A New Framework for Batch Process Optimization Using the Flexible Recipe

Javier Romero,[†] Antonio Espuña,[†] Ferenc Friedler,[‡] and Luis Puigjaner*[,†]

Chemical Engineering Department, Universidad Politécnica de Catalunya, Diagonal 647, E-08028 Barcelona, Spain, and Department of Computer Science, Universität Veszprém, Egyetem ú. 10, H-82000 Veszprém, Hungary

Batch processes are characterized by a high degree of flexibility that can be appropriately exploited to obtain a maximum production profit. In principle, the nominal production recipe assumes an optimum balance between quality and costs for batches of products. However, in practice, this optimum performance is achieved only when this balance is extended to management of the entire batch plant. In this work, a framework is presented that fully exploits this inner flexibility of batch processes at the plant level. The framework considers a batch recipe model that interacts with a plant-wide model to constitute the so-called flexible recipe model. First, the potential use of this framework is shown by integrating a general multipurpose batch process scheduling algorithm into a general linear recipe model. Next, this framework is illustrated in practice using a real industrial scenario that exhibits complex recipe behavior. Finally, the benefits of considering the flexible recipe concept for the scheduling of batch processes are discussed.

1. Introduction

Batch processes are normally thought to operate at nominal conditions following fixed recipes. Moreover, these nominal conditions are determined only once and sometimes considering only one stage of the process recipe. This traditional mode of operation, called *fixed recipe operation* in this work, does not allow for adjustments in the availability of plant resources and variations either in the quality of raw materials and/or in the actual process conditions. However, in practice, industrial processes are often subjected to such disturbances but are required to maintain the use of limited plant resources in the best possible way. This situation usually leads to the adaptation, in some way, of plant operation, which is done in a rather unsystematic manner based on the experience and intuition of operators. As an alternative, the concept of *flexible recipe operation* is introduced in this work, and a general framework is presented to systematically deal with the required adaptations at a plant-wide level.

The standard ISA-S88[1] defines reference models for batch process control at different levels in the process industries and sets the terminology that helps to explain the relationships among these models. This standard defines a recipe as an entity that contains all of the information specifically and uniquely required to produce batches of a specific product. Accordingly, recipes must provide a way to describe products and how these products are produced. However, this conceptual definition does not properly consider the complex flexibility that characterizes batch processes. In this context, the concept of flexible recipe seems appropriate as a way to incorporate systematic recipe adaptations to changing plant scenarios in the reference model.

* To whom correspondence should be addressed. Phone: +34 934 01 66 74. Fax: +34 934 01 71 50. E-mail: luis.puigjaner@upc.es.

[†] Universitat Politècnica de Catalunya.

[‡] Universität Veszprém.

An idea similar to the flexible recipe concept was considered for the first time in the context of evolutionary operation.[2] The main objective of that approach was to gain statistical insight into the problem behavior so that process efficiency could be gradually improved through suggestions of minor recipe modifications in each batch run. However, it was not until the work of Rijnsdorp[3] appeared 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 and applied to a batch fermentation process and some other academic case studies.[4,5] This philosophy distinguishes two main levels in the flexible recipe: (1) 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 (2) the recipe correction level, where the initialized control recipe is adjusted to run-time process deviations, thus generating corrected control recipes.

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 to an entire batch train was tried for the multiproduct case.[6] Here, recipe elements were optimized along with production scheduling by solving an MINLP problem using genetic algorithms. However, standard quality models were assumed for process recipes, and hence, no insight into recipe behavior was obtained.

In this work, a new framework for recipe initialization that integrates a recipe model into the batch plant-wide model is introduced. The aim of this approach is to optimize the entire batch process, from recipe set-point adjustment to product sequencing. Generally, this initialization is a function of the expected initial process

10.1021/ie020002n CCC: $25.00
Published on Web 12/12/2002

conditions, market requirements, customer requirements, and other factors. Therefore, this work considers the adjustment of these recipe elements in the context of the scheduling requirements. For this purpose, a recipe model and a plant-wide production model are required to build the flexible recipe model.

2. The Flexible Recipe Model

The flexible recipe model represents the relationship that correlates the output of a batch process as a function of the selected input parameters of the recipes for different batch plant scenarios. We identify four main components of the problem: quality, operating conditions, production costs, and production due dates (or, for simplicity, makespans).

The flexible recipe model can be applied to a variety of scenarios. For instance, during batch process operation, processing times of some tasks can 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 us assume a process in which A is converted into B; one batch with low conversion of A could be compensated by another batch with a higher conversion, assuming that these two batches are going to be mixed afterward, 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, consider 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 operating costs. For instance, the reaction time could be reduced by increasing the reaction temperature, although this recipe modification would imply a higher operating cost. Which of the above-mentioned strategies should be applied in each case will depend on the specific process and on the available knowledge about the different tasks of the process. For example, such methods of operation might not be very suitable for highly restrictive processes, such as those found in the pharmaceutical industry, but they are probably convenient for 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.

Proposed Concept for the Flexible Recipe Model. *The flexible recipe model is regarded as a constraint on quality requirements and on production costs.*

In our approach, recipe components are classified into four groups: (1) the vector of process operating conditions, $\mathbf{poc}_i$, of stage i of a recipe, which includes parameters such as temperature, pressure, type of catalyst, batch size, etc.; (2) the product specification vector, $\mathbf{ps}_i$, at the end of each process stage i of a recipe, which might include parameters such as reactant conversion, purity, or quality; (3) the processing time, TOP_i, at each stage i of a recipe.; and (4) waiting time, TW_i, that is, the time between the end of a stage and the beginning of the next stage.

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, $\mathbf{ps}$, and process operating conditions, $\mathbf{poc}$, are subject to optimization within flexibility regions $\boldsymbol{\sigma}$ and $\boldsymbol{\Delta}$, respectively.

A general algorithmic 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 batch process scheduling to be integrated into the recipe optimization procedure.

Each specific production plan $\boldsymbol{\Theta}$ is defined when the different batches are sequenced, $\mathscr{S}$, and when the different resources are assigned, $\mathscr{A}$, to each batch. In addition, each plan has to meet certain scheduling constraints, Ω, such as constraints on the multistage flowshop or jobshop batch plant topology, $\mathscr{T}$; the set $\mathscr{J}$ of equipment units; and the set $\mathscr{R}$ of process resources. Each plan will be generated to meet the market constraints: set $\mathscr{I}$ of production orders in a given set $\mathscr{PT}$ of time horizon or due dates.

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

$$
\begin{aligned}
&\text{optimize } \Phi(\mathrm{TOP}_i,\mathrm{TW}_i,\mathbf{ps}_i,\mathbf{poc}_i,\boldsymbol{\Theta})\\
&\text{subject to recipe constraints}\\
&\mathbf{ps}_i = \Psi(\mathrm{TOP}_i,\mathrm{TW}_i,\mathbf{ps}_{i^*},\mathbf{poc}_i)\\
&\mathbf{ps}_i \subset \boldsymbol{\sigma}\\
&\mathbf{poc}_i \subset \boldsymbol{\Delta}\\
&\text{subject to scheduling constraints}\\
&\boldsymbol{\Theta}(\mathscr{S},\mathscr{A}) = \Omega(\mathscr{T},\mathscr{J},\mathscr{R},\mathscr{I},\mathscr{PT})
\end{aligned}
\tag{1}
$$

3. S-Graph Framework Approach to Recipe Initialization

Here, the general recipe model is integrated into a multi-purpose scheduling model to show the potential application of the flexible recipe model described in section 2.

Production scheduling is an important area in chemical engineering that has received significant attention.[7–9] General purpose models, based on MILP and MINLP formulations, increase rapidly in dimension when used to solve scheduling problems far more complex than those that consider only a multipurpose or multiproduct batch plant without intermediate storage. To overcome this situation, a method is proposed here to integrate a recipe model into the specific multipurpose batch process scheduling algorithm S-graph.[10]

The S-graph approach has the advantage of exploiting problem-specific knowledge from the very beginning to develop efficient scheduling algorithms. Hence, this superior performance is utilized to derive an efficient algorithm for solving the scheduling of batch processes associated with linear flexible recipe models.

The inputs of the problem are the production master recipe for each product, that is, the different components that define each recipe; the available units for each task; the list of common utilities; 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 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 reached using the S-graph framework. This framework applies 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 by 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 with that leaf. The optimal schedule corresponds to the leaf with the best objective function value.[11]

Hence, a model for schedule timing integrated into the recipe model is necessary. The proposed model is linear simply to permit rapid convergence of the algorithm (see Appendix 1).

3.1. Flexible Recipe Model for Schedule Timing. In addition to timing restrictions, two sorts of flexible recipe constraints have to be considered to define Ψ: product specifications (quality of the final products) and process operating conditions (set points) and their consequences on the production cost.

3.1.1. Quality and Production Cost Model, Ψ. Product specifications, $\mathbf{ps}_i$, might depend on processing time, waiting time, process operating 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. Moreover, it will be assumed that, within a time interval, a linear model can be adjusted to predict slight deviations in process specifications, $\boldsymbol{\delta}\mathbf{ps}_i$, as a function of small deviations from the nominal values of TOP_i, TW_i, $\mathbf{poc}_i$, and $\mathbf{ps}_{i^*}$ (eq 2) where $\mathbf{a}_i$ and $\mathbf{b}_i$ are the vectors

$$\boldsymbol{\delta}\mathbf{ps}_i = \boldsymbol{a}_i\boldsymbol{\delta}\mathrm{TOP}_i + \boldsymbol{b}_i\delta\mathrm{TW}_i + \sum_i \mathbf{C}_{i,i^*}\boldsymbol{\delta}\mathbf{ps}_{i^*} + \mathbf{d}_i\boldsymbol{\delta}\mathbf{poc}_i \tag{2}$$

that linearly correlate the effects of the processing and waiting times of stage i on the product specifications. Let $\mathbf{C}_{i,i^*}$ be the matrix that linearly correlates the effects of the different product specification inputs to stage i from stage i^* on the product specifications, and let $\mathbf{d}_i$ the vector that correlates the effect of small deviations in process operating values on product specifications.

For instance, consider the production of one batch of product A. The ith stage of this process consists of heating A in equipment unit 1. Stage $i + 1$ considers the reaction of A to give B in equipment unit 2. The most important product specification at stage $i = 1$ is the temperature reached in unit 1, and at stage $i = 2$, it is the conversion of reactant A and the temperature at the end of this stage. Therefore, the vector $\mathbf{ps}_1$ will contain only one element (temperature at the end of the stage 1), whereas the vector $\mathbf{ps}_2$ will have two elements, conversion of reactant and temperature. The vector $\mathbf{a}_1$ 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, $\mathbf{a}_2$ will have two elements, and the elements will correlate the effect of processing time on each relevant product specification j, $\mathbf{ps}_{j,2}$. If the waiting time has no effect on product specifications, the vector $\mathbf{b}_i$ is null. Otherwise, the product specifications at stage 2 will clearly be affected by the product specifications at stage 1. Thus, matrix $\mathbf{C}_{2,i^*}$ will be 1×2. Its elements correlate the effects 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 requirements (product specifications). 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, $\boldsymbol{\delta}\mathbf{ps}_p^{\mathrm{o}}$. That is, only deviations up to a point will be permitted (eq 3)

$$\sum_m B_m\boldsymbol{\delta}\mathbf{ps}_m \le \boldsymbol{\delta}\mathbf{ps}_p^{\mathrm{o}}\sum_m B_m \qquad \forall p, \forall m \tag{3}$$

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.

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

$$\delta\mathrm{cost}_i = \mathbf{f}_i\boldsymbol{\delta}\mathbf{poc}_i \tag{4}$$

3.1.2. Flexibility Regions for $\mathbf{poc}_i$ and $\mathbf{ps}_i$. In eq 1, $\boldsymbol{\Delta}$ and $\boldsymbol{\sigma}$ define the flexibility regions for $\mathbf{poc}_i$ and $\mathbf{ps}_i$, respectively. The widths of these regions will basically depend on the accuracy of the model presented in section 3.1.1. 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 hyperplanes, eq 4, where n is the number of variables considered or the degree of flexibility of the batch process considered

$$\mathbf{L}_i\boldsymbol{\delta}\mathbf{poc}_i + \mathbf{l}_i'\delta\mathrm{TOP}_i + \mathbf{l}_i''\delta\mathrm{TW}_i \le \mathbf{M}_i \qquad \forall i \tag{5}$$

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

3.1.3. How the Model is Adapted to the Initial Process Conditions. Two types of deviations can be found at the beginning of a batch: deviations in the quality of the raw materials and expected deviations in some process operating conditions.

Let the vector $\mathbf{ps}_0$ correspond to the raw materials specification of a batch, so that deviations in the nominal properties of raw materials will be taken into account in this vector. Subsequent deviations in process operating conditions will be within the maximum and minimum values permitted for $\boldsymbol{\delta}\mathbf{poc}_i$ (eq 6)

$$\boldsymbol{\delta}\mathbf{poc}_i^{\min} \le \boldsymbol{\delta}\mathbf{poc}_i \le \boldsymbol{\delta}\mathbf{poc}_i^{\max} \tag{6}$$

Timing of the schedule of a given Θ;

$$TI_i \geq 0 \ \ \forall i$$

$$TF_i = TI_i + TOP_i + TW_i \ \ \forall i$$

$$\begin{cases} NIS \Rightarrow TI_i = TF_i \ \ \forall i, i'/\exists s \in \{\overline{S}_i \cap S_{i'}\} \\ UIS \Rightarrow TI_i \geq TF_i \ \ \forall i, i'/\exists s \in \{\overline{S}_i \cap S_{i'}\} \end{cases}$$

$$TW_i \leq TW_i^{max} \ \ \forall i$$

$$MS \leq TF_i \ \ \forall i$$

Flexible recipe model Ψ;

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

Flexibility Regions, $\boldsymbol{\Delta}$ and $\boldsymbol{\sigma}$;

$$\boldsymbol{L}_i\,\boldsymbol{poc}_i + \boldsymbol{l'}_i\,TOP_i + \boldsymbol{l''}_i\,TW_i \leq \boldsymbol{M}_i \ \ \forall i$$

Performance criterion Φ;

$$\sum_m B_i\,\boldsymbol{\delta ps}_i \leq \boldsymbol{\delta ps}_p^o \sum_m B_i \ \ \forall p,\ \forall m \in \{batches\ mixed\}\,\forall i$$

$$\delta Cost_i = \boldsymbol{f}_i\,\boldsymbol{\delta poc}_i \ \ \forall i$$

$$min\left(MS\,F^* + \sum_i \delta\,Cost_i\right)$$

Figure 1. Formulation for recipe initialization and multipurpose batch process schedule timing.

3.1.4. How the Model Deals with Qualitative and Quantitative Process Operating Conditions. The introduction of qualitative or decision variables into our model will result in an MILP formulation instead of the desired LP. However, the MILP formulation should be expected to have a reduced number of such variables so that the MILP solution will have a low computational cost, permitting an efficient solution following our approach.

For instance, let ps_i represent the conversion in a batchwise reaction that can be catalyzed by two different catalysts. As each catalyst has different reaction kinetics, the flexible recipe model can be represented by

$$\delta ps_i = a_i\delta TOP_i + b_i TW_i + (d_i, d_2)\begin{pmatrix}\delta poc_{i,1} \\ \delta poc_{i,2}\end{pmatrix} \quad (7)$$

where ($\delta poc_{i,1}$, $\delta poc_{i,2}$) is (1, 0) if catalyst 1 is used and (0, 1) otherwise.

3.1.5. 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 might already be scheduled, and some others might not. The problem is relaxed by solving the linear flexible recipe model proposed in section 3.1. Therefore, if a node has a relaxation higher than the best bound, the branch corresponding to that node is cut. Figure 1 shows the LP model to be solved at each node of the branch-and-bound algorithm procedure, where the objective function considers a tradeoff between the production makespan and TOP_iproduction costs. Thus, the recipe is optimized, as well as the timing of the partial schedule.

In Figure 1 is presented the formulation for recipe initialization and multipurpose batch process schedule timing, where TI_i and TF_i are the starting and ending times, respectively, of task i; $\overline{S}_i$ is the set of states generated by task i; and S_{i^*} is the set of states that feed task i*. The branch-and-bound procedure where the LP has to be integrated is shown in Appendix 1. This formulation has been implemented in C++, and the LP solved using the noncommercial linear programming code lp-solve.[12]

3.2. Motivating Example. The proposed framework for recipe initialization integrated into production scheduling was 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.[13] In that work, an input−output kind of black-box model was developed to describe the behavior of the reaction phase of the recipe. The model predicted the reaction yield, $ps_{i,1}$, as a function of the reaction temperature, $poc_{i,1}$; reaction time, TOP_i; amount of catalyst, $poc_{i,2}$; and amount of one reactant in excess, $poc_{i,3}$. Then, the model was used to optimize different recipe elements, analyzing the effects of model accuracy on the results. However, in that work, only one batch phase of the recipe was considered. In this study, the entire batch recipe train and a production environment are considered to fully exploit the potential of a more realistic batch process scenario.

The flexible recipe model, Ψ, for this reaction phase, given the linearity required by the model proposed in section 3.1. becomes

$$\boldsymbol{\delta ps}_{i,1} = 4\delta TOP_i + (4.4, 95, 95)\begin{pmatrix}\delta poc_{i,1} \\ \delta poc_{i,2} \\ \delta poc_{i,3}\end{pmatrix} \quad \forall i \in \{\text{reaction phase}\} \quad (8)$$

This model has been adapted from that presented by Keesman.[13] Then, the coefficients of eq 7 are the linear coefficients of the Keesman quadratic model. The flexibility of this batch stage, contained in regions $\boldsymbol{\Delta}$ and $\boldsymbol{\sigma}$ according to eq 1, is defined by the set of cutting planes (eq 9) that bounds the deviation of $\delta ps_{i,1}$ predicted by eq 8 and that predicted by the quadratic model.

For simplicity, it has been assumed that the hypervolume of $\mathbf{R}^4$ containing $\boldsymbol{\Delta}$ and $\boldsymbol{\sigma}$ is a hypercube. Equation 9 represents the hypercube of maximum volume that bounds the flexibility region with a tolerance of less than 1.5% for the reactant conversion.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{pmatrix}\begin{pmatrix}\delta poc_{i,1} \\ \delta poc_{i,2} \\ \delta poc_{i,3} \\ \delta TOP_i\end{pmatrix} \leq \begin{pmatrix} 0.5\ °C \\ 0.7\ °C \\ 8.5\ g \\ 27\ g \\ 7.5\ g \\ 30\ g \\ 0.1\ h \\ 0.3\ h \end{pmatrix} \quad (9)$$

This reaction stage was incorporated into the overall 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. The reaction temperature at the second stage, $\delta poc_{i,1}$, depends on the temperature reached at the first stage, $\delta ps_{i',2}$, as follows

$$\delta poc_{i,1} = \delta ps_{i',2} \quad (10)$$

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

$$\boldsymbol{\delta ps}_{i',2} = 10\delta TOP_{i'} \quad (11)$$

Table 1. Batch Production Environment

product	equipment unit processing time (h)				number of batches
P1	U1	U2	U3	U4	3
	0.5	1.75	2	0.5	
P2	U1	U3	U4	U6	1
	1	2	1.5	1	
P3	U7	U4	U6	U5	2
	2	1	1	1	
P4	U2	U3	U7	U5	1
	1.5	1	2	1.5	

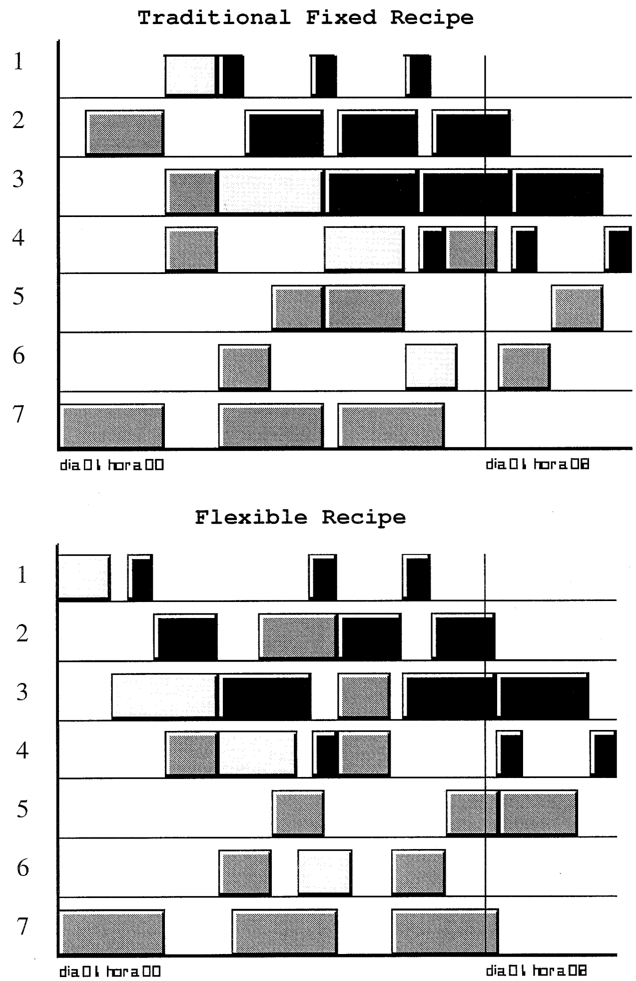

Figure 2. Optimal Gantt chart of batch production environment of Table 1 when considering the fixed recipe and the recipe adaptation. The case study recipe is represented in black.

This recipe was introduced into the production scenario given in Table 1. P1 represents the production of benzyl alcohol. The rest of the products P2, P3, and P4 share equipment units and resources with product P1.

Figure 2 shows the Gantt charts corresponding to the optimum production schedule for the proposed case study when the fixed recipe at nominal operating conditions is considered and when recipe adaptation is considered. The resulting production makespan is 10.75 h for the fixed recipe environment. When the proposed flexible recipe framework is considered, the production makespan decreases to 10.45 h (2.8% makespan reduction). Also, a different sequence of batches is obtained when the condition that the mixing of the three batches of alcohol has to meet the nominal reaction yield is imposed ($\delta \mathbf{ps}_p^{\mathrm{o}} = 0$). The optimal solution is obtained in 25,5 CPU seconds using a AMD-K7 Athlon 1-GHz computer.

Table 2. Formulation for Recipe Initialization and Multipurpose Batch Process Schedule Timing

batch	temp (°C)	processing time (h)	amount of KOH (g)	amount of H_2CO (g)	conversion (%)
1	64.5	1.2	500	425	75
2	64.5	1.2	500	425	75
3	63.8	1.2	500	425	72

Table 3. Optimal Process Operating Conditions for Three Batches of Alcohol after Limiting Reaction Temperature

batch	temp (°C)	processing time (h)	amount of KOH (g)	amount of H_2CO (g)	conversion (%)
1	63	1.55	512	438	75
2	63	1.55	512	438	71.9
3	63	1.55	512	438	71.9

The resultant process operating conditions for the three alcohol batches for the flexible recipe scenario are summarized in Table 2.

To see the effect of initial process deviations on the recipe, Keesman et al.[13] limited the reaction temperature to 63 °C ($\delta \mathrm{poc}_{i,1} = -1$). After optimizing the reaction stage alone, it was found that the reaction time had to be extended to 1.76 h, that the total amount of KOH reached 528 g, and that the amount of formaldehyde had to change to 475 g to keep the intended reaction yield. For these new nominal conditions, the resultant production makespan for the scenario described in Table 1 is 11.03 h, which means a reduction in productivity of 5.5%. Otherwise, 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 the new nominal conditions are shown in eqs 12 and 13, respectively.

$$\delta \mathrm{ps}_{i,\,1} = 3.75\delta \mathrm{TOP}_i + (101,\ 112.5)\begin{pmatrix}\delta \mathrm{poc}_{i,2}\\ \delta \mathrm{poc}_{i,3}\end{pmatrix} \quad \forall i \in \{\text{reaction phase}\} \tag{12}$$

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

Now, the optimal production makespan becomes 10.61 h. Therefore, using our proposed framework, limiting the reaction temperature to 63 °C implies only a 1.5% reduction in process productivity. The new process conditions for the different batches of the alcohol production appear in Table 3.

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 despite an eventual increase in plant productivity. Obviously, a more realistic scenario should also consider the costs associated with deviations in process operating conditions from nominal values.

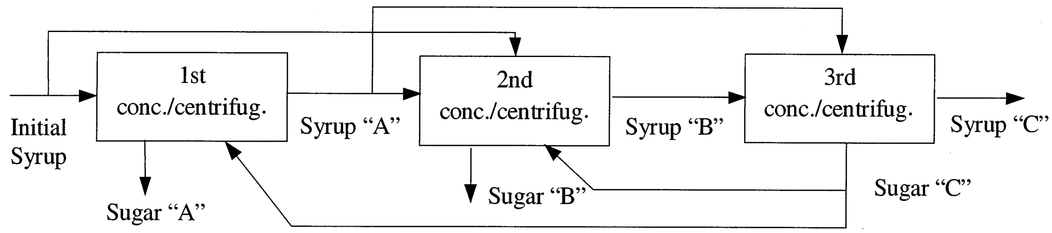

Figure 3. Raw sugar refinery process scheme.

4. Industrial Application: Sugar Cane Refinery Pans Scheduling

This section illustrates the proposed flexible recipe framework in industrial practice. The case study chosen addresses a simple scheduling problem but complex recipe behavior. To predict the recipe behavior, a first-principles model for the syrup concentration and crystallization was simulated using Aspen Custom Modeler. The interaction between this model and the scheduling algorithm was performed by means of a simulated annealing search optimization algorithm.

4.1. Process Description. The sugar cane refinery process comprises six main stages: raw sugar dissolution, syrup treatment, refinery boiling, centrifuging, sugar drying, and packaging. Refinery boiling is carried out in pans. Here, water is evaporated from thick juice, and saccharose is incorporated into sugar grains added to the pan as seed.

The boiling step is normally carried out batchwise and is the selected scenario of our study. The sugar extraction from the fresh syrup is normally carried out in three steps. In each of these steps, different types of sugar are obtained depending on the syrup purity. From the fresh syrup concentration and crystallization, sugar of quality A is obtained. The syrup obtained from the centrifuges when separating sugar A (called syrup A) still contains removable saccharose. From the syrup A concentration and crystallization (maybe mixed with some of the initial syrup), sugar of quality B is obtained. The syrup obtained when separating this sugar is again processed to obtain sugar of quality C. The last syrup might still contain a considerable amount of saccharose; however, it also contains a great amount of impurities that prevents further sugar crystallization. A scheme of this recipe is represented in Figure 3.

The bottleneck of the process is usually placed at the pan-boiling step. Therefore, a proper scheduling of the different facilities as a function of the plant resource availability is the key element to improving the performance of the entire plant.

To solve this problem, it has been imposed that, at the end of each boiling step, product specifications have to satisfy the nominal requirements, $\boldsymbol{\sigma} = \emptyset$. Therefore, different pan cycles are independent, and the process can be regarded as three one-stage recipes that produce sugars A, B, and C in turn. Then, the continuous step between each of these recipes is regarded as intermediate storage. The scheduling of pans is solved using the rule that the system should operate as many pans of A as possible and as many pans of B and C as allowed by the amount of raw material available in intermediate storage. This heuristic is based on the fact that the process bottleneck is in the pan section.

Decision variables that will define the plant schedule ($\boldsymbol{\Theta}$) are the number of pans allowed to operate simultaneously and the amount of steam supplied to each pan. The result will depend on the resource availability, $\boldsymbol{\Delta}$, on the modeling function Ψ, and on the specific scheduling algorithm Ω. The optimal result will be established by a specific performance criterion (Φ), which, in this case, will be to maximize sugar productivity.

4.2. Syrup Evaporation and Crystallization Model, $\boldsymbol{\Psi}$. In the literature can be found some dynamic mathematical models to describe saccharose crystallization in batch pans.[14] In addition, some useful data for predicting the effects of different pan process variables on the crystallization can be found in Hugot.[15]

In general, all of these models and data agree with the fact that the rate of crystallization depends on the square of the excess of saturation in the pan, the amount of sugar crystallized, and the amount of impurities in the syrup as follows

$$\text{crystallization rate} = \frac{\mathrm{dMA}_i}{\mathrm{d}t} = \frac{\mathrm{d}(B_i C_{\mathrm{s},i})}{\mathrm{d}t} = -v_0(\mathrm{MA}_i)^{2/3}(S_i - 1)^2 \mathrm{e}^{\alpha C_{\mathrm{p},i} + \beta} + C_{\mathrm{s,o}} q_i \quad (14)$$

where MA_i is the amount of sugar crystallized; B_i is the amount of syrup in the pan; $C_{\mathrm{s},i}$ and $C_{\mathrm{p},i}$ are the sugar concentration (Brix) and purity, respectively; S_i is the syrup saturation; and q_i is the syrup flow rate. Equation 15 describes the maximum pan steam consumption

$$V_i^{\max} = \frac{\mathrm{UA}_i[T(P_i) - t(p_i, C_{\mathrm{s},i})]}{\lambda[T(P_i)]} \quad (15)$$

where $V_i^{\max}$ is the maximum rate of steam consumption, $T(P_i)$ is the steam condensation temperature, $t(p_i, C_{\mathrm{s,i}})$ is the syrup boiling temperature, and λ is the steam enthalpy. Because the steam resource is a process constraint, the true pan steam consumption will depend on the steam availability, on the number of pans running simultaneously, and on the distribution of the steam among the pans as follows

$$\text{if } \sum_{\forall i} V_i X_i \le \mathrm{ST}^{\max} \quad \text{then } V_i = V^{\max}$$

$$\text{else } V_i = \frac{Y_i}{\sum_{\forall i} Y_i} \frac{\mathrm{ST}^{\max}}{\sum_{\forall i} X_i} \quad (16)$$

where V_i is the actual steam flow rate of pan i, X_i is an actual scheduling decision variable that is equal to 1 if the pan is operating at instant t and 0 otherwise, Y_i is

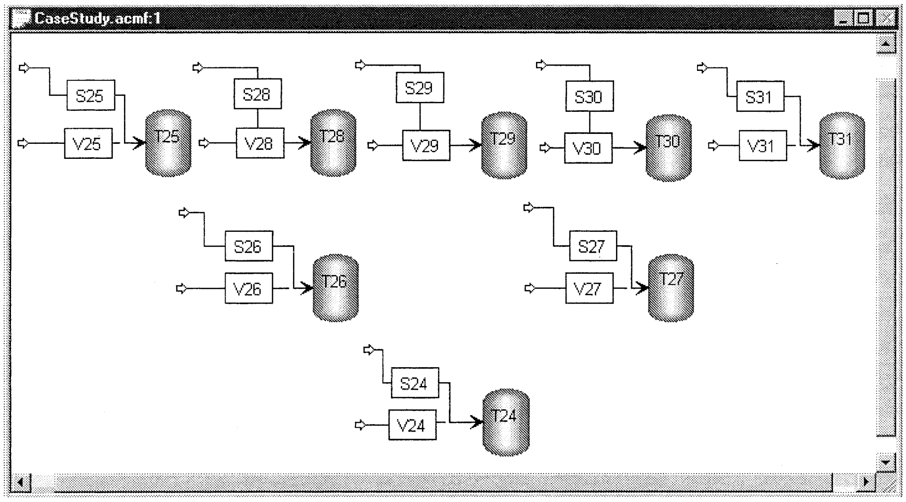

Figure 4. Process scheduling simulation environment.

a steam discretization variable that permits steam to be distributed unevenly among pans, and ST^{max} is the maximum steam availability in the refinery plant. Finally, the rate of syrup concentration is defined as follows

$$\frac{\mathrm{d}(B_i)}{\mathrm{d}t} = q_i - V_i \frac{\lambda[T(P_i)]}{\lambda[t(p_i),C_{s,i}]} \tag{17}$$

This formulation was implemented in Aspen Custom Modeler, version 10.1,[16] and the different parameters were adjusted with real plant data.

4.3. The Short-Term Scheduling Algorithm, Ω. Different possible plant configurations, involving different numbers of pans running simultaneously, were evaluated as a function of the sharing of resources among pans. To solve for the steam distribution in practice, it was assumed that a pan can receive 16% more or less steam than the average. A pan receiving more steam than the average implies that an amount of the steam flow was taken from one type of pan to another. This steam discretization was tested to be optimal to maximize the inherent flexibility of this case study.

To schedule different scenarios using the available equipment units and resources, an online scheduling algorithm was designed. The motivation for online scheduling is that uncertainty plays a key role in this kind of process, as processing time is an uncertain variable. In Appendix 2, this algorithm is described in detail. Each time a pan finishes processing, the algorithm decides which pan will be the next to run. Specifically, it runs as many A pans as permitted according to the maximum number of pans allowed to run simultaneously and as many B and C pans as possible depending on the amount of raw material available in the intermediate storage. In practice, for a given plant configuration (described by Max_number_pans_allowed and **Y**), each pan is run until its end conditions are reached. Each time a pan finishes, a new pan enters into operation according to the scheduling algorithm described in Appendix 2.

4.4. Case Study Results. Our case study involves five A pans (T25, T28, T29, T30, and T31), two B pans (T26 and T27) and one C pan (T24). The nominal batch size of A and B pans is 50 tons, and that of the C pan is 60 tons. The steam resource availability (ST^{max}) is assumed to be 45 tons/h, which is only enough to keep four pans working simultaneously at their maximum steam consumption. Resources might not be distributed evenly. Therefore, the case study presents 475 possible plant configurations.

Figure 4 shows the simulation environment. The solution of the flexible recipe model requires substantial computing time (simulating 24 h of production takes approximately 180 s on an Athlon 1-GHz computer); therefore, a simulated annealing (SA) optimization algorithm was used to search for an optimal configuration by modifying the number of pans allowed to run simultaneously and the steam distribution.

When no more than four pans are working simultaneously and no resource limitations are considered, the sugar production is 710 tons/day. In this case, the A-pan processing time equals 2.4 h. If more pans are allowed to run simultaneously, the steam availability per pan decreases, and so, the maximum steam demand cannot be supplied. Consequently, the processing time increases. If six pans running simultaneously are allowed, a total productivity of 820 tons/day is achieved. If eight pans are allowed, then the A-pan processing time varies from 3.2 to 3.7 h, depending on the actual number of pans running at each moment, as it is not possible to have eight pans working simultaneously because of the intermediate storage limitation. In this case, the productivity is 845 tons/day.

After 94 iterations of the SA algorithm, the best plant configuration found is to run eight pans simultaneously whenever possible with two A pans receiving 16% more steam than the average and one B and one C pan receiving 16% less steam. The optimal Gantt chart is represented in Figure 5. The productivity in this case is 850 tons/day.

5. Conclusions

This work presents a framework that includes the possibility of recipe adaptation in the optimization of batch processes. Specifically, a methodology is described for optimizing the production scheduling of complex batch processes where the recipes have some degree of flexibility. The proposed methodology represents the incorporation of one more level of detail (process operat-

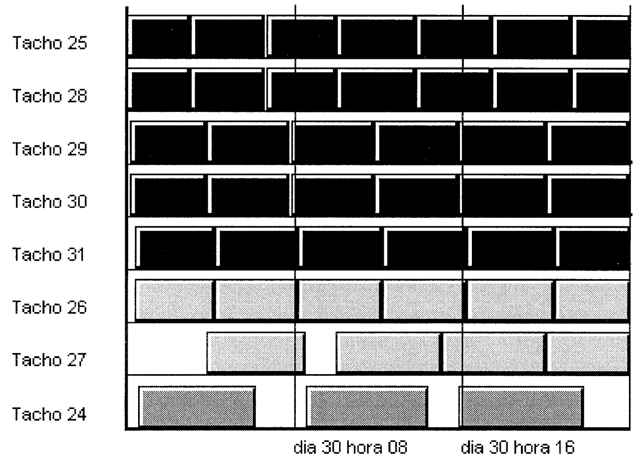

Figure 5. Optimal Gantt chart of sugar cane pans case study.

ing conditions and product specifications) during the scheduling procedure.

First, a conceptual analytical framework is presented, which, in its linear form, leads to the solution of quite reasonable and useful MILP scheduling problems, showing the potential use of this framework. The results obtained from a case study show that, within an acceptable computational time, the use of the flexible recipe in plant scheduling optimization leads to better performance of batch processes. In its practical implementation, the presented framework integrates an equation-oriented process simulator with a simple heuristic scheduling algorithm. The advantages of considering the flexible recipe in an industrial case study scenario are also discussed. It is concluded that the problem of the optimum management of limited resources in the sugar cane refinery industry is not simply a scheduling problem but rather an issue of the integration of simultaneous recipe optimization into the scheduling procedure.

This paper proposes a new philosophy for recipe management in batch process industries, and the conceptual framework presented is demonstrated to be a valuable tool that is able to handle a number of different scenarios.

Further work underway focuses on extending the present flexible recipe framework to real-time optimization environments. For this purpose, an on-line model for recipe adjustment will be necessary. This model will receive on-line information from process variables and will have to interact with a rescheduling tool, generating a *corrected control recipe* in front of deviations during each batch run. A process state assessment module for evaluation of off-normal diagnosis will decide when different actions should be taken.

Nomenclature of the Basic Framework

Φ = performance criterion function of the flexible recipe model
Ψ = quality and production cost modeling function
Ω = scheduling constraints
$\boldsymbol{\Theta}$ = specific production plan
$\boldsymbol{\Delta}$ = flexibility region for process operating conditions
$\boldsymbol{\sigma}$ = flexibility region for product specifications
$\mathcal{A}$ = assignment of different batch plant resources
$\mathcal{P}$ = set of production orders
$\mathcal{J}$ = set of equipment units
$\mathcal{PT}$ = set of production horizons or due dates
$\mathcal{R}$ = set of process resources
$\mathcal{S}$ = sequence of different batches
$\mathcal{T}$ = multistage flowshop or jobshop batch plant topology
$\mathbf{ps}_i$ = product specification vector at the end of each batch process stage i
$\mathbf{ps}_{i*}$ = product specification vector at the end of stage $i*$ where the different inputs to stage i are produced
$\mathrm{ps}_p^{\mathrm{o}}$ = product specifications vector for each product p required at the end of the production horizon
$\mathbf{poc}_i$ = process operating conditions vector of stage i
TOP_i = processing time of each stage i of a recipe
TW_i = waiting time at stage i

Appendix 1

The S-graph solution uses a graphical representation and algorithms to solve scheduling problems.

Mathematical Formulation of the S-Graph Method. In an S-graph, two classes of arcs, so-called recipe arcs and schedule arcs, are specified. Therefore, an S-graph is given in the form of $G(N,A_1,A_2)$, where N, A_1, and A_2 denote the sets of nodes, recipe arcs, and schedule arcs, respectively. A nonnegative value, $c(n_i,n_j)$, is assigned to each arc (n_i,n_j) that denotes the weight of that arc. In practice, if an arc is established from node n_i to node n_j, then the task corresponding to node n_j cannot start its activity before the task corresponding to node n_i has started. Specific types of S-graphs are identified for a recipe (i.e., recipe graph) and for a schedule of all tasks (i.e., schedule graph).

Recipe Graph. A recipe defines the order and type of tasks, the material transfers among them, and the set of plausible equipment units for each task. This type of information is represented by the recipe graph of a recipe.

Let one node be assigned to each task (task node) and one to each product (product node). An arc is established between the nodes of consecutive tasks and from the nodes of tasks generating products to the corresponding product node with the weights specified by the processing times of the tasks.

Schedule Graph. A specific S-graph, termed the schedule graph, is introduced to describe a single solution of a scheduling problem; there exists one schedule graph for each feasible schedule of the problem. S-graph $G'(N,A_1,A_2)$ is called a schedule graph of recipe graph $G(N,A_1,\varnothing)$ if all tasks represented in the recipe graph have been scheduled by taking into account equipment−task assignments. By an appropriate search strategy, the schedule graph of the optimal schedule can be effectively generated, as will be shown later.

The formal definition of a schedule graph and the axioms that $G'(N,A_1,A_2)$ must satisfy are given at Sanmartí et al.[10]

Therefore, this representation searches for the optimal schedule generating partial schedules (schedule graphs) from the batch process recipe (recipe graph). Each partial schedule is generated adding one schedule arc to the preceding schedule graph. For instance, for the nonintermediate storage (NIS) scheduling policy, let τ_j denote the set of tasks that follow task j according to the recipe. If equipment unit E_i is assigned to task j and, after completion of j, to task k, then, a zero-weighted arc (or an arc whose weight is equal to the length of the changeover time if applicable) is established from each element of τ_j to k.

From this process, a branch-and-bound algorithm is generated. The root node of the algorithm corresponds to the recipe graph. From this root, nodes of the tree (partial schedules) are built by adding one schedule arc to the preceding node. At each node, an LP (in our case, the flexible recipe model) is solved to calculate the relaxation of the algorithm. The solution of this LP at the end of the tree (leaf) gives the optimal timing of the schedule (and recipe elements) associated with this leaf. If, at any node, the bound is higher than the best objective function achieved up to this moment, the branch of study is cut. The optimal schedule will correspond to the leaf with the best objective function value. The main procedure, the branching procedure, and the bounding procedure of the branch-and-bound scheduling algorithm are as follows:

```
procedure main
comments:
    EQ: currently equipment unit being explored.
    SOUN: set of nodes to be scheduled of equipment unit EQ.
    UNEX: set of equipment units to be explored.
    last: last scheduled node.
begin
    SET = ∅; UNEX = Set of equipment units; SOUN = ∅;
    last = 0; EQ = 0; bound = 0;
    Put(G, EQ, SOUN, UNEX, last, bound) into SET;
    while SET ≠ ∅ do
        Select and remove element G from SET;
        Branching(G);
    return
end

procedure branching (G(N, A), EQ, SOUN, UNEX, last, bound)
begin
    if SOUN = ∅ then
        let EQ be an equipment unit ∈ UNEX;
        Last := 0;
        UNEX := UNEX \ {EQ};
        let SOUN be the tasks assigned to EQ;
    end
    for all n ∈ SOUN do
        if last = 0 then
            let last be an element of SOUN;
            SOUN' = SOUN \ {n};
            Put(G, EQ, SOUN, UNEX, Last, bound) into SET;
        else
            G'(N, A) = extend_graph(G(N, A), n, last);
            SOUN' = SOUN \ {n};
            last' = n;
        end
        if |SOUN'| = 1 then
            let m be the element of SOUN;
            G'(N, A) = extend_graph(G'(N, A), m, last');
            SOUN' = ∅;
            last' = 0;
        end
        bound = bounding(G'(N, A));
        if bound < current_best then
            if SOUN' = ∅ and UNEX = ∅ then
                update current_best and solution;
                delete all elements of SET with bound > current_best;
            else
                Put(G', EQ, SOUN', UNEX, last', bound) into SET;
            end
        end
    return
end

procedure bounding (G(N, A))
begin
    for all s ∈ N_p do cycle_search;
    if no cycle then
        bound = Solve_LP_Schedule_Timing_
                Considering_Flexible_Recipe(G(N, A));
    else
        bound = cycle;
    return bound
end
```

Appendix 2

The on-line scheduling algorithm of pans determines which pan is going to be next to run when a pan finishes. It runs as many A pans as permitted according to the maximum number of pans allowed to run simultaneously and as many B and C pans as allowed by the amount of raw material available in the intermediate storage units as follows

```
procedure Ω(Max_number_pans_allowed, Y)
comments
    let X_i be the actual variable equal to 1 if a pan is working at an instant t
        and equal to 0 otherwise;
    let n_i be the number of times that pan i has already worked on the plan;
    let ISB and ISC be the intermediate storage levels of syrup 'B' and 'C';
    let J_A, J_B and J_C be the set of pans that produce syrup A, B and C;
begin
    if (ISB ≥ ISC) then
        if (ISB ≥ B_i) and X_i = 0, i ∈ J_B then
            X_k = 1, k| min(n_i);
        else
            if (ISC ≥ B_i) and X_i = 0, i ∈ J_C then
                X_k = 1, k| min(n_i);
            end;
        end
    else
        if (ISC ≥ B_i) and X_i = 0, i ∈ J_C then
            X_k = 1, k| min(n_i);
        end
    end
    while Σ_∀i X_i < Max_number_pans_allowed, i ∈ J_A ∪ J_B ∪ J_C
        if X_i = 0, i ∈ J_A then
            X_k = 1, k| min(n_i);
        end
    wend
end
```

Literature Cited

(1) ANSI/ISA-S88.01-1995 *Batch Control. Part 1: Models and Terminology*; American National Standards Institute: Washington, DC, 1995.

(2) Box, G. E. P.; Draper, N. R. *Evolutionary Operation*; John Wiley & Sons: New York, 1969.

(3) Rijnsdorp, J. E. *Integrated Process Control and Automation*; Elsevier: Amsterdam, 1991.

(4) Verwater, Z.; Keesman, K. J. Computer-aided development of flexible batch production recipes. *Prod.Planning Control* **1995**, *6*, 320.

(5) Verwater, Z. A Practical Approach to Recipe Improvement and Optimization in the Batch Processing Industry. Ph.D. Dissertation, Eindhoven Technische Universiteit, Eindhoven, The Netherlands, 1997.

(6) Graells, M.; Loberg, E.; Delgado, A.; Font, E.; Puigjaner, L. Batch production scheduling with flexible recipes: The single product case. In *Foundations of Computer Aided Process Operations*; AIChE Symposium Series No. 320; American Institute of Chemical Engineers (AIChE): New York, 1998; Vol. 94, p 286.

(7) Reklaitis, G. V. Perspectives on scheduling and planning of process operations. In *Proceedings of the Fourth IEEE International Symposium on Process System Engineering*; IEEE Press: Piscataway, NJ, 1991.

(8) Shah, N. Single and multisite planning and scheduling: Current status and future challenges. In *Foundations of Computer Aided Process Operations*; AIChE Symposium Series No. 320; American Institute of Chemical Engineers (AIChE): New York, 1998; Vol. 94, p 91.

(9) Grossmann, I. E.; Pinto, J. M. Assignment and sequencing models of the scheduling of process systems. *Ann. Oper. Res.* **1998**, *81*, 433.

(10) Sanmartí, E.; Holczinger, T.; Puigjaner, L.; Friedler, F. Combinatorial framework for effective scheduling of multipurpose batch plants. *AIChE J.* **2002**, *48*, 2557.

(11) Romero, J.; Friedler, F.; Puigjaner, L. Scheduling multipurpose batch processes considering the flexible recipe. In *Proceedings* of *PRES'01*; AIDIC: Milan, Italy, 2001; p 350.

(12) Berkelaar, M. *lp_solve 2.0* (software package); Eindhoven University of Technology: Eindhoven, The Netherlands, 1995 (available online at ftp:// ftp.es.ele.tue.nl/pub/lp_solve).

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

(14) Hing, R.; Nicado, M.; Navarro, A. Applicacion de métodos numéricos de control óptimo al modelo matemático del proceso de cocción en tachos azucareros. *Cent. Azúcar* **1994**, *21*, 2, 3.

(15) Hugot, E. *Handbook of Cane Sugar Engineering*; Elsevier Science: Amsterdam, 1986.

(16) AspenTech. *Aspen Custom Modeler v10.1: Reference Manual*; Aspen Technology, Inc.: Cambridge, MA, 1998.

Received for review January 2, 2002
Revised manuscript received September 2, 2002
Accepted October 18, 2002

IE020002N