing times of some tasks of some recipes. This will have an impact on the actual production schedule, so some orders will have to be shifted forward (increased). The plant adaptation set is optimizing this order-shifting to minimize a function depending on delivery-date delays. That is, not all order delays will have the same (for instance, economic) impact on the overall objective function, but the plant adaptation set, tries to shift orders so that the overall impact is minimized. This problem results in an LP formulation that is solved using MATLAB optimization toolbox.

6.5.4
Advisory System Case Study

The case study corresponds to a multiproduct batch plant, over an advising-span of 1 week. During this week, the plant is producing three different products with a specific given sequence. The recipes and sequence of the case study are shown at Tables 6.4 and 6.5, respectively.

Table 6.4 Master recipes of advisory system case study

Recipes	Processing time of stages (h)			
R1	10	5	3	4
R2	3	5	10	5
R3	2	10	5	6

Table 6.5 Master Schedule of advisory system case study

	Master schedule							
Sequence	R3	R1	R2	R3	R1	R2	R3	R3
Due dates	41	45	58	64	68	81	87	97

Recipe adaptation set (ras) variables are classified into a perturbed variable (*P*), manipulated variable (*M*), output variable (*O*) and processing time (*PT*). For simplification, it is assumed that there is only one (critical) variable of each. Besides, the relationship among these variables is considered to be defined by a linear model and it is considered that there is only one flexible task for each product recipe. Table 6.6 summarizes the recipe adaptation set (ras) parameters.

Table 6.6 Recipe adaptation set of case study

	Flexible		Coefficients for	
Recipe	phase	δM	δI	δPT
R1	2	2.5	2.0	1.5
R2	2	1.0	0.5	0.5
R3	3	2.0	1.0	0.5

A disturbance may be totally offset by modifying the manipulated variable and keeping the master processing time and master output (or quality). Or may be ignored, keeping the manipulated variable and processing time unmodified, so that the output variable (quality) will be affected, or, otherwise, a disturbance may be totally compensated by processing time, or partially by all variables.

The integrated advisory criterion computes the effect of modifying manipulated and output variables from the nominal values. Table 6.7 show the costs associated with modifying these variables. Also, the cost of delivery-dates delay from due dates for each order is shown at Table 6.8.

Table 6.7 Integrated advisory criterion 1

Recipes	Output deviation cost (u)	Manipulated variable deviation cost (u)
R1	1.0	0.1
R2	2.0	0.5
R3	0.5	0.7

Table 6.8 Integrated advisory criterion 2

Delivery-dates deviation cost (u)									
0.2	0.1	0.02	0.5	0.0	0.2	0.1	0.1	0.2	0.03

Two types of disturbances are considered; process disturbances and rush orders. From plant operation, recipe perturbation input variable is retrieved. In this case study, disturbances follow an exponential increase. This would be the case, for instance, of a catalyst being used for all product recipes whose activity is decaying at each use along the production makespan. In the face of disturbances (process disturbances and rush order) the application gives advice on how to react following three policies;

- Policy I. This policy ignores process disturbances, and therefore does not modify manipulated variables or processing times. This situation has a direct impact on the output variable, that is, quality of products.
- Policy II. Here, process disturbances are totally compensated for by modifying processing time (not modifying manipulated variable and keeping output variables equal to nominal values). This situation has a direct impact on delivery dates of products.
- Policy III This policy modifies all recipe items: manipulated variable, processing time and output variable. Within this situation, disturbances may have an impact on delivery dates as well as on product quality, depending on their weight on the overall objective function.

In all policies, a rush order is always accepted. In the case study shown, disturbances have a negative impact of −14.7 u. for policy I, −11.3 u. for policy II and of −9.1 u. for policy III. Policy III is the one containing more degrees of freedom to react in the face of disturbances, and therefore is showing the best performance. Figure 6.11 shows the application results window.

6.6
Final Considerations

The increasing interest recently observed in rationalizing batch/hybrid process operations is well-justified. The great advantage offered by batch process stages resides in their inherent flexibility, which may give an adequate answer to present uncertain product demand, variable customer specifications, uncertain operating conditions, market prices variations and so on [29]. In batch plants, there is no reason why the same product must be made every batch; there is the possibility of tailoring a product recipe specifically for a particular customer. In this chapter firstly a review of relevant approaches to represent and exploit this potential flexibility of batch/hybrid process has been given. A novel framework (flexible recipe) has been presented that allows further exploration of flexible manufacturing capabilities of such types of processes. This framework proposes a new philosophy for recipe management in batch process industries that includes the possibility of recipe adaptation in a real-time optimization environment.

Based on these novel concepts, a model-based integrated advisory system is presented. The system gives on-time advice to operators on how to react when process disturbances occur. This advice takes into account modification in recipe parameters (product quality, specifications, processing time, process variables) as well as modifications in the production schedule. A process state assessment module for evaluation of abnormal situations should advise when proper actions should be taken. The result is a user-friendly application for optimal batch process operation in industrial practice.

Acknowledgments

Financial support for this research received from the "Generalitat de Catalunya", FI program and project GICASA-D (No I-353) are fully appreciated. Also, support received in part by the European Community (project n° G1RD-CT-2001-00466-0466) is acknowledged. Funds were also received from Spanish MCyT (project n° DPI2002-00806). Enlightening discussions and suggestions received from Prof. Antonio Espuña and Prof. Verwater-Lukszo are thankfully appreciated.

Nomenclature

A	Assignment of different batch plant resources.
B_m	Batch size of product p at stage m.
C_{i,i^*}	Correlation matrix with the effect of product specifications inputs to stage i from stage i^*.
DD	Set of production horizon or due-dates.
$\boldsymbol{I}$	Observed perturbed variable.
I	Set of production orders.

J	Set of equipment units.
$\mathbf{M}$	Manipulable variable of advisory system.
$\mathbf{O}$	Output variable of batch process.
$\mathbf{P}$	Perturbed variable.
PT_i	Processing time of each stage i of a recipe.
$poc(t)i$	Process operation conditions vector as a function of time t of stage i.
ps_i^w	Observed product specification vector at the wth process state assessment moment of batch process stage i.
ps_{i*}	Observed product specification vector at the end of batch process stage i^*.
$pvs^w{}_i$	Expected vector of product specification vector at the wth process state assessment moment of batch process stage i.
R	Set of process Resources.
S_i	Set of states generated by task i.
S_{i*}	Set of states that feed tasks i^*.
S	Sequence of different batches.
T	Multistage flowshop or jobshop batch plant topology.
$TI_{i,j}$	Starting time of task i.
$TF_{i,j}$	Ending time of task i.
TPA_i^w	wth moment at which stage i of a batch is being assessed.
TW_i	Waiting time at stage i.
$T(P_i)$	Steam temperature condensation at pressure Pi.
Δ	Flexibility region for process operation conditions.
λ	Steam enthalpy.
Ω	Scheduling constraints.
ϕ	Performance criterion function of the Flexible Recipe model.
Ψ^{pred}	Quality and production cost modelling function of prediction model.
$\Psi^{correct}{}_w$	Quality and production cost modelling function of correction model of the wth process assessment moment.
Θ	A specific production plan.
σ	Flexibility region for product specifications.

References

1 *Reynold T. S.* (1983) 75 years of Progress. A History of the American Institute of Chemical Engineers 1908–1983. American Institute of Chemical Engineers, New York

2 *Parakrama R.* Improving batch chemical processes. The Chem. Eng. 1985 p. 24–25

3 *Reklaitis G. V.* (1985) Perspectives for computer-aided batch process engineering. Chem Eng Prog, 8 (1985) p. 9–16

4 *Reklaitis G. V. Sunol A. K. Rippin D. W. T. Hortacsu D.* (1996) Batch Processing Systems Engineering. NATO ASI Series 143, Spinger-Verlag, Berlin

5 *Stephanopoulos G. Ali S. Linninger A. Salomon E.* AIChE Symp. Ser. 323 (2000) p. 46–57

6 *Puigjaner L. Espuña A. Reklaitis G. V.* (2002). Frameworks for discrete/hybrid production systems. In: Braunschwerg B. Gain R. (eds.) Software Architectures and Tools For Computer Aided Process Engineering. Computer-Aided Chemical Engineering, 11. 88 (No. 9) (1985) Elsevier, Amsterdam, pp 663–700

7 *Engell S. Kowalenski S. Selmlz C Stursberg O.* Continuous-discrete interaction in chemical processing plants. Proc. IEEE 88 (2000) p. 1050–1068

8 *Barton P. I. Pantelides C. C.* Modeling of continued discrete/continuous processes. AIChE J. 40 (6) (1994) p. 966–979

9 *Kondili E. Pantelides C. C. Sargent R. N. H.* A general algorithm for short-term scheduling of batch operations – 1. Mixed integer linear programming formulation. Comput. Chem. Eng. 17 (1993) p. 211–227

10 *Pantelides C. C. Unfixed frameworks for optimal process planning and scheduling. Proceedings of the. 2nd Conference on Foundation of Computer-Aided Process. Operations. CACHE (1994) pp. 253–274, New York*

11 *Schilling G. Pantelides C. C.* A simple continuous time process scheduling fromulation and a novel solution algorithm. Comput. Chem. Eng. S20 (1996) p. S1221–S1226

12 *Reklaitis G. V. Mockus L.* Mathematical programming formulation for scheduling of batch operations based on non-uniform time discretization. Acta Chim. Esloven. 42 (1995) p. 81–86

13 *Mockus L. Reklaitis G. V.* Continuous time representation in batch/semicontinuous process scheduling-randomized heuristics approach. Comput. Chem. Eng. S20 (1996) p. S1173–S1178

14 *Zhang X. Sargent R. W. H.* The optimal operation of mixed production facilities – extensions and improvements. Comput. Chem. Eng. S20 (1996) p. S1287–S1292

15 *Shah N.* Single and multi-site planning and scheduling: current status and future challenges. AIChE Symp. Ser 320 (1998) p. 75–90

16 *Silver E. Ryke D. Peterson R.* (1998) Inventory Management and Production Planning and Scheduling. John Wiley and Sons, New York

17 *Graells M. Canton J. Peschaud B. Puigjaner L.* General approach and tool for the scheduling of complex production systems. Comput. Chem. Eng. 225 (1998) p. S395–S402

18 *Puigjaner L.* Handling the increasing complexity of detailed batch process simulation and optimization. Comput. Chem. Eng. 23S (1999) p. S929–S943

19 *Canton J.* (2003) Integrated Support System for Planning and Scheduling of Batch Chemical Plants. PhD Thesis Universitat Polithcnica de Catalunya

20 *Rijnsdorp J. E.* (1991) Integrated Process Control and Automation. Elsevier Amsterdam

21 *Verwater-Lukszo Z.* (1997) A Practical Approach to Recipe Improvement and Optimization in the Batch Processing Industry. PhD Thesis. Eindhoven Technische Universiteit, Eindhoven, The Netherlands

22 *Box G. E. P. Draper N. R.* (1969) Evolutionary Operation. Wiley, New York

23 *Graells M. Loberg E. Delgado A. Font E. Puigjaner L.* Batch production scheduling with flexible recipes: the single product case. AIChE Symp. Ser. 320 (1998) p. 286–292

24 *Romero J. Espuña A. Friedler F. Puigjaner L.* A newframework for batch process optimization using the flexible recipe. Ind. Eng. Chem. Res. 42 (2003) p. 370–379

25 *ANSI/ISA – S88.01 (1995) Batch Control. Part 1: Models and Terminology. American National Standards Institute, Washington D.C.*

26 *Verwater-Lukszo Z. Keesman K. J.* Computer-aided development of flexible batch production recipes. Prod. Planning Control 6 (1995) p. 320–330

27 *Sanmarti E. Holczinger T. Puigjaner L. Friedler F.* Combinatorial framework for effective scheduling of multipurpose batch plants. AIChE J. 48 (11) (2002) p. 2557–2570

28 *Keesman K. J.* Application of flexible recipes for model building batch process optimization and control. AIChE J. 39 (4) (1993) p. 581–588

29 *Rippin D. W. T.* Batch process systems engineering: a retrospective and prospective review. Comput. Chem. Eng. S17 (1993) p. S1–S13