Optimal Reactive Scheduling of Manufacturing Plants with Flexible Batch Recipes

Sergio Ferrer-Nadal,† Carlos A. Méndez,‡ Moisès Graells,† and Luis Puigjaner*,†

Chemical Engineering Department−CEPIMA, Universitat Politècnica de Catalunya ETSEIB, Av. Diagonal 647, E-08028, Barcelona, Spain, and INTEC (Universidad Nacional del Litoral−CONICET), Güemes 3450, 3000 Santa Fe, Argentina

The inherent dynamic nature of industrial environments often needs not only the cyclic revision of scheduling decisions (typical rescheduling actions) but also an efficient adjustment of the production recipe to the current process conditions. Therefore, the concept of flexible recipe becomes an important part of the rescheduling framework that allows full exploitation of the process flexibility in batch plants. This work introduces a rigorous mathematical approach that incorporates the concept of recipe flexibility to plantwide batch operation rescheduling. The proposed mixed-integer linear programming (MILP)-based approach is able to address the rescheduling problem of multistage, multipurpose batch plants involving different storage policies, nonzero transfer times, and flexible batch product recipes. This model relies on the concept of general precedence, which reduces the number of binary variables and, therefore, the computational effort. Flexible-recipe constraints are incorporated in this model to account for the possibility of changing the processing time of some tasks, tweaking the rest of the parameters of the product recipe. The cost for modifying these process variables from their optimal economic conditions is taken into account to represent how productivity is increased despite the cost of altering the nominal plant conditions. Different incidences, such as insertion of new orders, equipment failures, due-date changes, maintenance tasks, and delay in arrivals, variations in the cost, and quality of the raw materials or products, taking place throughout the scheduling horizon, are considered to evaluate the effectiveness of the proposed approach.

1. Introduction

Over the last few years, the chemical industry has been evolving into a globally competitive and demand-driven mode of operation where companies are required to respond quickly to a constantly changing market situation with stricter product specifications. Process operations optimization can effectively increase plant profitability after the safety and profit quality objectives have been satisfied.[1−3] Within this context, batch processes have received great attention over the last few years because of their higher flexibility compared to continuous processes and the increasing demand for specialty, high-added-value chemical and pharmaceutical products. The most flexible and attractive batch operations usually take place in the so-called multipurpose batch plants, where a wide variety of products with different recipes can be produced by sharing a limited number of processing units capable of performing different batch operations. In these manufacturing environments, the scheduling and rescheduling activities acquire a special importance because of the higher complexity associated to the higher process flexibility. A comprehensive state-of-the-art review of optimization techniques for short-term scheduling can be found in the paper by Méndez et al.[4] Most of the scheduling approaches assume that batch processes are operated at nominal conditions following predefined production recipes. However, if we consider the dynamic environment of the process industry, in many cases these recipes need to be frequently updated. In the face of, for example, variations in the raw materials, availability of the plant equipment units, or changes in the product specifications, a flexible-recipe operation may result in a more suitable way of incorporating systematic recipe adaptations depending on the actual process conditions.

The flexible-recipe concept was originally introduced by Rijnsdorp[5] as a set of adaptable elements that controls the process output. For instance, it might be possible to increase the amount of catalyst against a poorer feed quality or increase the rate of heating if shorter processing times were required to accelerate the production. Afterward, Verwater-Lukszo[6] presented a flexible-recipe approach for the adjustment of control recipes during production. This author proposed a statistical analysis from available process data or, more specifically, making the process work under a range of conditions around the nominal settings, leading to valid black-box models. Several studies have applied these basic concepts for the quality control in the batch production. Sel et al.,[7] for example, successfully tested the flexible-recipe concept for an industrial process in TiO_2 pigment production. Rutten and Bertrand[8] studied the performance of three production planning procedures for the use of flexible recipes to cope with occasional shortages in raw materials. One of the first attempts to extend the flexible-recipe approach to a plantwide scheduling problem was carried out by Romero et al.[9] These authors proposed to integrate a linear flexible-recipe model into a multipurpose batch process scheduling model based on a graph-oriented method.

Nevertheless, in addition to changes in nominal process conditions, frequent unexpected events can also take place during the normal batch-plant operation (equipment failures, late order arrivals, order cancellations, and so on). These unforeseen changes may lead the in-progress schedule to become suboptimal or even infeasible. As a result, any schedule will need frequent revision, and the scheduler's ability to react to any unexpected events becomes a central issue during the batch-plant operation. Two different approaches have been typically proposed in the literature to deal with the uncertainty in batch-process operations. On the one hand, a first approach considers that, if the presence of uncertainty can be characterized at the time of scheduling, it might be advantageous to take possible

* To whom correspondence should be addressed. Tel.: +34-93-401-6678. Fax: +34-93-401-0979. E-mail: luis.puigjaner@upc.edu.
† Universitat Politècnica de Catalunya ETSEIB.
‡ INTEC (Universidad Nacional del Litoral−CONICET).

10.1021/ie061255+ CCC: $37.00
Published on Web 08/10/2007

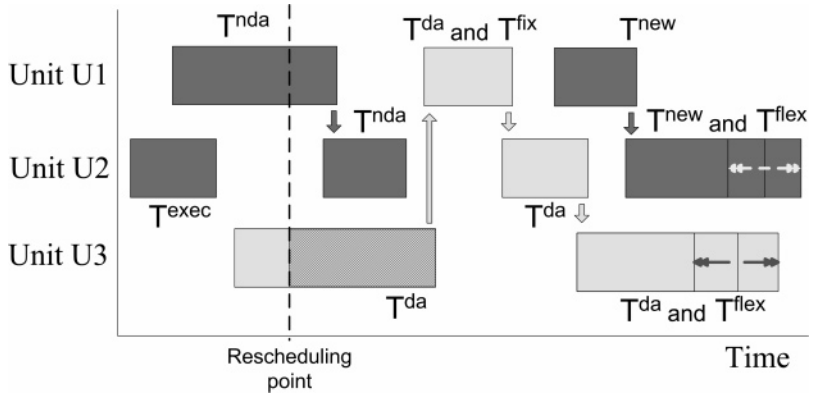

Figure 1. Basic representation of task types: T^{exec}, T^{nda}, T^{da}, T^{new}, and T^{flex}.

Table 1. Allowable Rescheduling Actions (X = No, √ = Yes)

task type	(re)alloc.	(re)seq.	(re)timing	recipe adjustment (only for T^{flex})
1. executed tasks, T^{exec}	X	X	X	X
2. nondirectly affected tasks, T^{nda}	X	√	√	√
3. directly affected tasks, T^{da}	√	√	√	√
4. new tasks, T^{new}	√	√	√	√

future events into consideration before they happen in order to minimize their eventual impact.[10−13] These techniques are usually classified into the so-called proactive scheduling strategies. In spite of the more-robust schedules that can be generated, proactive scheduling techniques cannot ensure optimality or even feasibility along the whole scheduling horizon. Consequently, companies are not usually determined to accept this loss of benefits as a result of working at suboptimal conditions. On the other hand, rescheduling systems aim to modify the original schedule during the manufacturing process in order to periodically react to changes in the production environment and try to minimize their effects on the plant operation. While the original schedule is usually generated at the beginning of the scheduling horizon in a deterministic manner, the rescheduling strategy handles uncertainty by performing proper adjustment to the schedule in-progress.

Although rescheduling techniques have a central role in process operations, only a few developments have focused their attention on this challenging problem. Probably one of the first approaches to this problem was carried out by Cott and Macchietto,[14] who considered uncertainty in processing time and proposed a time-shifting algorithm to minimize these effects. Kanakamedala et al.[15] considered also unit unavailability and proposed a heuristic algorithm based on the search-tree analysis, emphasizing the importance of maintaining the original schedule. Sanmartí et al.[16] developed a set of heuristic rules to deal with uncertain processing-unit failures. These authors proposed a rescheduling strategy based on modifying starting times and proposing alternative units. Later on, Vin and Ierapetritou[17] presented a two-stage solution procedure based on a continuous-time mixed-integer linear programming (MILP) approach. This procedure aimed at reacting against machine breakdown and rush order arrival in multiproduct batch plants without considering any heuristic rule. Méndez and Cerdá[18] developed a MILP model for reactive scheduling of single-stage batch plants, and later these authors extended the original approach to multistage, multiproduct batch plants with limited discrete resources.[19] All these rescheduling approaches allow performing certain corrective actions such as partial resource reallocation and tasks resequencing and retiming assuming a fixed batch production recipe.

The aim of this work is to introduce an optimization tool for the reactive batch operation scheduling. This tool extends the rescheduling process to multipurpose batch plants and, at the same time, incorporates the concept of recipe flexibility as an additional rescheduling action to fully exploit the inherent flexibility of batch processes. To address this problem, a mixed-integer linear programming (MILP) model is proposed. This paper is organized as follow. The general problem definition and the rescheduling strategy are first outlined. Then, Section 3 describes the mathematical model along with the major assumptions considered. The effectiveness of the proposed approach as a decision-making tool is demonstrated through its application to several scenarios of a scheduling case study presented in Section 4. And finally, the main contributions and future work related to this contribution are outlined in Section 5.

2. Problem Definition

The typical scheduling techniques generate a priori production schedules assuming known and stationary operating conditions and demands along the entire time horizon. However, the dynamic production environment and the natural uncertainty present in the production data force the rescheduling system to frequently revise and update the schedule in progress. This mechanism could be executed periodically (daily or hourly) or extraordinarily because of the occurrence of unforeseen events or external factors. Thus, the main goal here is to optimize the schedule in progress by creating a modified production schedule that more accurately reflects the current state of the production environment. For practical limitations, every time that the rescheduling is carried out at the rescheduling point, the system should be able to generate updated schedules in an efficient and relatively fast way in view of the fact that short reaction times are always required in practice. Besides, minimum changes to the original schedule are desired to maintain a smooth plant operation. Therefore, rather than reoptimizing all the remaining tasks (full-scale rescheduling), only local and partial changes are allowed, reducing in this way the impact upon the schedule in progress. This means that a full-scale rescheduling is avoided in most of the cases.

Schematically, in this work, we have defined different sets of tasks at the rescheduling point in order to clearly identify the local actions that will be permitted for each type of batch task. These different sets are as follows:

(a) *Executed tasks* (T^{exec}) already processed at the rescheduling point, which are not included in the formulation since they are past events with no influence on the remaining schedule.

Table 2. Process Data for the Case Study

	product P1		product P2		product P3		product P4		product P5	
stage	unit	npt, h	unit	npt, h	unit	npt, h	unit	npt, h	unit	npt, h
1	U1	0.5	U1	1	U7	2	U2	1.5	U8	2
	(flexible stage)		U2	0.75	U9	2.5	U8	2		
			U8	2						
2	U2	1.75	U3	2	U3	2	U3	1	U5	1
	(flexible stage)		U9	2	U4	1	U9	2.5		
3	U3	2	U4	1.5	U6	1	U7	2	U2	1.25
	U9	1.75	U7	2			U4	2		
4	U4	0.5	U6	1	U5	1	U5	1.5	U1	1
	U7	0.75								
				due dates, h						
order I1		10	10		9		10		15	
order I2		10	20[a]		9		22[a]		15	
order I3		10							15	
order I4		15							20[a]	
order I5		19[a]								
order I6		19[a]								
pto, m.u.		5	6		7		2		5	

[a] New orders.

(b) *Nondirectly affected tasks* (T^{nda}) by the unexpected event, which are either being processed or waiting for processing at the rescheduling point.

(c) *Directly affected tasks* (T^{da}) by the unexpected event. They include batch tasks running at the rescheduling point that are rejected because of the event and need to be transferred to an alternative unit in order to be reprocessed. This group also includes successive stages of these rejected tasks in the processing sequence.

(d) *New tasks* (T^{new}) associated to late order arrivals that need to be inserted into the current schedule.

Alternatively, processing stages have also been classified according to the type of recipe as

(e) *Fixed recipe stages* (T^{fix}), which are processing stages that can only be performed at nominal operating conditions.

(f) *Flexible recipe stages* (T^{flex}), which are enclosed tasks that allow for changing their recipe parameters in a flexible manner. Flexibility in the recipe is represented in this work as a mathematical process model that relates the process outputs with the recipe items. Then, the batch tasks related to flexible stages may vary their processing times or other recipe parameters in order to optimize the global performance of the plant.

All these different tasks are depicted in the Gantt chart shown in Figure 1. In this figure, a production scenario is illustrated where several orders of two different products (dark gray and light gray) are manufactured. At the rescheduling point, because of the occurrence of an unforeseen event, the first stage of the light-gray product has to wait to be transferred and reprocessed in an alternative unit, in this case, unit U1. In addition, a late rush order of a dark-gray product has to be manufactured. Allowed partial rescheduling actions in each group of tasks are summarized in Table 1. According to this table, directly affected tasks (T^{da}) may require performing reallocation, resequencing, and retiming actions until the originally allocated processing unit will be again available. Moreover, batch tasks related to new orders (T^{new}) also need to be simultaneously inserted in the updated schedule. Tasks nondirectly affected (T^{nda}) by the incidence are just allowed to modify their starting and ending times or partially change their position in the unit processing waiting line (resequencing). In this case, reallocation actions are not allowed for this group of tasks. Moreover, tasks following flexible recipes (T^{flex}) are allowed to modify the current product recipe as an additional rescheduling action.

In contrast to the classical strategy of optimizing unit operation conditions individually, here, suboptimal individual unit operation conditions can be implemented in the plant whenever they can help to improve the global plant performance under consideration of the actual situation of the plant. Therefore, this will be a complex decision-making process posing a tradeoff between working under suboptimal single unit operation conditions and obtaining a better performance in the global plantwide operation. Thus, a detailed description of the mathematical formulation used for this novel reactive scheduling framework using flexible recipes presented in this work is described in the next section.

3. Mathematical Model

The underlying structure of the proposed mathematical model relies both on a continuous-time representation and the notion of general precedence originally introduced by Méndez et al.[20] The generalized precedence notion extends the immediate predecessor concept to consider all batches belonging to the same processing sequence. This model handles allocation and sequencing decisions through different sets of binary variables, which are defined as follows. Y_{pisu} is a binary variable equal to one whenever task (p, i, s), that is the sth stage for manufacturing the ith batch of product p and is allocated to equipment unit u. Regarding the sequencing decisions, $X_{pis,p'i's'}$ is a binary variable that establishes the general precedence relationship between a pair of tasks (p,i,s) and (p',i',s') executed at the same processing unit (otherwise $X_{pis,p'i's'}$ is meaningless). If $X_{pis,p'i's'}$ is equal to 1, task (p,i,s) is a direct or nondirect predecessor of task (p',i',s') on the waiting line for allocated unit. Alternatively, in the case where task (p',i',s') is processed before task (p,i,s) in the same unit, $X_{pis,p'i's'}$ takes the value zero. It is worth noting that the six subscripts defined for sequencing variables are needed to deal with the general rescheduling problem arising in multipurpose batch plants, where the same equipment unit can perform several operations related to the same or different products. Consequently, the sequencing variable can distinguish not only the batches and the products involved but also the stages that are being sequenced. Although the number of binary variables seems to be very large at first sight, it should be noted that sequencing variables are only defined for every pair of tasks (p,i,s) and (p',i',s') that can be performed in the same unit, which is an intrinsic characteristic of multipurpose equipment. If the general proposed rescheduling method is applied to a multiproduct batch plant, the subscripts related to the stages in the sequencing variables are no longer required.

Table 3. Recipe Parameters, Flexibility Region, and Cost for Deviation from Nominal Conditions

flexible recipe item	flexible recipe variable	flexibility region		
		fplb_{psf}	fpub_{psf}	dcost_{psf}
δ_{DPS}	reaction yield	0	0	-[a]
δ_{DTEMP}	reaction temperature	−0.7 °C	0.5 °C	3 m.u./°C
δ_{DTOP}	reaction duration	−0.3 h	0.1 h	2 m.u./ h
δ_{DKOH}	amount of KOH	−2.7 g	8.5 g	5 m.u./g
δ_{DFOR}	amount of formaldehyde	−30 g	7.5 g	4 m.u./g

[a] No deviation is allowed in quality.

Furthermore, the model is able to achieve a great saving of binary variables by (a) defining a single sequencing variable for each pair of tasks, (b) using a unique binary variable to sequence both processing and storage units, and (c) predefining the sequence of multiple batches of the same product. All these features can be exploited without compromising the global optimality of the solution.

This model considers the following general assumptions:

(1) Model parameters all are deterministic.

(2) Batch splitting or mixing is not allowed.

(3) Allowable rescheduling actions are predefined for each group of tasks as reported in Table 1.

(4) No resource constraints except equipment are considered.

(5) Linear flexible recipe models are used.

(6) A predefined flexibility region around nominal operating conditions is used.

3.1. Model Constraints. 3.1.1. Processing Unit Allocation Constraints. The following constraint enforces the assignment of a single processing unit $u \in U_{ps}$ to every task (p,i,s).

$$\sum_{u \in U_{ps}} Y_{pisu} = 1 \quad \forall (p,i,s) \in (T^{\mathrm{new}} \cup T^{\mathrm{da}}) \tag{1}$$

This constraint allocates the corresponding processing unit u to (a) tasks of new orders (T^{new}) that need to be inserted into the current schedule and (b) directly affected tasks (T^{da}) that, after the occurrence of the unforeseen event, need to be reallocated to another unit. The directly affected tasks that were being processed at the rescheduling point will need to be transferred to an alternative unit in order to be processed again.

The generality of this formulation, which can be used not only for rescheduling purposes but also for solving the short-term scheduling arising at the beginning of the time horizon of interest, should be noted. This can be done by including all the tasks into the set of new tasks (T^{new}).

3.1.2. Flexible-Recipe Model. A linear flexible-recipe model relates the deviation of process outputs to the deviation of the main flexible-recipe items. Let δ_{pisf} be the deviation of the recipe item f in a task (p,i,s) following a flexible recipe. The linear model presented in eq 2 is an approximation of the real nonlinear model that is valid in the region of the nominal process conditions, while additional auxiliary models may be required for neighboring operating conditions:

$$\sum_{f \in \mathrm{FP}_{ps}} \mathrm{lfmod}_{psf} \delta_{pisf} = 0 \quad \forall (p,i,s) \in (T^{\mathrm{new}} \cup T^{\mathrm{da}} \cup T^{\mathrm{nda}}) \cap T^{\mathrm{flex}} \tag{2}$$

3.1.3. Recipe-Flexibility Region. The above linear flexible model should be valid around the given nominal conditions representing the recipe-flexibility region. Constraint 3 establishes the maximum negative (fplb_{psf}) and positive (fpub_{psf}) deviations allowed for each recipe item.

$$\mathrm{fplb}_{psf} \le \delta_{pisf} \le \mathrm{fpub}_{psf} \quad \forall (p,i,s) \in (T^{\mathrm{new}} \cup T^{\mathrm{da}} \cup T^{\mathrm{nda}}) \cap T^{\mathrm{flex}}, f \in \mathrm{FP}_{ps} \tag{3}$$

3.1.4. Associated Cost to Deviation from Nominal Conditions. Constraint 4 computes the deviation cost for every recipe item at each task (p,i,s) following a flexible recipe. Both positive and negative deviations from an optimal operating condition are always penalized by a cost factor (dcost_{psf}). This constraint will be always active as the deviation cost (DC_{pis}) is minimized in the objective function. If the task is performed at the nominal operating condition, the deviation cost will be equal to zero.

$$\mathrm{DC}_{pis} = \sum_{f \in \mathrm{FP}} \mathrm{dcost}_{psf} \delta_{pisf} \quad \forall (p,i,s) \in (T^{\mathrm{new}} \cup T^{\mathrm{da}} \cup T^{\mathrm{nda}}) \cap T^{\mathrm{flex}} \tag{4}$$

3.1.5. Lower Bound on the Starting Time of Task (*p,i,s*). A task can only be processed if both the corresponding unit u is ready and the task (p,i,s) is also prepared to be executed. For the rescheduling problem, the ready time of a unit u (ru_u) is the completion time of the nondirectly affected tasks (T^{nda}) that were running in the schedule in progress at the rescheduling point or simply the rescheduling point if that unit was idle at that time. The ready time of a unit also can be employed in this formulation to express the period while a unit is not available for any reason. The ready time (ro_{pis}) for new tasks $(p,i,s) \in T^{\mathrm{new}}$ is the time for these late order arrivals.

$$\mathrm{ST}_{pis} \ge \sum_{u \in U_{ps}} \max(\mathrm{ru}_u, \mathrm{ro}_{pis}) Y_{pisu} \quad \forall (p,i,s) \in (T^{\mathrm{new}} \cup T^{\mathrm{da}} \cup T^{\mathrm{nda}}) \tag{5}$$

The starting time ST_{pis} for running tasks $(p,i,s) \in T^{\mathrm{nda}}$ is not a variable in the rescheduling formulation because it corresponds to a time prior to the rescheduling point. Thus, it becomes a fixed parameter rather than a variable of the model.

3.1.6. Duration of a Task (*p,i,s*). The following constraints establish the duration of a task, taking into account processing times, unit setup, and transfer times. These are not applicable for the first stage of products. In order to consider the duration of a task, a distinction has to be made between stages with or without flexible recipes.

3.1.6.a. For Stages with Fixed Recipe. These constraints express that the finishing time of task (p,i,s) can be computed based on (a) the starting time of the task (ST_{pis}), (b) the unit setup time for product p (ust_{pu}), (c) the transfer time depending on the unit used in the previous stage ($\mathrm{tt}_{pu'}$), (d) the nominal processing time for this task in this unit (npt_{psu}), (e) a possible waiting time (WT_{pis}) if it is allowed, and (f) the transfer time (tt_{pu}) from this unit. Here, the transfer of a rejected batch and its reprocessing in an alternative unit are also considered.

$$\mathrm{FT}_{pis} = \mathrm{ST}_{pis} + \mathrm{WT}_{pis} + \sum_{u' \in U_{ps'}} \mathrm{tt}_{pu'} Y_{pis'u'} + \sum_{u \in U_{ps}} (\mathrm{npt}_{psu} + \mathrm{ust}_{pu} + \mathrm{tt}_{pu}) Y_{pisu} \quad \forall (p,i,s) \in (T^{\mathrm{new}} \cup T^{\mathrm{da}} \cup T^{\mathrm{nda}}) \cap T^{\mathrm{fix}}:\ s = s' + 1, s \ne \{s_{pi}^{\mathrm{f}}\} \tag{6}$$

3.1.6.b. For Stages with Flexible Recipe. These constraints defined for tasks (p,i,s) with flexible recipes take into account that the predefined optimal processing time may be modified by a deviation ($\delta_{pis,\mathrm{DTOP}}$) of the nominal recipe-processing time (npt_{psu}). This situation may arise when others flexible-recipe items are changed in order to improve the plantwide performance.

$$\mathrm{FT}_{pis} = \mathrm{ST}_{pis} + \mathrm{WT}_{pis} + \sum_{u' \in U_{ps'}} \mathrm{tt}_{pu'} Y_{pis'u'} + \sum_{u \in U_{p,s}} (\mathrm{npt}_{psu} + \delta_{pis,\mathrm{DTOP}} + \mathrm{ust}_{pu} + \mathrm{tt}_{pu}) Y_{pisu} \quad \forall\, (p,i,s) \in (T^{\mathrm{new}} \cup T^{\mathrm{da}} \cup T^{\mathrm{nda}}) \cap T^{\mathrm{flex}}:\ s = s' + 1, s \neq \{s^{\mathrm{f}}_{pi}\} \quad (7)$$

3.1.7. Duration of the First Stage of Batch *i* of Product *p*. Specific constraints are needed for the first processing stage in order to account for the initial transfer time, which is represented here by the parameter itt_{pu}.

3.1.7.a. Stages with Fixed Recipe.

$$\mathrm{FT}_{pis} = \mathrm{ST}_{pis} + \mathrm{WT}_{pis} + \sum_{u \in U_{ps}} (\mathrm{npt}_{psu} + \mathrm{ust}_{pu} + \mathrm{tt}_{pu} + \mathrm{itt}_{pu}) Y_{pisu} \quad \forall\, (p,i,s) \in (T^{\mathrm{new}} \cup T^{\mathrm{da}} \cup T^{\mathrm{nda}}):\ s = \{s^{\mathrm{f}}_{pi}\} \quad (8)$$

3.1.7.b. Stages with Flexible Recipes.

$$\mathrm{FT}_{pis} = \mathrm{ST}_{pis} + \mathrm{WT}_{pis} + \sum_{u \in U_{ps}} (\mathrm{npt}_{psu} + \delta_{pis,\mathrm{DTOP}} + \mathrm{ust}_{pu} + \mathrm{tt}_{pu} + \mathrm{itt}_{pu}) Y_{pisu} \quad \forall\, (p,i,s) \in (T^{\mathrm{new}} \cup T^{\mathrm{da}} \cup T^{\mathrm{nda}}) \cap T^{\mathrm{flex}}:\ s = \{s^{f}_{pi}\} \quad (9)$$

3.1.8. Sequencing Constraints. For two different tasks (p,i,s) and (p',i',s') allocated in the same unit u:

3.1.8.a. If Task (p,i,s) Precedes Task (p',i',s'). Constraint 10 ensures that, if task (p,i,s) precedes task (p',i',s'), i.e., $X_{pis,p'i's'}$ is 1, and both tasks are processed in the same unit u, i.e., $Y_{pisu}=1$ and $Y_{p'i's'u}=1$, task (p',i',s') cannot start until task (p,i,s) is finished and the corresponding unit changeover time between products p and p' is completed.

$$\mathrm{ST}_{p'i's'} \geq \mathrm{FT}_{pis} + \mathrm{uch}_{pp'u} - M(1 - X_{pisp'i's'}) - M(2 - Y_{pisu} - Y_{p'i's'u}) \quad \forall\, (p,i,s),(p',i',s') \in (T^{\mathrm{new}} \cup T^{\mathrm{da}} \cup T^{\mathrm{nda}}),\ u \in (U_{ps} \cap U_{p's'}):\ (p < p') \cup (p = p', s < s') \quad (10)$$

3.1.8.b. If Task (p',i',s') Precedes Task (p,i,s). Constraint 11 considers the opposite case to constraint 10. Thus, if task (p,i,s) is performed after task (p',i',s'), so $X_{pisp'i's'}$ is 0, and both tasks are processed in the same unit u ($Y_{pisu}=1$ and $Y_{p'i's'u}=1$), task (p,i,s) cannot be processed until task (p',i',s') is finished and the changeover time between product p' and p in unit u is completed.

$$\mathrm{ST}_{pis} \geq \mathrm{FT}_{p'i's'} + \mathrm{uch}_{p'pu} - M(1 - X_{pisp'i's'}) - M(2 - Y_{pisu} - Y_{p'i's'u}) \quad \forall\, (p,i,s),(p',i',s') \in (T^{\mathrm{new}} \cup T^{\mathrm{da}} \cup T^{\mathrm{nda}}),\ u \in (U_{ps} \cap U_{p's'}):\ (p < p') \cup (p = p', s < s') \quad (11)$$

3.1.8.c. Different Batches (i, i') of the Same Product *p* at the Same Processing Stage *s*. This constraint sequences pairs of batches of the same product at the same processing stage, establishing that batch $i' > i$ will always be processed after batch i and the corresponding unit-dependent changeover time between the same product if this is applicable. In other words, this constraint arranges the batches of the same product, aiming at reducing the inherent combinatorial complexity of this problem.

$$\mathrm{ST}_{pi's} \geq \mathrm{FT}_{pis} + \sum_{u \in U_{ps}} \mathrm{uch}_{ppu} Y_{pisu} \quad \forall\, (p,i,i',s) \in (T^{\mathrm{new}} \cup T^{da} \cup T^{\mathrm{nda}}):\ i < i' \quad (12)$$

3.1.8.d. Consecutive Processing Stages of the Same Batch *i* of the Same Product *p*. This constraint synchronizes two consecutive stages, taking into account transfer times and unit setup times.

$$\mathrm{FT}_{pis} - \sum_{u \in U_{ps}} \mathrm{tt}_{pu} Y_{pisu} = \mathrm{ST}_{pis'} + \sum_{u' \in U_{ps'}} ust_{pu'} Y_{pis'u'} \quad \forall\, (p,i,s) \in (T^{\mathrm{new}} \cup T^{\mathrm{da}} \cup T^{\mathrm{nda}}):\ s' = s + 1, s \neq \{s^{\mathrm{l}}_{pi}\} \quad (13)$$

3.1.9. Tardiness and Earliness. Tardiness is defined here as the delay in the fulfillment of the order i of product p with respect to its due date (d_{pi}),

$$\mathrm{TA}_{pi} \geq \mathrm{FT}_{pis} - d_{pi} \quad \forall (p,i,s) \in (T^{\mathrm{new}} \cup T^{\mathrm{da}} \cup T^{\mathrm{nda}}):\ s = \{s^{\mathrm{l}}_{pi}\} \quad (14)$$

while earliness is the amount of time that order (p,i) is anticipated with respect to its due date.

$$\mathrm{EA}_{pi} \geq d_{pi} - \mathrm{FT}_{pis} \quad \forall (p,i,s) \in (T^{\mathrm{new}} \cup T^{\mathrm{da}} \cup T^{\mathrm{nda}}):\ s = \{s^{\mathrm{l}}_{pi}\} \quad (15)$$

3.1.10. Problem Objective Function. The objective of the proposed formulation is to minimize the associated cost of a weighted function of earliness and tardiness penalty costs as well as the deviation from nominal recipe costs. The level of importance of tardiness and earliness penalties is estimated by two cost factors tcost and ecost, respectively. In this way, orders are enforced to be processed near to their respective due dates, i.e., in a just-in-time manner, minimizing inventory cost while taking into account allowable recipe changes. Keeping inventory at a minimum level reduces the inventory cost, the amount of spoilage (i.e., chemicals with low stability over time), and the investment cost in storage tanks. Therefore, the objective function can be represented by the following expression:

$$\min \mathrm{TCOST} = \sum_{p \in P} \sum_{i \in I_p} (\mathrm{tcost} \cdot \mathrm{TA}_{pi} + \mathrm{ecost} \cdot \mathrm{EA}_{pi}) + \sum_{(p,i,s) \in T^{\mathrm{flex}}} \mathrm{DC}_{pis} \quad (16)$$

Furthermore, other performance criteria can also be incorporated as makespan, maximum throughput, or minimum number of tardy orders, taking always into account the cost associated with recipe modifications.

4. Case Study

The proposed rescheduling strategy under the flexible-recipe framework will be illustrated by solving a modified version of the case study proposed by Romero et al.[9] In the example addressed, five products are manufactured in four processing stages with available alternative equipment units. The flexible recipe for the production of benzyl alcohol is introduced within this production scenario, i.e., second stage of product P1 at unit U2. Transfer times as 5% of the processing time of each processing task have been also considered. For each product, a set of production orders comprising a single batch of each product with a specific due date is defined. Four late orders are reported in Table 2. They are order 2 of products P2 and P4 and orders 5 and 6 of product P1. Nominal batch processing times, available processing units, order due dates, and penalty factors for tardy orders are tabulated in Table 2.

The crossed-Cannizaro reaction for the batchwise production of benzyl alcohol from the reduction of benzaldehyde has been

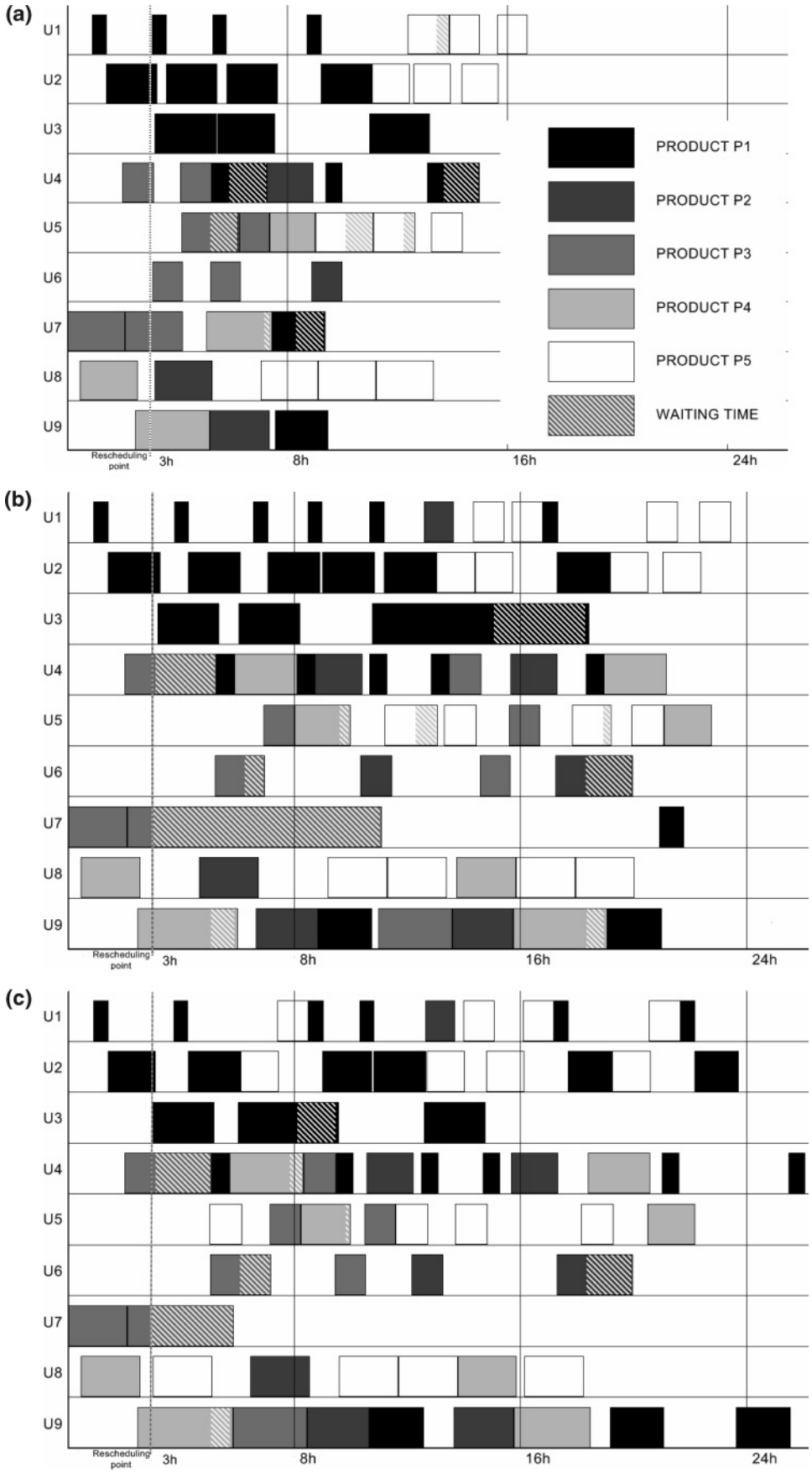

Figure 2. (a) Schedule in progress at the rescheduling point for scenario 1; (b) optimal rescheduling without considering recipe flexibility for scenario 1 (O.F. = 139.74 m.u.); (c) optimal rescheduling considering recipe flexibility for scenario 1 (O.F. = 129.93 m.u.).

studied by Keesman.[21] This author proposed a quadratic model to predict the variation of the yield of the reaction (DPS) for a priori known disturbances in the process inputs: DTEMP (temperature deviation), DTOP (operation time deviation),

Table 4. Comparison between Fixed and Flexible Rescheduling for Scenario 1

	fixed recipe (Figure 2b)	flexible recipe (Figure 2c)
tardiness cost, m.u.	131.81	116.44
earliness cost, m.u.	7.93	11.54
recipe modification cost, m.u.	0.00	1.95
objective function, m.u.	139.74	129.93
binary vars., cont. vars., constraints	283, 555, 974	283, 676, 1131
CPU time, s (AMD Athlon 2600 MHz, CPLEX 7.5)	78	207

Table 5. Recipe Modifications as Rescheduling Actions for Scenario 1

product, batch, stage	δ_{DTEMP}, °C	δ_{DTOP}, h	δ_{DKOH}, g	δ_{DFOR}, g
P1, I1, S2	0	−0.175	0	+0.007
P1, I3, S2	0	−0.125	0	+0.005
P1, (I5−I6), S2	0	−0.300	0	+0.013

DKOH (deviation of the amount of catalyst), and DFOR (deviation of the amount of the excess reactant formaldehyde). Equation 17 shows the linearized model proposed by Romero et al.,[9] which was incorporated into the proposed MILP rescheduling formulation. Table 3 shows the flexible-recipe variables and the maximum negative and positive deviations allowable for the flexible items around the nominal operating conditions.

$$\delta_{P1,i,S2,DPS} = 4.4\delta_{P1,i,S2,DTEMP} + 4\delta_{P1,i,S2,DTOP} + 95\delta_{P1,i,S2,DKOH} + 95\delta_{P1,i,S2,DFOR} \quad \forall i \in I_{P1} \quad (17)$$

Additional flexibility has been considered in the first stage of product P1 at unit U1 (eq 18):

$$\delta_{P1,i,S1,DTEMP} = 10\delta_{P1,i,S1,DTOP} \quad \forall i \in I_{P1} \quad (18)$$

This is a preheating stage in which the temperature is directly proportional to the duration of the task. The final temperature achieved by this task corresponds to the temperature for the reaction in the next processing stage, as shown in constraint 19.

$$\delta_{P1,i,S2,DTEMP} = \delta_{P1,i,S1,DPS} \quad \forall i \in I_{P1} \quad (19)$$

Three different scenarios are addressed to demonstrate the applicability of the proposed approach, which is implemented within the modeling language GAMS[22] using CPLEX version 7.5.

4.1. Scenario 1. The proposed rescheduling approach is applied to the schedule in progress shown in Figure 2a, which has to be updated at time 3 h in order to face the breakdown of unit 7 with a repairing time of 17 h. In addition, new batches corresponding to the arrival of late orders must also be inserted in the ongoing schedule (marked with stars in Table 2). So, two unexpected events, a unit breakdrown and late order arrivals, occur in the plant simultaneously. The selected objective function to be minimized is the total cost associated to the order tardiness (tcost = 5 m.u./h) and earliness (ecost = 1 m.u./h), as well as the corresponding cost for manipulating the process conditions shown in Table 3. Thus, the proposed formulation for this scenario minimizes eq 16 subject to constraints 1−15.

Figure 2b shows the proposed reschedule plan without considering the recipe flexibility. Figure 2c depicts a flexible reschedule that, despite the recipe-modification cost, results in a better solution in terms of the proposed objective function.

Table 6. Comparison between Fixed and Flexible Rescheduling for Scenario 2

	fixed reschedule (Figure 3b)	flexible reschedule (Figure 3c)
number of tardy orders	5	4
tardy orders penalization, m.u.	20.00	15.00
recipe modification cost, m.u.	0.00	1.04
objective function, m.u.	20.00	16.04
binary vars., cont. vars., constraints	276, 527, 902	276, 626, 1022
CPU time, s (AMD Athlon 2600 MHz,CPLEX 7.5)	14.7	4.7

Table 7. Recipe Modifications as Rescheduling Actions for Scenario 2

product, batch, stage	δ_{DTEMP}, °C	δ_{DTOP}, h	δ_{DKOH}, g	δ_{DFOR}, g
P1, I4, S2	0	−0.300	0	+0.013
P1, I5, S2	0	−0.187	0	+0.008

This improvement (12% reduction of the objective function value) comes not only from the recipe changes but also from several modifications of sequencing decisions, which can be easily observed by comparing parts b and c of Figure 2. The reduction of processing times allowed by flexible recipes involves an additional cost but produces new opportunities and gaps permitting reallocating or resequencing actions otherwise disallowed. Advantage may be taken of these extra chances to reduce tardiness and earliness and to obtain better objective function values. This is the situation illustrated by Figure 2c, in which, despite the makespan increase, completion times are tightened to their due dates by a set of starting-time adjustments not allowed in the rigid schedule in Figure 2b. For example, a first batch of product P5 has been rescheduled and brought forward to start just after the rescheduling point.

In this scenario, the proposed recipe modifications are shown in Table 5. In order to improve the customer satisfaction, the second stage of the first, third, fifth, and sixth batches of product P1 (flexible-reaction tasks) reduces their processing time at the expense of increasing the amount of formaldehyde. However, in none of the batches of P1, flexibility is exploited for the preheating stage (first stage). This situation arises because flexibility always has an associated cost and this stage does not suppose a bottleneck in the process, and consequently, no improvement in the performance of the plant can be achieved. Therefore, this example clearly reflects the high importance of recipe flexibility in the rescheduling process of critical and hard-constrained batch operations.

It is worth noting that the computational effort for updating the current schedule remains very low, with short reaction times being highly important in real industrial environments. Table 4 summarizes the main features of the schedules generated without and with recipe flexibility.

4.2. Scenario 2. In this second scenario, rescheduling actions are required to deal with the breakdown of unit U3 at time 4 h with a repairing time of 20 h. Economical factors have changed from the previous scenario, and now, the same cost (3 m.u./g) is assumed for deviations in KOH and formaldehyde. The rest of the deviation costs remain the same as stated in Table 3.

Furthermore, in this second scenario, an alternative objective function is evaluated. Now, the number of tardy orders in the schedule is penalized, i.e., orders ending after their respective due dates. In order to calculate this number of tardy orders in the schedule, a new binary variable must be defined, TO_{pis}, which denotes if the last stage of the batch i of a product p is completed after the promised due date ($TO_{pis} = 1$).

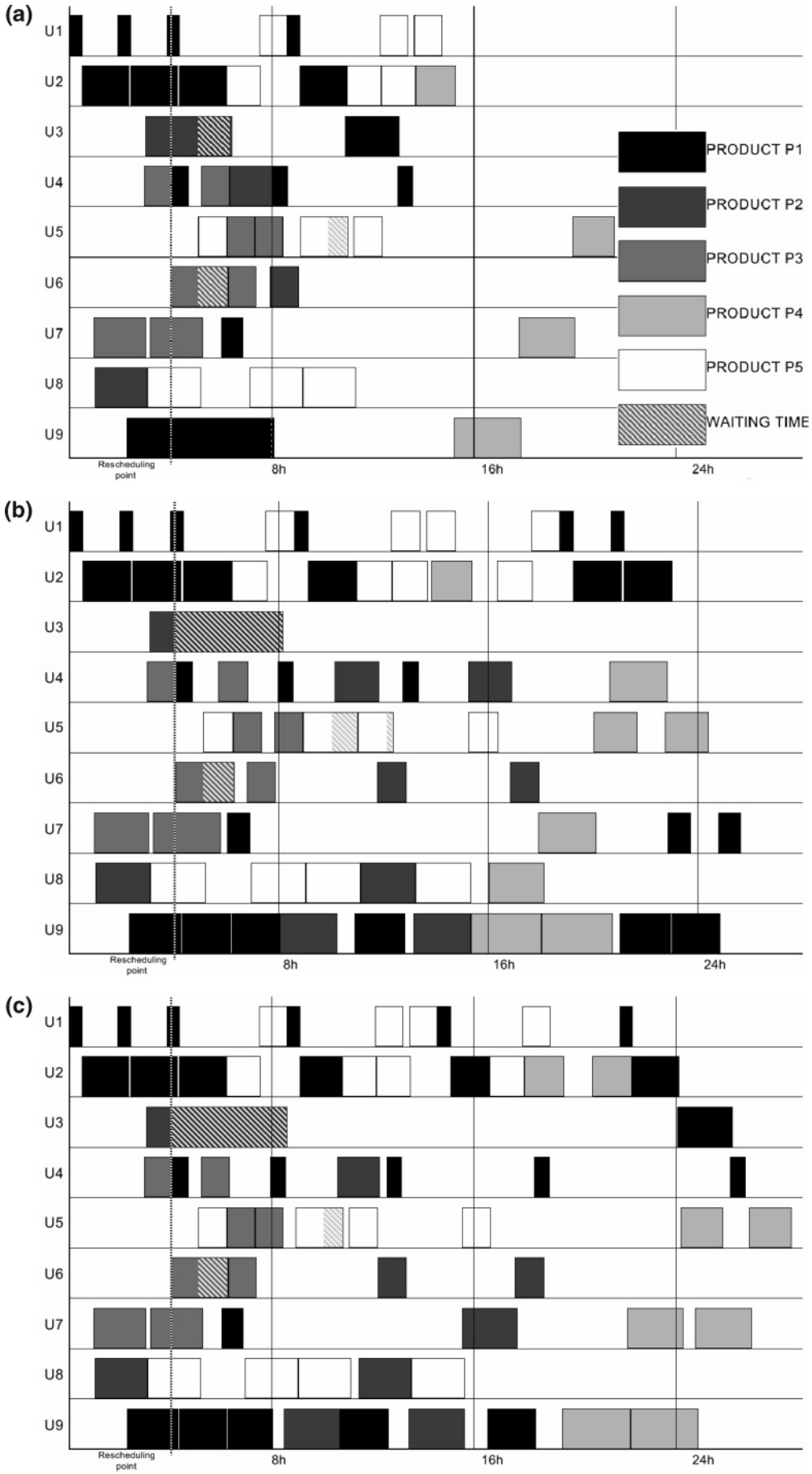

Figure 3. (a) Schedule in progress at the rescheduling point for scenario 2; (b) optimal rescheduling without considering recipe flexibility for scenario 2 (O.F. = 20.00 m.u.); (c) optimal rescheduling considering recipe flexibility for scenario 2 (O.F. = 16.04 m.u.).

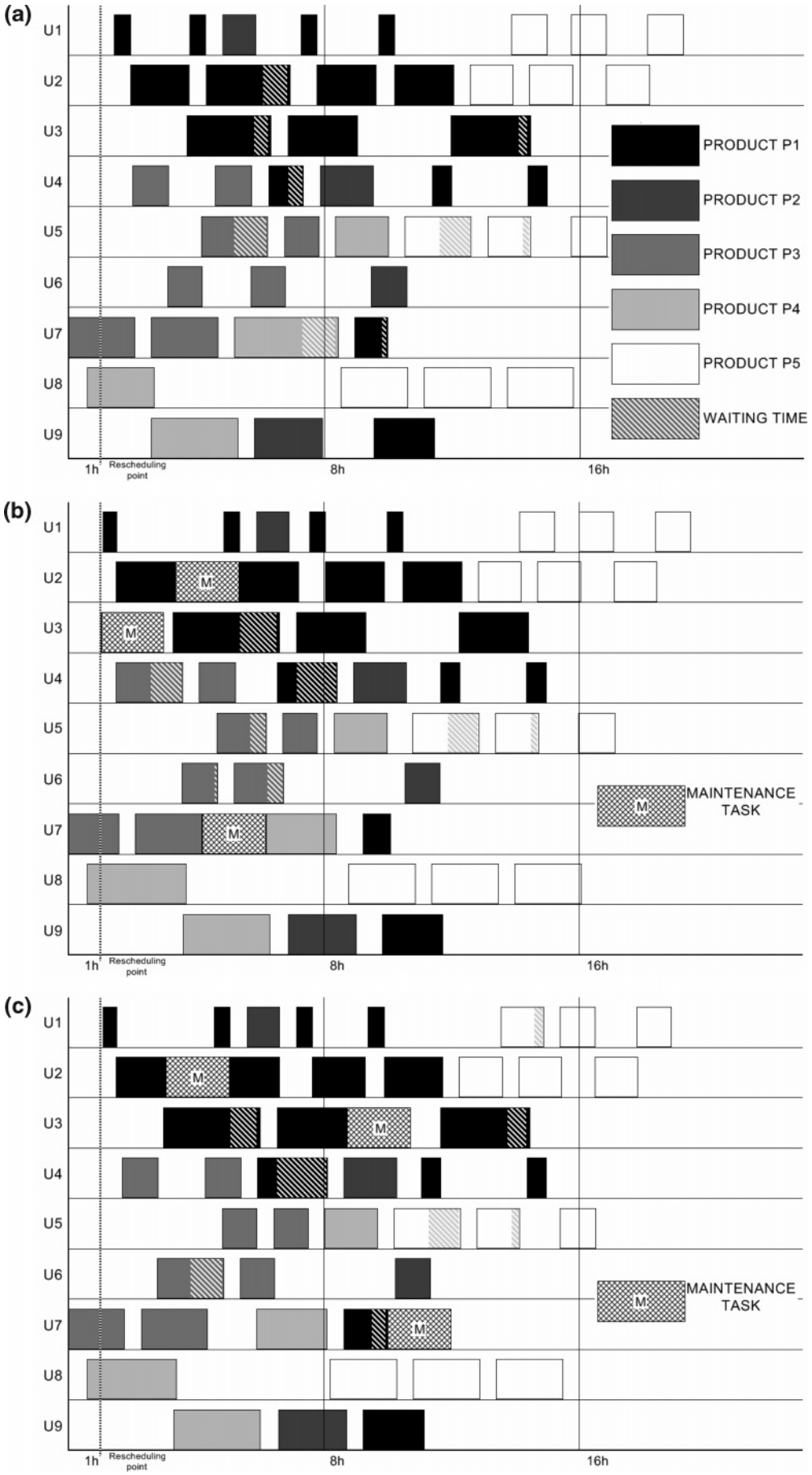

Figure 4. (a) Schedule in progress for scenario 3; (b) optimal rescheduling without considering recipe flexibility and maintenance operations for scenario 3 (O.F. = 60.08 m.u.); (c) optimal rescheduling considering recipe flexibility and maintenance operations for scenario 3 (O.F. = 50.89 m.u.).

$$\mathrm{TA}_{pis} - M\cdot\mathrm{TO}_{pis} \le 0 \quad \forall (p,i,s) \in (T^{\mathrm{new}} \cup T^{\mathrm{da}} \cup T^{\mathrm{nda}}):\ s = \{s^{\mathrm{l}}_{pi}\} \quad (20)$$

The objective function includes a deviation cost from the nominal recipe parameters. Also, the parameter pto defines the product-dependent penalty cost for tardy orders (see Table 2).

$$\min \mathrm{TCOST} = \sum_{p\in P}\sum_{i\in I_p}\sum_{s=s^{\mathrm{l}}_{pi}} (\mathrm{TO}_{pis}\cdot \mathrm{pto}_p) + \sum_{(p,i,s)\in T^{\mathrm{flex}}} \mathrm{DC}_{pis} \quad (21)$$

Therefore, in this scenario, objective function 21 is minimized subject to constraints 1−15 and 20.

Looking at the results, Figure 3a illustrates the initial schedule, Figure 3b illustrates the rescheduling solution considering fixed recipes, and, finally, Figure 3c illustrates the rescheduling solution using flexible recipes. In a similar way as in the previous scenario, Table 6 shows the main results for these approaches and the computational effort that has radically decreased because of the nature of the objective function. Different assignments and sequencing of tasks are obtained by using flexible recipes, which finally leads to an improved objective function. Table 7 summarizes the recipe modification executed as a rescheduling action, i.e., second stage of batches I4 and I5 of product P1 reduces their processing times by increasing the amount of formaldehyde.

4.3. Scenario 3. In this scenario, no unexpected events occur, and only a series of maintenance tasks have to be performed in some equipment units. The aim of these maintenance tasks is precisely to avoid the occurrence of these incidences shown in the previous scenarios, which led to longer unavailability intervals. This scenario proposes the scheduling problem of maintenance tasks that have to be carried out within a restricted period of time. Therefore, in this situation, all the tasks previously scheduled are nondirectly affected tasks, so the rescheduling actions allowed are just resequencing and recipe modifications. Then, a new type of task appears, the maintenance task (T^{maint}), which is not included in the recipe of any product but needs to be considered in the schedule. These maintenance tasks can be included in the model and treated in the same way as the new tasks (T^{new}), so the mathematical model remains the same. Three maintenance tasks of 2 h have been considered for units U2, U3, and U7. Additionally, the technical service will be available from the rescheduling point at hour 1 until hour 12, so these tasks must be arranged within this interval. This just supposes a new constraint, determining that the finishing time for these tasks has to be less than or equal to 12 h. Uniform sequence-dependent changeover times ($\mathrm{uch}_{pp'u}$) of 0.5 h for transition between different products have also been considered. The objective function to be minimized is the one presented in scenario 1.

Figure 4a shows the initial ongoing schedule that did not consider any maintenance task. Figure 4b illustrates the schedule proposed including these maintenance tasks, while the schedule in Figure 4c considers recipe modification as additional rescheduling actions. These figures point out how maintenance tasks are sequenced differently in these two approaches. Table 8 compares rescheduling using fixed and flexible recipes, while Table 9 presents the rescheduling recipe modification actions. CPU time is again smaller, since all assignment decisions are already taken and just sequencing decisions are made in order to insert the maintenance tasks.

Table 8. Comparison between Fixed and Flexible Rescheduling for Scenario 3

	fixed reschedule (Figure 3b)	flexible reschedule (Figure 3c)
tardiness cost, m.u.	54.56	42.44
earliness cost, m.u.	5.51	6.68
recipe modification cost, m.u.	0	1.65
objective function, m.u.	60.08	50.89
binary vars., cont. vars., constraints	156, 555, 974	156, 472, 726
CPU time, s (AMD Athlon 2600 MHz, CPLEX 7.5)	0.3	0.3

Table 9. Recipe Modifications as Rescheduling Actions for Scenario 3

product, batch, stage	δ_{DTEMP}, °C	δ_{DTOP}, h	δ_{DKOH}, g	δ_{DFOR}, g
P1, I1, S2	0	−0.300	0	+0.013
P1, I2, S2	0	−0.287	0	+0.012
P1, I3, S2	0	−0.175	0	+0.007

5. Conclusions

This work presents an efficient MILP-based rescheduling framework that incorporates the recipe flexibility as an additional rescheduling opportunity. The approach is based on a continuous-time domain representation and the generalized notion of precedence. The rescheduling strategy considers the ongoing production schedule and the current process conditions in order to simultaneously adapt the production recipe to the new scenario and reoptimize the schedule with regard to the batches still to be processed. Different objective functions can be employed to regain feasibility or optimality at minimum cost. Efficiency and applicability of the proposed strategy is demonstrated by successfully solving a complex rescheduling problem in a multipurpose batch plant with reasonable computational effort. Finally, the future work in this research area is envisaged to develop an effective method to deal with rescheduling problems arising in batch plants with flexible production recipes characterized by nonlinear models.

Acknowledgment

Financial support received from the European Community projects (MRTN-CT-2004-512233; RFC-CR-04006; and INCO-CT-2005-013359) and the Generalitat de Catalunya with the European Social Fund (FI grant) is fully appreciated.

Nomenclature

Subscripts

p, p' = product
i, i' = batch
s, s' = processing stage
p,i,s = task (p,i,s)
u, u' = processing unit
f, f' = flexible recipe items

Sets

T^{exec} = tasks (p,i,s) already executed
T^{nda} = nondirectly affected tasks (p,i,s) by the unexpected event
T^{da} = directly affected tasks (p,i,s) by the unexpected event
T^{new} = tasks (p,i,s) corresponding to new orders to be inserted into the current schedule
T^{fix} = tasks (p,i,s) following a traditional fixed recipe
T^{flex} = tasks (p,i,s) following a flexible recipe
T^{maint} = maintenance tasks (p,i,s)
P = set of products p
I_p = batches of product p

S_p = stages for producing product p
U_{ps} = set of available units for processing product p at stage s
FP_{ps} = set of recipe items of the linear flexible model for stage s of product p

Parameters

d_{pi} = due date of single-batch order of product p
npt_{psu} = nominal processing time for sth stage of product p in unit u
s_{pi}^{f} = first processing stage for batch i of product p
s_{pi}^{l} = last processing stage for batch i of product p
tt_{pu} = transfer time of product p from unit u
itt_{pu} = transfer time of product p to unit u at the first processing stage
ust_{pu} = setup time for unit u for processing product p
$uch_{pp'u}$ = product-dependent changeover time for unit u between products p and p'
ro_{pis} = release time for task (p,i,s)
ru_u = ready time of unit u
M = very large number
tcost = weighting coefficient for tardiness cost penalty
ecost = weighting coefficient for earliness cost penalty
$lfmod_{psf}$ = linear flexible recipe model coefficient of recipe item f at processing stage s of product p
$fplb_{psf}$ = maximum negative deviation for recipe item f of stage s of product p
$fpub_{psf}$ = maximum positive deviation for recipe item f of stage s of product p
$dcost_{psf}$ = unitary deviation cost for recipe parameter f at processing stage s of product p

Continuous Variables

FT_{pis} = completion time for the task (p,i,s)
ST_{pis} = starting time for task (p,i,s)
WT_{pis} = waiting time for task (p,i,s)
δ_{pisf} = deviation of the flexible parameter f in flexible-recipe task (p,i,s)
TA_{pi} = tardiness of batch i of product p
EA_{pi} = earliness of batch i of product p
DC_{pis} = recipe cost of flexible task (p,i,s) due to deviation from nominal parameters
TCOST = weighted objective function

Binary Variables

$X_{pisp'i's'}$ = binary variable equals 1 if task (p,i,s) is processed before another task (p',i',s') and 0 otherwise
Y_{pisu} = binary variable equals 1 if task (p,i,s) is allocated to equipment unit u and 0 otherwise
TO_{pis} = binary variable equals 1 if task (p,i,s) is a tardy order and 0 otherwise

Literature Cited

(1) Darby, M. L.; White, D. C. On-line optimization of complex processes. *Chem. Eng. Prog.* **1988**, *51*.

(2) Marlin, T. E.; Hrymak, A. N. Real-time optimization of continuous processes. *AIChE Symp. Ser.* **1997**, *93*, 156.

(3) Ferrer-Nadal, S.; Yélamos-Ruiz, I.; Graells, M.; Puigjaner, L. An integrated framework for on-line supervised optimization. *Comput. Chem. Eng.* **2007**, *31*, 401.

(4) Méndez, C. A.; Cerdá, J.; Harjunkoski, I.; Grossmann, I. E.; Fahl, M. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput. Chem. Eng.* **2006**, *30*, 913.

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

(6) Verwater-Lukszo, Z. A practical approach to recipe improvement and optimization in the batch processing industry. *Comput. Ind.* **1998**, *36*, 279.

(7) Sel, D.; Hvala, N.; Strmcnik, S.; Milanic, S.; Suk-Lubej, B. Experimental testing of flexible recipe control based on a hybrid model. *Control Eng. Pract.* **1999**, *7*, 1191.

(8) Rutten, W. G. M.; Bertrand, J. W. M. Balancing stocks, flexible recipe costs and high service level requirements in a batch process industry: A study of a small scale model. *Eur. J. Oper. Res.* **1998**, *110*, 626.

(9) Romero, J.; Espuña, A.; Friedler, F.; Puigjaner, L. A new framework for batch process optimization using the flexible recipe. *Ind. Eng. Chem. Res.* **2003**, *42*, 370.

(10) Bassett, M. H.; Pekny, J. F.; Reklaitis, G. V. Using detailed scheduling to obtain realistic operating policies for a batch processing facility. *Ind. Eng. Chem. Res.* **1997**, *36*, 1717.

(11) Balasubramanian, J.; Grossmann, I. E. Scheduling optimization under uncertainty—An alternative approach. *Comput. Chem. Eng.* **2003**, *27*, 469.

(12) Lin, X.; Janak, S. L.; Floudas, C. A. A New Robust Optimization Approach for Scheduling under Uncertainty: I. Bounded Uncertainty. *Comput. Chem. Eng.* **2004**, *28*, 1069.

(13) Bonfill, A.; Espuña, A.; Puigjaner, L. Addressing robustness in scheduling batch processes with uncertain operation times. *Ind. Eng. Chem. Res.* **2005**, *44*, 1524.

(14) Cott B. J.; Macchietto S. Minimizing the effects of batch process variability using on-line schedule modification. *Comput. Chem. Eng.* **1989**, *13*, 105.

(15) Kanakamedala, K. B.; Reklaitis, G. V.; Venkatasubramanian, V. Reactive schedule modification in multipurpose batch chemical plants. *Ind. Eng. Chem. Res.* **1994**, *33*, 77.

(16) Sanmartí, E.; Espuna, A.; Puigjaner, L. Batch production and preventive maintenance scheduling under equipment failure uncertainty. *Comput. Chem. Eng.* **1997**, *21*, 1157.

(17) Vin, J. P.; Ierapetritou, M. G. A new approach for efficient rescheduling of multiproduct batch plants. *Ind. Eng. Chem. Res.* **2000**, *39*, 4228.

(18) Méndez, C. A.; Cerdá, J. Dynamic scheduling in multiproduct batch plants. *Comput. Chem. Eng.* **2003**, *27*, 1247.

(19) Méndez, C. A.; Cerdá, J. A MILP framework for batch reactive scheduling with limited discrete resources. *Comput. Chem. Eng.* **2004**, *28*, 1068.

(20) Méndez, C. A.; Henning G. P.; Cerdá, J. An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities. *Comput. Chem. Eng.* **2001**, *25*, 701.

(21) Keesman, K. J. Application of Flexible Recipes for model building, batch process optimization and control. *AIChE J.* **1993**, *39*, 581.

(22) Brooke, A.; Kendrick, D.; Meeraus, A; Raman, R. *GAMS—A user's guide*; The Scientific Press: San Francisco, CA, 1998.

Received for review September 27, 2006
Revised manuscript received June 21, 2007
Accepted June 26, 2007

IE061255+