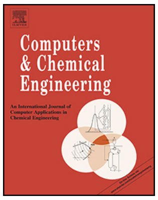
Computers and Chemical Engineering 67 (2014) 83–102

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng



Simultaneous regular and non-regular production scheduling of multipurpose batch plants: A real chemical–pharmaceutical case study

Samuel Moniz[a,c,*], Ana Paula Barbosa-Póvoa[a,*], Jorge Pinho de Sousa[b,c]

[a] *Centro de Estudos de Gestão, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisboa, Portugal*
[b] *INESC TEC (formerly INESC Porto), Porto, Portugal*
[c] *Faculdade de Engenharia da Universidade do Porto, Porto, Portugal*

ARTICLE INFO

Article history:
Received 20 April 2013
Received in revised form 25 March 2014
Accepted 26 March 2014
Available online 4 April 2014

Keywords:
Multipurpose batch plants
Campaign and short-term scheduling
Rolling horizon
MILP models
Resource-Task Network

ABSTRACT

Regular and non-regular production can often be found in multipurpose batch plants, requiring two distinct operating strategies: campaign and short-term production. This paper proposes a solution approach for simultaneous scheduling of campaign and short-term products in multipurpose batch plants. Regular products follow a cyclic schedule and must cover several product deliveries during the scheduling horizon, while non-regular products have a non-cyclic schedule. The proposed approach explores the Resource-Task Network (RTN) discrete-time formulation. Moreover, a rolling horizon approach, and reformulation and branching strategies have been applied to deal with the computational complexity of the scheduling problem. Real case instances of a chemical–pharmaceutical industry are solved, showing the applicability of the solution approach.

1. Introduction

Multipurpose batch plants may operate in campaign or short-term modes, or may have both operational modes running in the same facility. The latter is the case of some process industries such as the chemical–pharmaceutical industry where high and low volume products need to be produced simultaneously. Products that are already in commercialization commonly present stable demands and are produced in large batches, being the campaign mode the preferred operational mode. In this case, the production resources are allocated to tasks that are executed in a cyclic way, thus defining production lines that tend to be stable for long periods of time. This approach leads to obvious benefits such as minimizing the changeovers costs while reducing the complexity of the operations. Alternatively, plants may also have short-term demands. Here, customers' orders of low quantities are placed for specific time windows. In the case of the chemical–pharmaceutical industry the products under development fit in this situation.

Globally, multipurpose batch plants need then to respond to a heterogeneous demand and plant resources have to be shared between campaign and short-term production modes. The plant responsiveness becomes critical and should be able to accommodate new orders at the minimum cost and with the minimum perturbation of the existing schedule, since as pointed out by Shah (2004), time-to-market is certainly the most important driver in the pharmaceutical industry.

Modeling and optimization methods have been extensively applied in batch processes problems, requiring a clever exploitation of the problem structure (Reklaitis, 1995). Moreover, the integration of different dynamic decisions such as design, planning and scheduling proved to be a good way of tackling the complexity of these problems (Barbosa-Povoa, 2007; Verderame et al., 2010).

The present paper addresses this problem and proposes a solution approach for scheduling multipurpose batch plants that simultaneously consider two different operating conditions – regular and non-regular production. The former encompasses products that are manufactured regularly in predefined production lines and the latter includes under development products having no defined production lines. The production resources are shared between both types of products. The rest of the paper is structured as follows. In Section 2, a literature review is presented. The main characteristics of the scheduling problem are presented in Section 3 and the proposed algorithm is described in Section 4. In Section 5, a mathematical model for the problem is presented. In Section 6 are presented the solution methods and in Section 7 the numerical results are discussed. Finally, in Section 8 some concluding remarks are given.

[*] Corresponding author at: Centro de Estudos de Gestão, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisboa, Portugal. Tel.: +351 214233507.
E-mail address: apovoa@tecnico.ulisboa.pt (A.P. Barbosa-Póvoa).

http://dx.doi.org/10.1016/j.compchemeng.2014.03.017
0098-1354/© 2014 Elsevier Ltd. All rights reserved.

2. Background

2.1. Scheduling of multipurpose batch plants

Scheduling of multipurpose batch plants has been intensively addressed in the literature, covering a wide range of problems. On the production planning and scheduling problems a variety of modeling tools has been developed to tackle the associated problems, see reviews by (Mendez et al., 2006; Barbosa-Povoa, 2007; Li and Ierapetritou, 2008; Maravelias and Sung, 2009; Li et al., 2010; Verderame et al., 2010). Both the State-Task Network (STN) presented by Kondili et al. (1993) and the Resource-Task Network (RTN) suggested by Pantelides (1994) became two major frameworks used for solving scheduling problems in the chemical process industry, where discrete and continuous representations of time have been explored. Discrete-time formulations easily model inventory and backlog costs, intermediate and delivery dates, and often result into compact formulations that can be easily modified. However, they present some problems when modeling variable processing times and sequence-dependent changeovers. Moreover, the efficiency of the discrete-time models and the feasibility of the solutions depend on the number and duration of the time intervals considered. To overcome these issues, continuous-time models were developed, where different time grids were used. Common time grid formulations for all resources were developed by (Schilling and Pantelides, 1996; Castro et al., 2001; Maravelias and Grossmann, 2003) and unit-specific time events formulations were developed by (Ierapetritou and Floudas, 1998; Janak et al., 2004; Vooradi and Shaik, 2012). Continuous time formulations lead however to more complex models and present larger integrality gaps. Even though the above mentioned developments represent a large step in the optimization of the process industry operation, the requirements found in real production environments often lead to new challenges that have not yet been adequately addressed in the literature.

One relevant scheduling issue regards the determination of detailed schedules in large time horizons. Such problem is due to various reasons. Scheduling problems may depend on recipes with short and long processing tasks, thus a sufficient large time horizon is required to accommodate all products. Moreover, production planning may need to be checked and validated at the operational level. These cases can be found in many chemical industries and are difficult to solve especially if different products recipes are present. The obvious and immediate approach for tackling this type of problems is to apply a short-term scheduling model for the entire planning horizon. However, a scheduling model with such dimension would hardly be solved. Alternative approaches such as cyclic scheduling, campaign planning and decomposition methods have been developed aiming at decreasing this modeling challenge.

Shah et al. (1993) presented a general framework for periodic scheduling of multipurpose batch plants. The model is based on the State-Task Network representation in which the “wraparound operator” is developed. This approach can deal with complex operations of batch plants, but it is only suitable for single campaigns. Later on, Schilling and Pantelides (1999) proposed a mixed integer non-linear programming (MINLP) model for addressing the periodic scheduling problem. Due to difficulties in the linearization the authors developed a special branch-and-bound (B&B) algorithm that branches the discrete and continuous variables. More recently, Pochet and Warichet (2008) propose a continuous time MILP formulation for solving the periodic scheduling problem and use strengthening techniques to improve the model computational time, and MIP based heuristic methods to obtain good solutions quickly in the larger instances. Addressing the same type of problem, You et al. (2009) compared the Dinkelbach’s algorithm with commercial MINLP solvers and verified that this algorithm performed better. Castro et al. (2003) proposed discrete and continuous-time formulations based on the RTN formulation for deriving optimal periodic schedules. Results favor the discrete-time periodic formulation in the case study addressed by the authors. Wu and Ierapetritou (2004) developed a cyclic schedule approach based on the STN using a continuous time formulation. This approach assumes stable demand for the time horizon under consideration and aims at determining the optimal cyclic schedule and cycle length. Moreover, the approach has a decomposition scheme for determining the startup and shutdown phases. Pinto et al. (2005) increased the complexity of the periodic scheduling by simultaneously considering the design and retrofit of multipurpose batch plants. The model is based on a discrete-time RTN formulation. Castro et al. (2008) solved an industrial scheduling problem from the chemical–pharmaceutical industry by proposing a periodic RTN formulation.

For campaign planning a number of algorithms have been presented in the literature. Mauderli and Rippin (1979) proposed a sequential approach where first alternative production lines with single products are generated, and then campaigns with several products are formed from the combination of two or more single product production lines. A screening procedure is applied to identify the dominant campaigns and, in a last step, a production plan is generated by solving a LP or MILP problem that allocates the dominant campaigns to the available production time. Papageorgiou and Pantelides (1993) proposed a hierarchical approach for multipurpose batch plants that takes into account the inherent flexibility of such plants with respect to intermediate storage policies and processing units utilization. A three-step approach is presented. The first step determines the number of campaigns and active stages in each campaign. The second step addresses the campaigns separately to derive the optimal cyclic schedules for the active stages and aims at improving the production rates of some stages. Finally, the third step reconsiders the timing of the campaigns determined in the previous step attempting to maximize the overall production value. Later Papageorgiou and Pantelides (1996a,b) proposed a single-level model for planning and scheduling of multipurpose batch plants capable of simultaneously determining the campaigns (duration and products), the unit-task allocation and the task timings. Sundaramoorthy and Karimi (2004) propose a multi-period continuous-time MILP model. Computational tests have shown that the model is quite efficient even for long term planning periods. A limiting aspect of the approach followed by the authors is that production lines are considered instead of processing units, thus it is assumed that processing units are permanently allocated to a specific production line and cannot be shared. In practice, it is common to select a set of production resources to define a production line that will operate during a certain time period, sufficient to supply a given demand. More recently, Fumero et al. (2012) presented a solution approach for the scheduling of multistage multiproduct batch plants. They first solve a simplified slot-based continuous – time formulation that involves preordering constraints for the assignment of batches to slots in each stage. This provides a good upper bound for the campaign length of the detailed scheduling model solved in the second phase.

Regarding decomposition methods a good discussion is presented by Bassett et al. (1996a,b). The authors analyze several decomposition methods for large-scale scheduling problems. Considering the same type of problems Wu and Ierapetritou (2003) developed an iterative approach that uses a lower bound obtained by heuristic-based decomposition approaches and an upper bound based on Lagrangean relaxation and Lagrangean decomposition. Lin et al. (2002) developed a rolling horizon approach. A two-level decomposition model is proposed to determine the current horizon and the products that shall be included. Wu and Ierapetritou (2007) also used a rolling horizon strategy to solve a planning and scheduling problem with uncertainty. A *sequence factor* is used to estimate the impact of the tasks sequencing in the planning problem. This parameter is used to make the planning and scheduling results converge. Erdirik-Dogan and Grossmann (2007) addressed the single stage problem with parallel units and sequence dependent changeovers. They propose an aggregate planning model that underestimates the effects of the changeovers and sequencing variables, but can be solved very efficiently; and a detailed scheduling model that models accurately the tasks sequencing and changeovers. To solve the larger instances a rolling horizon approach is suggested. Amaro and Barbosa-Póvoa (2008a,b) also studied the large scale scheduling and planning problems of batch plants using an extended STN representation (Chain-STN) in a supply chain context. An hierarchical decomposition procedure was proposed to link the planning with the scheduling decisions and a real case-study of a pharmaceutical industry was solved. Stefansson et al. (2011) proposed a decomposition algorithm that prioritizes the scheduling of the bottleneck units. The approach is applied to a multistage batch plant and the problem is decomposed into two parts. They start by solving the bottleneck stage and then solve the remaining stages. Moreover, they compare discrete-time formulation based on Kondili et al. (1993) with a continuous-time general precedence formulation based on Méndez et al. (2001). The continuous-time formulation (limited to sequential processes) have provided more accurate solutions and used less computational time, compared with the discrete-time general formulation. Recently, Sundaramoorthy and Maravelias (2011) shown that discrete-time models have many advantages over continuous-time formulations. Their study indicates that discrete-time models have better performance concerning the solution times and integrality gap. Moreover, discrete-time formulations can be easily modified to account for other processing characteristics. In order to address the computational burden of the MILP models, Velez and Maravelias (2013) propose three reformulations to define the number of batches of each task and use as basis the STN formulation. Tests have shown that branching on the integer variable number of batches eliminates many symmetric solutions, leading to improve the model performance.

The existing approaches can deal with several problem complexities, but they are still quite limited in simultaneously managing mixed operating strategies such as the regular and non-regular production, or the campaign and short-term scheduling. The work described in this paper aims at reducing this gap and proposes a simple three-step approach that tries to explore the specific problem structure and the current industrial planning procedures, as used in the chemical–pharmaceutical industry.

2.2. Motivation for a mixed strategy

Depending on the product recipes structure and on the allowable task/processing unit assignment, we may have multiproduct or multipurpose operating strategies (Reklaitis, 1995). Multiproduct batch plants are settled to manufacture products that have similar recipes, with production lines employing many-to-one processing unit/task assignments and operating cyclically to accommodate serial campaigns. Multipurpose batch plants under campaign operation are more appropriate for products with dissimilar recipes, allowing many-to-many processing unit/task assignments, and possibly having several campaigns involving several production lines, each operating cyclically. General multipurpose plants can also be defined and refer to multipurpose plants that operate with no defined production lines and with non-periodic production, where different types of products are simultaneously produced.

In practice, a mixed strategy may be present in a given plant. This occurs when the product portfolio combines characteristics of both strategies. In these cases, part of the plant may operate using dedicated production lines, while the other part operates in the multipurpose mode; or the same resources may be shared among the processes that have to be performed. Since production resources such as processing units, raw materials or utilities are shared, scheduling integration is required. Table 1 summarizes the characteristics of regular and non-regular production for the case of the chemical–pharmaceutical industry. Non-regular products are products that do not have long-term demand, they are produced in relatively low quantities and for specific time windows. Therefore, they do not justify the establishment of dedicated production lines. The pharmaceutical products under development fit into this category. On the other hand, regular products have typically well-defined recipes and stable production lines and involve the delivery of large amounts of products during long periods of time.

3. Problem description

In the problem under investigation we consider that the following information is available: (i) the detailed recipes of the products that will be produced in campaign and short-term modes; (ii) the maximum and minimum capacity of the processing units; (iii) the demand and the delivery dates; (iv) changeover times required to clean units between products; (v) the minimum and the maximum cycle time for the regular products; and (vi) the costs and economic value of the products.

The objective is to maximize the overall profit for all products, while determining: (i) the cycle time T for the products to be scheduled in a campaign mode; (ii) the task unit assignment and sequencing for all products; (iii) the tasks batch sizes and storage levels; and (iv) the number of campaign cycles. Each product is defined by a recipe (see Fig. 1) that identifies the task sequence with the respective processing time and allowable processing units. Raw materials and final products have unlimited storage, while storable intermediaries have finite storage. Task batch sizes are limited by the capacity of the processing unit chosen.

4. Proposed algorithm

To solve the integrated problem as described above, we propose the following three-step procedure (see Fig. 2). In the first step,

Table 1
Characteristics of regular and non-regular production in the chemical–pharmaceutical industry.

Non-regular production	Regular production
There are stable and unstable product recipes. Recipes may change as a result of the process development.	Products have stable product recipes. Changes in the recipes are possible to do, but require legal and customer approvals.
Production has an irregular demand pattern and is triggered by customer orders.	Production has a regular demand pattern, usually established by a master production plan.
Demand needs are specified for a short period of time, typically few weeks.	Demand needs are planned for long term, typically from several months to one year.
Assignment of processing units to tasks can vary (*e.g.* scale-up of the production processes)	Assignment of processing units to tasks tends to be permanent, despite the existence of alternative processing units.
Products have tight delivery windows.	Products have relaxed delivery dates.

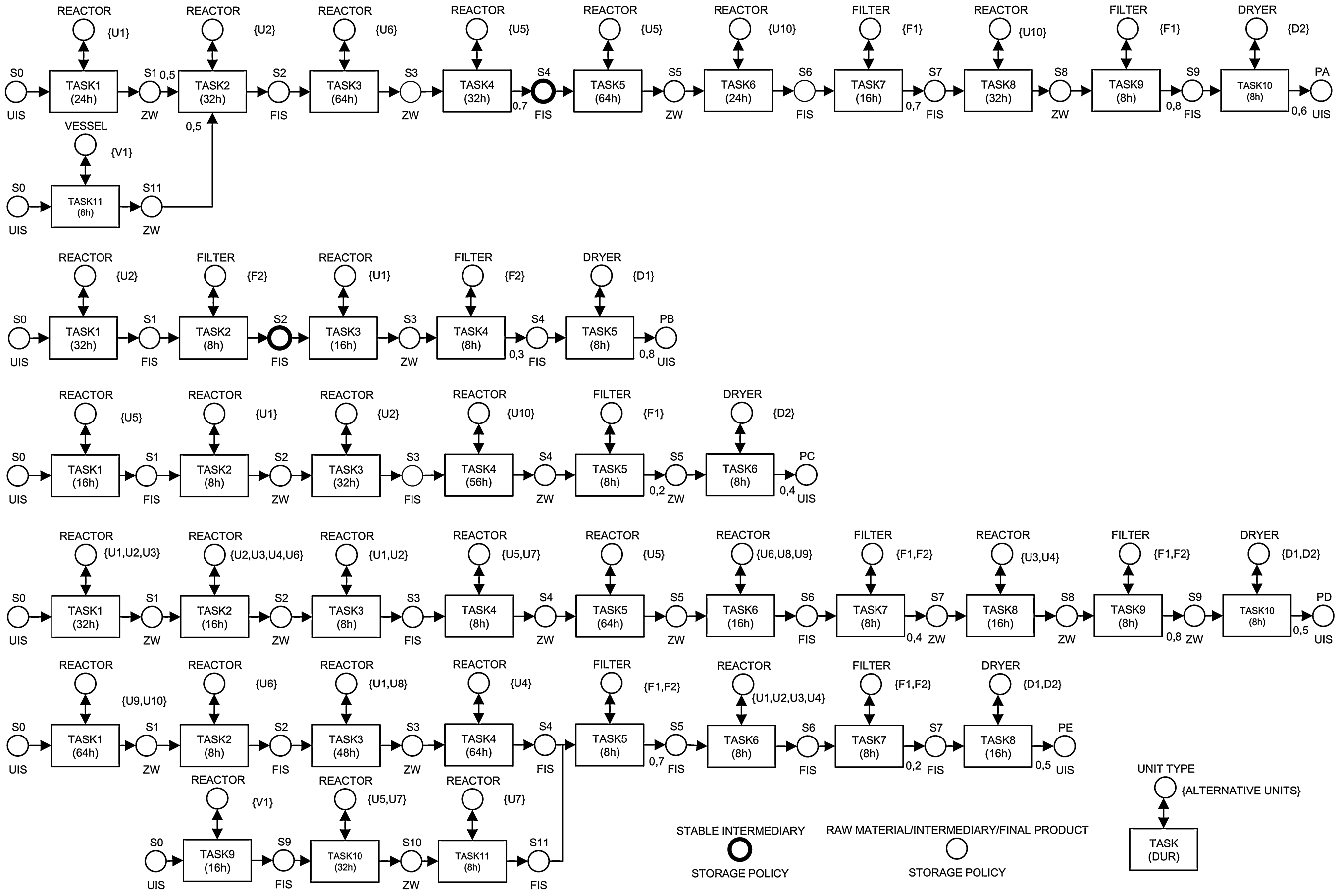


Fig. 1. Recipes of products PA, PB, PC, PD and PE.

we determine the campaign schedule for the products that will be manufactured in campaign mode, with regular products being distributed into campaigns that may have one or more products. In the second step, we create campaign tasks for each schedule determined in the first step. These are aggregate tasks that consume and produce resources according to the campaign schedule. In the third step, we run the scheduling model having the campaign tasks of the regular products and the detailed recipes of the non-regular products. Campaign tasks follow the concept of supertasks that were firstly suggested by Zentner et al. (1994) and Bassett et al. (1996a,b), and more recently by Moniz et al. (2012).

4.1. Step 1 – Determination of the campaign schedule

One possible approach to derive the campaign schedule is to run a periodic schedule formulation as suggested by Shah et al. (1993). In this case, the periodic schedule consists in a plan in which tasks are executed with a cyclic pattern. Processing units will have a cyclic operation as well. This schedule can be repeated successively until the demand is satisfied, assuming that periodic schedules can be derived and applied during a long time horizon, under stable operation and product demand (Shah et al., 1993). Cyclic scheduling implies the existence of two distinct time periods: the startup and shutdown phases. The former is related to the initial schedule that produces the intermediaries needed for the periodic schedule and the latter is related to the final schedule required for the conclusion of the production of all remaining intermediaries.

In practice, the periodic scheduling is a valid approach under the following assumptions. In the cases where the products are produced during long time horizons and the schedule can be replaced by a shorter and cyclic schedule that is repeated until the fulfillment of the demand. Products should have well defined recipes and few alternative production routes. If recipes have several alternative production routes, the tasks-units assignment will tend to vary as well. Therefore, it would be preferable to derive schedules in which the task-unit assignment is not limited to a repeating pattern initially determined. In the practical case of recipes having several alternative units per task, it would be desirable to use the task-unit assignment flexibility across the schedule, instead of using a repeating pattern during the entire time horizon.

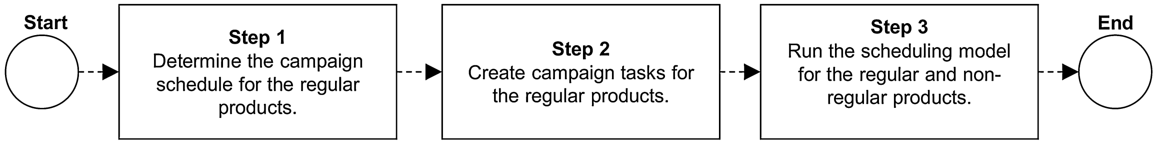

Start
Step 1
Determine the campaign schedule for the regular products.
Step 2
Create campaign tasks for the regular products.
Step 3
Run the scheduling model for the regular and non-regular products.
End

Fig. 2. Suggested approach for regular and non-regular production scheduling.

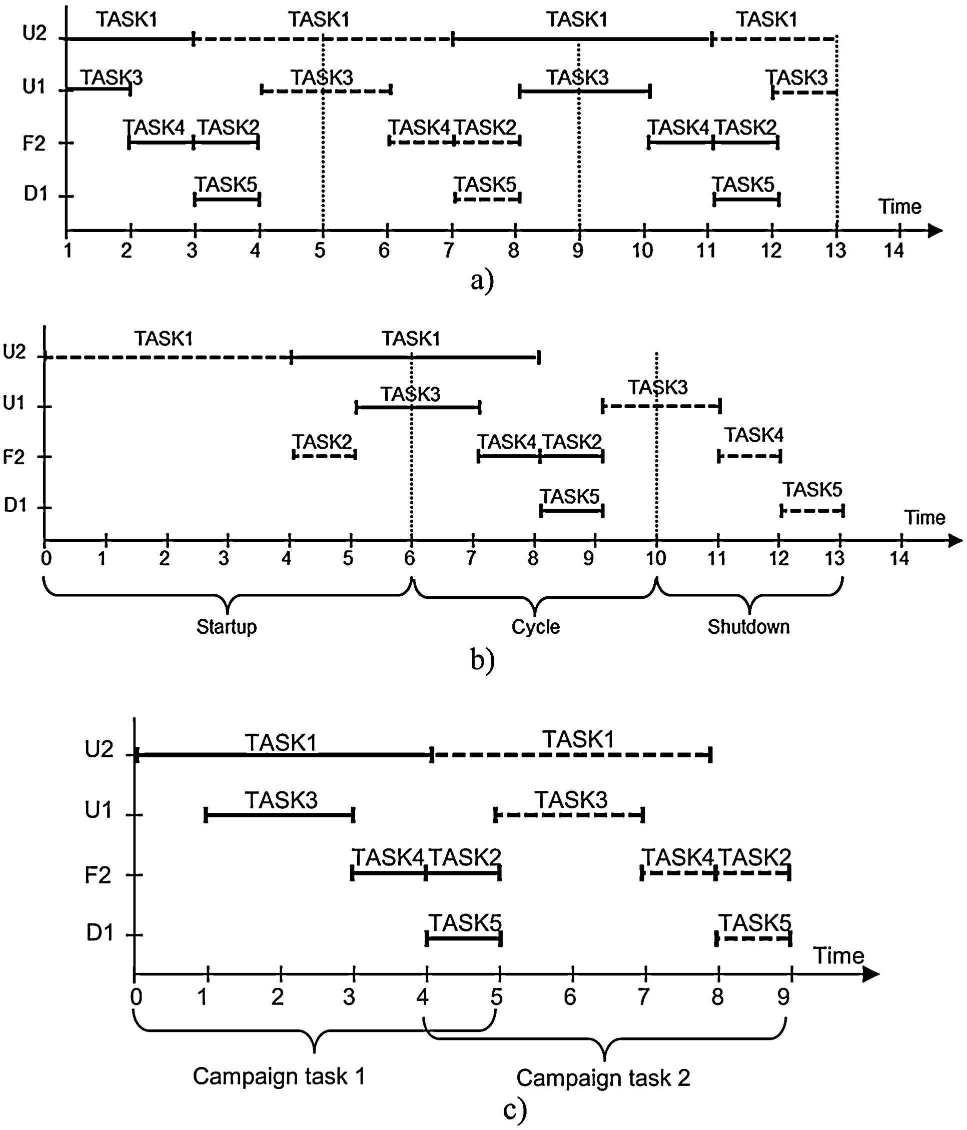


Fig. 3. (a) cyclic schedule of PB (T = 4); (b) cyclic schedule of PB considering the startup and shutdown phases; (c) scheduling of PB considering campaign tasks.

Several advantages can be pointed out to the periodic scheduling. The suggested schedule is easier to implement due to the repetitive pattern of the tasks execution. Moreover, the computational burden of solving a large and non-periodic scheduling problem can be avoided by solving a smaller periodic scheduling problem.

To overcome the assumption of stable production demands and to avoid startup and shutdown effects an alternative approach is used in this work. A non-cyclic schedule can be derived assuming that the storable intermediaries are available in the beginning of the schedule execution and that are replaced in the same quantity when the schedule finishes. This schedule can be modeled using a campaign task that can be repeated successively during the scheduling horizon to satisfy the product demand. This allows the execution of campaign tasks without the need of startup and shutdown periods, being the overall schedule more responsive. The major disadvantage of this approach is the fact that storage costs of the intermediaries tend to be higher, this representing a tradeoff between schedule responsiveness and storage costs. Note that, although the schedule formulation being used is non-cyclic, this approach retrieves schedules in which tasks are executed with a specific cycle, thus we can still call this a cyclic schedule. This concept is better explained later on.

The storable intermediaries are specific characteristic of each recipe since they depend on the chemical stability and storage conditions of the material. For example, in Fig. 1, the storable intermediaries are identified by the bold states.

As it can be seen in Table 1, the characteristics of regular production are appropriate for a cyclic schedule operation. Regular products have stable recipes, the assignment of the processing units to tasks tends to be permanent, the demand is known in advance for a long time horizon, and delivery dates are more flexible if compared with non-regular production, suggesting that this kind of production can be managed through a make-to-stock policy.

In this context, the most common objective function is the maximization of the net production over the cycle time under consideration. However, objectives such as maximizing the average profit or minimizing costs can be used too. In this work, we have used net production as the objective because we have assumed that campaigns have single products and that the control over the schedule production rate is a relevant indicator for measuring the schedules performance.

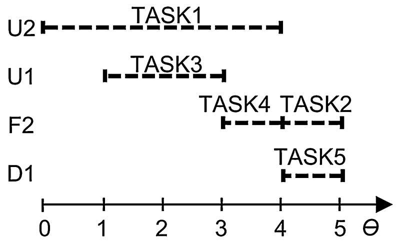


Fig. 4. Campaign task for product PB, with respective resource allocation profile.

4.2. Step 2 – Creation of the campaign tasks

The level of abstraction chosen for modeling recipes will have a direct impact on the model size and therefore on its applicability. High detailed recipes will conduct to more exact models that are however more difficult to tackle computationally. On the other hand, less detailed recipes result into simpler models, and those are easier to handle. The strategy followed in this work exploits the problem structure as described above. Thus, regular products are modeled using campaign tasks. These are aggregate tasks that model the cycle-time and resources allocation/release profile of the schedule determined in step 1. Instead of having a detailed schedule that considers all resources and tasks, we have created a single task for modeling the entire schedule. In this way, many resources and tasks that are considered in step 1 can be disregarded. This reduces the model size in step 3, in terms of the number of binary and continuous variables and constraints.

For example, product PB requires five tasks and has a total of four intermediaries (see Fig. 1), and from these only the intermediary S2 is storable. Using a cyclic schedule formulation any schedule having a cycle time equal to four ($T=4$) will serve the purpose (see Fig. 3a). Nevertheless, the implementation of this schedule requires startup and shutdown phases, as shown in Fig. 3b. On the contrary, using the non-periodic formulation presented in Section 5.1, startup and shutdown phases are not required, since materials availability is ensured by the campaign task (see Fig. 3c). In this way, to produce two batches of product PB the cyclic scheduling requires 13 time intervals, while if campaign tasks are used 9 time intervals are sufficient.

The corresponding campaign task is depicted in Fig. 4 and will have a length of 5 time intervals ($\theta=5$). Unit U2 is allocated to task TASK1 at the beginning of the campaign task execution ($\theta=0$). Task TASK3 is executed one time interval after in unit U1 and consumes the previously stored intermediary S2, and at ($\theta=4$) tasks TASK2 and TASK5 are executed to replenish the intermediary S2 and to produce product PB at ($\theta=5$), respectively. It can be verified that the cycle of the periodic scheduling is 4 shifts and that the campaign task takes 5 shifts. However, since the campaign task allows for superposition of 1 shift the resulting throughput time of product PB is also equal to 4 shifts. The superposition of campaign tasks is then allowed as can be seen in Fig. 3c and is defined in the mathematical formulation through the coefficient units' allocation/release of the RTN formulation.

This approach does not imply any reduction of the solution space of the schedule obtained in step 1, since processing units allocation/release to/from tasks is transposed to the campaign task respecting the sequencing obtained. Moreover, it is important to make a distinction between materials that need and do not need inventory control. If we need to have control over the availability of certain materials, for example stable intermediaries or final products, then these materials need to be modeled in the campaign task. These materials must be storable, to allow the execution of the campaign task without the need of the startup and shutdown phases. All the other materials can be omitted from the campaign task because they are produced and/or consumed within the campaign tasks, and we can simply assume that they are available when required.

4.3. Step 3 – Scheduling model

Finally, the scheduling model in step 3 integrates both production types but with different aggregation levels. Regular production is modeled by campaign tasks, while non-regular production is represented by the detailed recipes. The model used in this step is presented in Section 5.3.

The approach suggested in this work addresses the complex modeling challenge of the scheduling problem, by proposing different decomposition schemes, for different production types, that are typically found in the chemical–pharmaceutical industry.

5. Mathematical formulations

The integrated algorithmic approach described above is based on a set of mathematical formulations that are characterized below.

5.1. Step 1 – Determination of the campaign schedules (CS model)

The campaign schedules are obtained using the non-periodic RTN discrete-time formulation (1)–(8). By assuming that the storable intermediaries are replenished until the end of the schedule (time T) through constraints (4), the resulting schedule can be repeated successively and startup and shut-down phases can be avoided. The production resources R_l include processing units, raw materials, intermediaries and final products ($R_l = E \cup W_l \cup I_l \cup P_l$), where l is the campaign task ($l=1, \ldots, L$). In this way, we can define campaign tasks having different products that share the set of processing units E. The availability of the production resources is given by the resources balance constraints (1). R_{rt} are continuous variables that denote the availability of the resource r at time interval t, while N_{kt} are binary variables that are equal to one if task k starts at time interval t. The amount of resource (processing units) allocated or released by each task is specified by the parameter $\mu_{kr\theta}$, which can take values during the processing time of the task (τ_k). Similarly, materials are consumed and produced at the proportion $\nu_{kr\theta}$ of the task batch size that is modeled through the continuous variables ξ_{kt}. The resources maximum availability is guaranteed by constraints (2). Task batch size variables ξ_{kt} are activated through the binary variables N_{kt} in constraints (3), which also ensure that the task batch sizes are within the capacity limits of the processing units. The set of constraints (4) ensures the intermediaries balance at the end of the schedule. These constraints are essential to guarantee the replacement of the storable intermediaries at the end of the schedule. The net production of the final products over time T is given by constraints (5), where Δ_r is the net production of final product r. And constraints (6) define the production bounds for Δ_r, by imposing minimum and maximum amounts $\Theta_r^{\min}$ and $\Theta_r^{\max}$, respectively. The variables domain is defined in (7).

5.1.1. Constraints

$$R_{rt} = (R_r^{init}|_{(t=0)}, R_{r,t-1}|_{(t>0)}) + \sum_{k \in K_r} \sum_{\theta=0}^{\tau_k} (\mu_{kr\theta} N_{k,t-\theta} + \nu_{kr\theta} \xi_{k,t-\theta}) \quad \forall r \in R_l, \quad t \in H \tag{1}$$

$$0 \le R_{rt} \le R_{rt}^{\max} \quad \forall r \in R_l, \quad t \in H \tag{2}$$

$$V_{ke}^{\min} N_{kt} \le \xi_{kt} \le V_{ke}^{\max} N_{kt} \quad \forall e \in E, \quad k \in K_e^l, \quad t \in H \tag{3}$$

$$R_r^{init} = R_{r,T} \quad \forall r \in I_l \tag{4}$$

$$\Delta_r = R_{r,T} - R_{r0} \quad \forall r \in P_l \tag{5}$$

$$\Theta_r^{\min} \leq \Delta_r \leq \Theta_r^{\max} \quad \forall r \in P_l \tag{6}$$

$$\begin{aligned} &R_{rt} \in \mathbb{R}_+ \quad \forall r \in R_l, \quad t \in H \\ &\xi_{kt} \in \mathbb{R}_+ \quad \forall k \in K_l, \quad t \in H \\ &N_{kt} \in \{0, 1\} \quad \forall k \in K_l, \quad t \in H \\ &\Delta_r \in \mathbb{R}_+ \quad \forall r \in P_l \end{aligned} \tag{7}$$

5.1.2. Selection of the intermediaries

The formulation presented above assumes that the storable intermediaries are given so as to avoid startup and shutdown phases of the campaigns. Alternatively, the storable intermediaries can be determined by the optimization model, assuming that these intermediaries are not raw materials and final products, and that they have an initial amount that is replenished at the end of each campaign. The CS model can be easily modified to account for these requirements. The initial amount of the intermediaries is now given by the decision variable $R_r^{init'}$ that replaces the parameter R_r^{init} at constraints (1) and (4). For processing units, raw materials and final products $R_r^{init'}$ must be equal to the initial availability R_r^{init}, as expressed by (1.1). For the intermediaries, $R_r^{init'}$ is confined by the maximum availability $R_r^{\max}$, see (1.2). Note that, if $R_r^{\max}$ is equal to 0 then this intermediary is not eligible to be a storable intermediary.

$$R_r^{init'} = R_r^{init} \quad \forall r \in E \cup W_l \cup P_l \tag{1.1}$$

$$0 \leq R_r^{init'} \leq R_r^{init} \cdot \lambda_r \quad \forall r \in I_l \tag{1.2}$$

$$\sum_{r \in I_l} \lambda_r \leq a \tag{1.3}$$

To have a limit on the number of storable intermediaries a new binary variable λ_r can be used. So, λ_r is equal to 1 if intermediary r has been selected as storable intermediary. Constraints (1.3) ensure that no more than a intermediaries can be storable.

5.1.3. Objective function

The objective function is the maximization of the production rate and is given by expression (8). Several alternative schedules can be derived by solving the same model for different values of T, with T between *Tmin* and *Tmax*. *Tmin* is equal to the maximum processing time required to produce the stable intermediaries or the final product. *Tmax* is defined as the maximum acceptable duration for the campaign schedule. The selected schedule will give the maximum production Δ_r for each product r. Moreover, in order to calculate the minimum amount of product that can be delivered by each campaign cycle, the model was solved fixing the binary variables N_{kt} determined previously and assuming a minimization version of the objection function (8). The minimum and maximum values of Δ_r represent the production bounds of product r, and are used in step 2 as the minimum and maximum lot size ($L_{rk}^{\min}$, $L_{rk}^{\max}$) of product r at campaign task k.

$$\max \frac{1}{T} \sum_{r \in P_l} \Delta_r \tag{8}$$

5.2. Step 2 – Creation of the campaign tasks

Campaign tasks now are created taking as a basis the time T chosen in Step 1. These tasks will consume/allocate and produce/release resources according to the resources/tasks assignment made in Step 1. This approach allows modeling campaigns, as they are viewed as single production tasks, taking advantage of the uniform representation of the RTN formulation. Fig. 5 depicts the campaign tasks for regular products PA and PB. The lot size is between the maximum ($L_{rk}^{\max}$) and minimum ($L_{rk}^{\min}$) allowable production taking into account the requirements of the recipes. The consumption and production proportions of the materials in the campaign tasks are calculated through the ratio *amount of material required/amount of final product*. Therefore, the campaign task of PA has a processing time of 152 h which results in a net production of 235 kg at the maximum lot size and PB campaign task has a processing time of 40 h and delivers 120 kg. In step 3, it is used a RTN non-periodic formulation for scheduling all products. In order to account for lot-size-dependent processing times and also alternative units, a piecewise approximation can be done by creating multiple instances of the campaign tasks. The new campaign tasks will have different lot-size intervals that correspond to different processing times and/or units.

5.3. Step 3 – Detailed scheduling model (DS model)

Finally, a single schedule with campaign and short-term products is built by using constraints (9)–(18) and objective function (19). Again, we use as basis the RTN formulation.

In order to model sequence-dependent changeovers, the product index p is considered in the resource availability R_{pet}^{DS} continuous variable. Thus, R_{pet}^{DS} variables give the processing unit e availability for product p at time interval t. The changeover tasks are defined by $C_{epp't}^{DS}$ binary variables that are equal to 1 if a changeover task occur on the processing unit e between products p and p' at time interval t. The assignment/sequencing N_{kt}^{DS} variables and the batch size ξ_{kt}^{DS} variables are similar to the CS model. The superscript DS in the variables and sets indicate that they refer to the detailed model.

In this way, the resources balance constraints (9) determine the availability of the processing units for each product and time interval. The unit availability R_{pet}^{DS} is equal to the availability in the previous time interval $R_{pe,t-1}^{DS}$ plus the availability resulting from the unit's allocation/release to/from the production or changeover tasks at time interval t. The production tasks coefficients $\mu_{ke\theta}$ define the unit e allocation/release done by task k at time θ relative to the start of the task, and the changeovers coefficients $\alpha_{ep'p''p\theta}$ give the allocation/release of unit e from product p' to p'' being the product p held by the unit e at time θ relative to the start of the changeover task. Constraints (10) do the initial assignment of processing units to products. Since constraints (9) ensure that no processing units are eliminated or created, no resource bounds on these variables are required.

Constraints (11) are needed to determine the materials availability R_{mt}^{DS}. The set material resources M includes raw materials, intermediaries and products, $M = W \cup I \cup P$, of both campaign and short-term products. The coefficient $\nu_{km\theta}$ defines the proportion of materials consumed and produced of the batch size ξ_{kt}^{DS}. The continuous variables Π_{mt}^{DS} express the deliveries of the products at each the time interval t. We assume that Π_{mt}^{DS} will always have non positive values, thus no material receipts are expected to occur during the scheduling horizon H^{DS}. Constraints (12) define the minimum and maximum materials availability allowed for each time interval. Constraints (13) ensure that the batch size ξ_{kt}^{DS} is between the minimum $V_{ke}^{\min}$ and maximum $V_{ke}^{\max}$ allowed capacities of the processing units e and are just defined for the non-regular products, while constraints (14) define the minimum and maximum lot size of the campaign tasks (regular products).

On the demand side, the variables Π_{mt}^{DS} must be equal to zero for all materials, except for final products, see constraints (15), and at the time intervals different of the delivery dates t_d, see constraints (16). The minimum and maximum amount of each delivery is specified by constraints (17). Production requirements were modeled

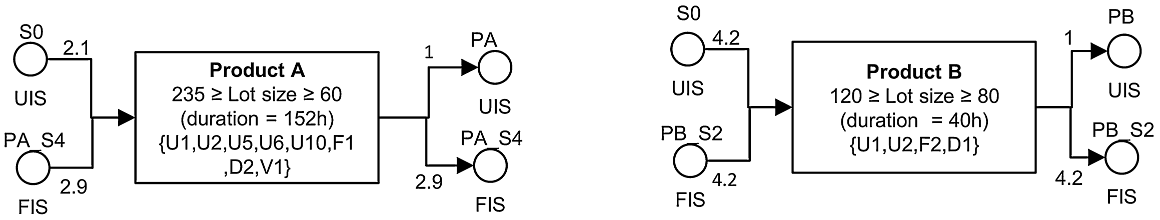


Fig. 5. Campaign tasks for the regular products PA and PB.

as "soft constraints" to avoid schedule infeasibilities. The missing deliveries are expressed by the continuous variables $\Pi_{md}^{DSslack}$, which are penalized in the objective function through coefficient a_m. Practice demonstrates that this is often the case when dealing with medium and long term scheduling. Finally, expressions (18) express the variables domain.

5.3.1. Constraints

$$R_{pet}^{DS} = (R_{pe}^{init}|_{(t=0)}, R_{pe,t-1}^{DS}|_{t>0}) + \sum_{(p',k)\in KP_e} \sum_{\theta=0}^{\tau k} \mu_{ke\theta} N_{k,t-\theta}^{DS}$$

$$+ \sum_{p'\in p_e} \sum_{p''\in p_e} \sum_{\theta=0}^{c_{ep'p''}} \alpha_{ep'p''p\theta C_{ep'p'',t-\theta}^{DS}} \quad \forall p \in P, \quad e \in E, \quad t \in H^{DS} \tag{9}$$

$$\sum_{p\in P_e} R_{pe}^{init} \le R_e^{init} \quad \forall e \in E \tag{10}$$

$$R_{mt}^{DS} = \left(R_m^{init}\Big|_{(t=0)}, R_{m,t-1}^{DS}\Big|_{(t>0)} \right) + \sum_{k\in K_m} \sum_{\theta=0}^{\tau k} \nu_{km\theta} \xi_{k,t-\theta}^{DS}$$

$$+ \Pi_{mt}^{DS} \quad \forall m \in M, \quad t \in H^{DS} \tag{11}$$

$$0 \le R_{mt}^{DS} \le R_{mt}^{\max} \quad \forall r \in M, \quad t \in H \tag{12}$$

$$V_{ke}^{\min} N_{kt}^{DS} \le \xi_{kt}^{DS} \le V_{ke}^{\max} N_{kt}^{DS} \quad \forall e \in E, k \in K_e^{NR}, \quad t \in H^{DS} \tag{13}$$

$$L_{rk}^{\min} N_{kt}^{DS} \le \xi_{kt}^{DS} \le L_{rk}^{\max} N_{kt}^{DS} \quad \forall r \in P, \quad k \in K^R, \quad t \in H^{DS} \tag{14}$$

$$\Pi_{mt}^{DS} = 0 \quad \forall m \in M \backslash P, t \in H^{DS} \tag{15}$$

$$\Pi_{mt}^{DS} = 0 \quad \forall m \in P, t \in H^{DS} \backslash \{t_d\}_{d\in D_m^{DS}} \tag{16}$$

$$Q_{md}^{DS\max} \ge -\Pi_{m,t_d}^{DS} \ge Q_{md}^{DS\min} - \Pi_{md}^{DSslack} \quad \forall m \in P, d \in D_m^{DS} \tag{17}$$

$$\begin{aligned} & R_{pet} \in \mathbb{R}_+ \quad \forall p \in P, \ e \in E, \ t \in H^{DS} \\ & R_{mt}^{DS} \quad \forall m \in M, t \in H^{DS} \\ & \xi_{kt} \in \mathbb{R}_+ \quad \forall k \in K, t \in H^{DS} \\ & \Pi_{md}^{slack} \in \mathbb{R}_+ \quad \forall m \in P, d \in D_m^{DS} \\ & \Pi_{mt}^{DS} \in \mathbb{R}_- \quad \forall m \in M, t \in H^{DS} \\ & N_{kt} \in \{0, 1\} \quad \forall k \in K, t \in H \\ & C_{ep'p''t} \in \{0, 1\} \quad \forall e \in E, p'peP_e, t \in H \end{aligned} \tag{18}$$

5.3.2. Objective function

The objective function is given by expression (19) and maximizes the profit, taking into account the value of the products, inventory costs of the materials and changeovers costs. The last term introduces a penalty cost for missing deliveries.

$$\max \left[\sum_{m\in P} \sum_{d\in D_m^{DS}} -\Pi_{m,td}^{DS}(\nu_m - c_m^{raw}) - \sum_{m\in M} \sum_{t\in H^{DS}} c_m^{sto} R_{mt}^{DS} \right.$$

$$\left. - \sum_{e\in E} \sum_{p\in P} \sum_{p'\in P} \sum_{t\in H^{Ds}} c_e C_{epp't} - \sum_{m\in P} \sum_{d\in D_m^{DS}} \alpha_m \Pi_{md}^{DSslack} \right] \tag{19}$$

6. Solution methods

The DS model can directly be solved using an exact method such as the branch-and-bound (B&B). However, with the increase of the number of resources or the number of tasks or, mainly, with the increase of the time periods (resulting from decreasing the duration of the time intervals or increasing the scheduling horizon), the model would lead to large optimization problems that would hardly be solved by exact methods in acceptable amount of time. Alternatively, decomposition approaches can be applied to obtain satisfactory solutions quickly.

In this work, we have decided to apply a rolling horizon approach based on the works by Dimitriadis et al. (1997) and Erdirik-Dogan and Grossmann (2007), and the reformulation and branching strategy proposed by Velez and Maravelias (2013). The rolling horizon approach considers the detailed scheduling model (DS) and an aggregate planning model (AP), and is applied as depicted in Fig. 6. The algorithm will progressively increase the horizon of the DS model and shrink the horizon of the AP model. The

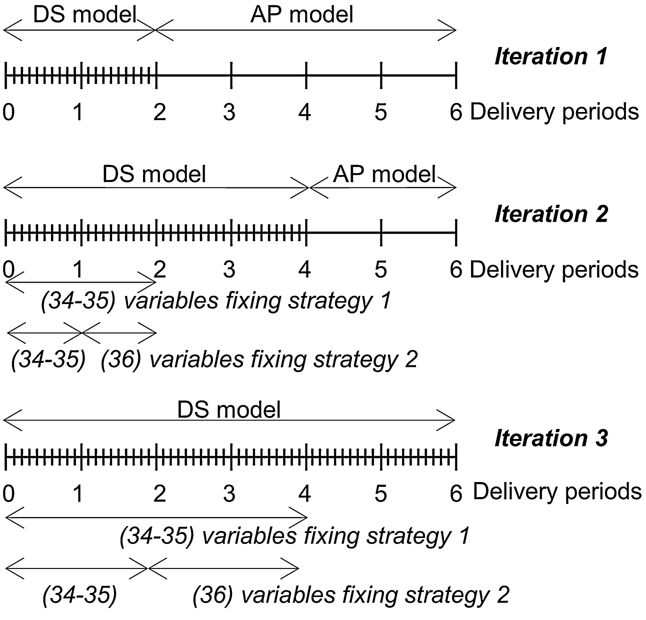


Fig. 6. Rolling horizon approach.

reformulation and branching strategy goal is to improve the performance of the B&B by reducing the symmetry of the scheduling solutions. Several modifications were performed in both methods so as to improve their performance.

6.1. Aggregate planning model (AP model)

The main objectives of the AP model are to obtain a fair estimative of the scheduling solution, at a low computational time, and to trigger adequate production needs at the time interval boundary with the DS model, when running the rolling horizon approach. To achieve that, sequencing and detailed timing variables of the DS model were ignored and the planning horizon was divided in periods having duration of one week. Product demand and the corresponding deliveries take place only at these periods; therefore they are called delivery periods. Note that, the AP model considers the same delivery periods as the DS model.

Although solutions obtained by the AP model cannot be applied because tasks-sequencing are not modeled, the model yields upper bounds on the profit value. The AP model is based on the aggregate planning model proposed by Erdirik-Dogan and Grossmann (2007) and is defined by constraints (20)–(30) and objective function (31). We have considered the continuous variables R^{AP}_{md} that define the availability of material m at delivery period d, the continuous variables ξ^{AP}_{md} that define the total amount of material processed by task k at delivery period d and the continuous variables Π^{AP}_{md} that define the amount delivered of final product m at delivery period d.

Materials balance constraints (20) are defined for all delivery periods and materials. The proportion of material consumed and produced is given by the parameter υ_{km}. Since the detailed timing and sequencing constraints have not been considered, there is no need to model the availability of the processing units. The minimum and maximum materials availability is given by constraints (21); the demand "soft-constraints" are given in (22); and deliveries cannot take place for raw materials and intermediaries, see constraints (23) and (24). Constraints (25) and (26) bound the total amount of material processed by tasks of the non-regular products and of the campaign tasks, respectively. They are similar to constraints (13) and (14) of the DS model, however in the AP model they are required to compute the number of batches of each task (the integer variables N^{AP}_{kd}).

The production capacity is expressed in terms of time available in the processing units by delivery period it^{AP}_{d}. The first summation of constraints (27) defines the total time required by tasks $k \in K_e$ in processing unit e and the second summation accounts for an estimation of the changeovers times. The binary variables Y^{AP}_{ped} determine if product p is produced in unit e at delivery period d and the parameter chg^{AP} is the changeover time, which is assumed to be equal for all products and units. Since the tasks sequence is not known, the expression $\sum_{p \in P}(Y^{AP}_{ped} chg^{AP})$ could lead to an overestimation of the changeovers times in the cases that the unit ends with one product in delivery period d and starts with the same product in delivery period $d+1$. Thus, the third term of constraints (27) is added so as to express the fact that the number of changeovers is equal to the number of products minus one. Constraints (28) and (29) are used to define the variables Y_{ped} and constraints (30) to define the variables domain.

The task processing times is given by the parameter τ_{ke}, but is defined in a different way for the regular and non-regular products. Since the regular products are modeled through campaign tasks, τ_{ke} value is equal to the sum of the processing times of all tasks assigned to unit e in the campaign task. Thus in campaign tasks, τ_{ke} retrieves the total time campaign task k requires from processing unit e. Regarding the non-regular products, the value of τ_{ke} is just determined by the processing time of each task, so $\tau_{ke} = \tau_k$.

6.1.1. Constraints

$$\begin{aligned} R^{AP}_{md} = {} & (R^{AP_{init}}_{m} \Big|_{d=0}, \; R^{AP}_{m,d-1} \Big|_{d>0}) \\ & + \sum_{k \in K_m} \upsilon_{km} \xi^{AP}_{kd} + \Pi^{AP}_{md} \quad \forall m \in M, d \in D^{AP} \end{aligned} \tag{20}$$

$$0 \leq R^{AP}_{md} \leq R^{\max}_{md} \quad \forall m \in M, d \in D^{AP} \tag{21}$$

$$Q^{DS\max}_{md} \geq -\Pi^{AP}_{md} \geq Q^{DS\min}_{md} - \Pi^{APslack}_{md} \quad \forall m \in P, d \in D^{AP} \tag{22}$$

$$\Pi^{AP}_{md} = 0 \quad \forall m \in M \backslash P, d \in D^{AP} \tag{23}$$

$$\Pi^{APslack}_{md} = 0 \quad \forall m \in M \backslash P, d \in D^{AP} \tag{24}$$

$$V^{\min}_{ke} N^{AP}_{kd} \leq \xi^{AP}_{kd} \leq V^{\max}_{ke} N^{AP}_{kd} \quad \forall e \in E, k \in K^{NR}_{e}, d \in D^{AP} \tag{25}$$

$$L^{\min}_{k} N^{AP}_{kd} \leq \xi^{DS}_{kd} \leq L^{\max}_{k} N^{AP}_{kd} \quad \forall k \in K^{R}, d \in D^{AP} \tag{26}$$

$$\sum_{k \in K_e} N^{AP}_{kd} \tau_{ke} + \sum_{p \in P} Y^{AP}_{ped} chg^{AP} - chg^{AP} \leq it^{AP}_{d} \quad \forall e \in E, d \in D^{AP} \tag{27}$$

$$N^{AP}_{kd} \geq Y^{AP}_{ped} \quad \forall e \in E, (p,k) \in KP_e, d \in D^{AP} \tag{28}$$

$$N^{AP}_{kd} \leq \left\lfloor \frac{it^{AP}_{d}}{\tau_{ke}} \right\rfloor Y^{AP}_{ped} \quad \forall e \in E, (p,k) \in KP_e, d \in D^{AP} \tag{29}$$

$$\begin{aligned} & R^{AP}_{md} \in \mathbb{R}_{+} \quad \forall m \in M, d \in D^{AP} \\ & \xi^{AP}_{kd} \in \mathbb{R}_{+} \quad \forall k \in K, d \in D^{AP} \\ & \Pi^{slack}_{md} \in \mathbb{R}_{-} \quad \forall m \in M, d \in D \\ & \Pi^{AP}_{md} \in \mathbb{R}_{-} \quad \forall m \in M, d \in D^{AP} \\ & N^{AP}_{kd} \in \mathbb{Z}_{+} \quad \forall k \in K, d \in D^{AP} \\ & Y^{Ap}_{ped} \in \{0,1\} \quad \forall p \in P, e \in E, d \in D^{AP} \end{aligned} \tag{30}$$

6.1.2. Objective function

The objective function (23) aims at maximizing the profit and is similar to the objective function of the DS model, differing only in the time and tasks sequencing aggregation.

$$\begin{aligned} \max \Bigg[& \sum_{m \in P} \sum_{d \in D^{AP}_{m}} -\mathit{\Pi}^{AP}_{md} (\upsilon_m - c^{raw}_{m}) - \sum_{m \in M} \sum_{d \in D^{AP}} c^{sto}_{m} R^{AP}_{md} \\ & - \sum_{e \cup E} \sum_{p \in P} \sum_{d \in D^{AP}} c_e Y_{ped} - \sum_{m \in P} \sum_{d \in D^{AP}_{m}} \alpha_p \Pi^{DSslack}_{md} \Bigg] \end{aligned} \tag{31}$$

6.2. Rolling horizon (RH approach)

The RH approach is defined by the DS model constraints for the detailed scheduling horizon and by the AP model constraints for the aggregate planning horizon. The objective is to maximize the profit given by the sum of the objective functions of the two models.

Fig. 6 depicts three iterations of the RH approach considering a scheduling horizon of six weeks and a rolling horizon window of two weeks. In each iteration, the task-unit assignment binary variables N^{DS}_{kt} determined in the previous iteration are fixed. By fixing those variables the computational complexity of the DS is reduced while some flexibility is kept on the batch size continuous variables ξ^{DS}_{kt}. In the last iteration, the DS model is applied to the entire scheduling horizon. Two different fixing strategies are tested. This will be explained in detail below.

An important choice in this approach is the length of the scheduling horizon (rolling horizon window) that the DS model should consider. This length cannot be too large as it would result in prohibitive solution times of the DS model, and it cannot be too small as it is limited by the production lead time of the products.

An additional set of constraints is added to link both models. Constraints (32) impose that the materials available at the end of the detailed scheduling horizon are equal to the initial amount of materials available for the AP model. Constraints (33) enforce that no task is executed in the DS model if it cannot be finished. These constraints are important to ensure feasibility in the intervals boundaries between the DS and AP models, by blocking the occurrence of tasks that may lead to overproduction, as explained by Dimitriadis et al. (1997).

6.2.1. Constraints

$$R^{DS}_{mT^{DS}} = R^{APinit}_{m} \quad \forall m \in M \tag{32}$$

$$\sum_{t=T^{DS}-\tau_k}^{T^{DS}-1} N^{DS}_{kt} = 0 \quad \forall k \in K \tag{33}$$

$$N^{DS}_{kt} = 1 \quad \forall k \in K^{fixed}, t \in H^{DSfixed} \tag{34}$$

$$C^{DS}_{epp't} = 1 \quad \forall e \in E^{fixed}, p, p' \in P^{fixed}, t \in H^{DSfixed} \tag{35}$$

$$\sum_{k \in A_{k'}} \sum_{t \in H^{DSfixed}} N^{DS}_{kt} \geq NT^{fixed}_{k'} \quad \forall k' \in AK \tag{36}$$

6.2.2. Variables fixing strategies

As mentioned, two distinct strategies are followed regarding variables fixing (see Fig. 6). *Strategy 1* is similar to the approach followed by Dimitriadis et al. (1997) and Erdirik-Dogan and Grossmann (2007). Here, the binary variables N^{DS}_{kt} and $C^{DS}_{epp't}$ that are equal to 1 in each iteration of the RH are fixed in the next iteration through constraints (34) and (35). Additionally, we proposed a mixed approach, *Strategy 2*, which determines, in each iteration of the RH, the number of times a task runs NT^{fixed}_{k} in the DS model. In the right-hand side of expression (36) the parameter NT^{fixed}_{k} gives the number of tasks occurrences grouped by alternative tasks. The set AK gives the group of tasks, while set $A_{K'}$ gives the tasks considering the existing alternative processing units to task k'. Then, in the following iteration of the RH, constraints (34) and (35) are applied in the first time intervals, while constraints (36) are applied in the last time intervals of the DS model, as shown in Fig. 6. In this way, *Strategy 2* fixes the binary variables in beginning of the scheduling horizon H^{DS} and allows for some flexibility on the task-unit assignment at the end of this time horizon, where there is the link with the AP model.

6.3. Reformulation and branching strategies

Velez and Maravelias (2013) studied the scheduling problem and proposed a reformulation for the MILP model that considers new integer variables NT^{DS}_{k} for determining the number of times task k runs. The authors demonstrated that giving higher branching priority to the NT^{DS}_{k} variables lead to the elimination of many symmetric solutions and improved the computational performance of the scheduling model. To account for this approach, constraints (37) and (38) are added to the DS model. Moreover, since the DS model accounts for sequence-dependent changeovers, we propose new integer variables NC^{DS}_{p} to determine the number of changeovers associated to product p. These new variables are defined by constraints (39).

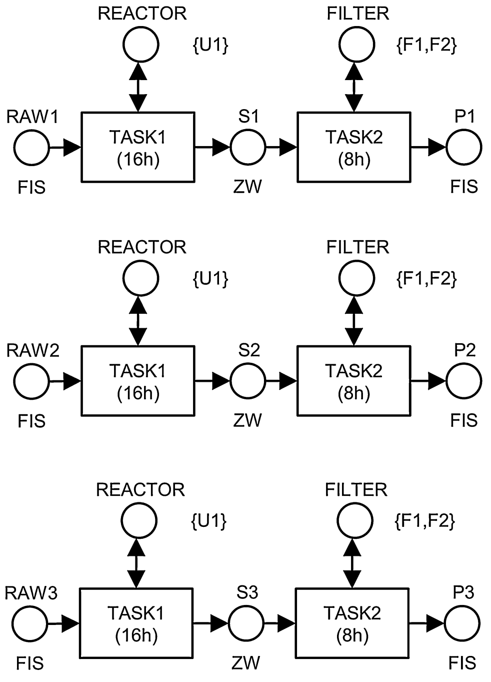


Fig. 7. Product recipes for the illustrative example.

6.3.1. Constraints

$$\sum_{t \in H^{DS}} N^{DS}_{kt} = NT^{DS}_{k} \quad \forall k \in K \tag{37}$$

$$0 \leq NT^{DS}_{k} \leq \left\lfloor \frac{T^{DS}}{\tau_k} \right\rfloor \quad \forall k \in K \tag{38}$$

$$\sum_{e \in E} \sum_{p' \in P} \sum_{p'' \in P} \sum_{t \in H^{DS}} C^{DS}_{ep'p''t} = NC^{DS}_{p} \quad \forall p \in P : (p' = p \vee p'' = p) \tag{39}$$

7. Results

In this section, we propose to solve the illustrative example depicted in Fig. 7 and a real case-study from a chemical–pharmaceutical industry shown in Fig. 1. The proposed algorithm for regular and non-regular production scheduling and the solution methods are tested for several time horizons (4, 8 and 12 weeks). Although scheduling scenarios using campaign tasks cannot be directly compared with scenarios that consider the detailed recipes, since they target different scheduling solutions, we extensively compare both scenarios so as to evaluate the impact of the cyclic operation in the schedules.

The formulations used are summarized in Table 2 and were implemented using ILOG/CPLEX version 12.5.1, running on an Intel Xeon at 3.33 GHz machine with 24 GB of RAM. We test three reformulations of the DS model. DS1 and DSp1 models account for the reformulation and branching priority as proposed by Velez and Maravelias (2013). DS2 model includes the reformulation with the NT^{DS}_{k} and NC^{DS}_{p} integer variables and giving no branching priority. Finally, RH1 implements the variables fixing strategy 1 and RH2 considers the variables fixing strategy 2. In strategy 2, constraints (34) and (35) are applied to the detailed scheduling horizon minus the last week, while constraints (36) are applied over the last week. For example, in the 8 weeks scheduling and assuming a rolling horizon window of 3 weeks, in the second iteration of the RH constrains (34) and (35) are applied in the two first weeks and constrains (36) are applied in the third week.

Table 2
Formulations.

Model	Description	Formulation
AP	Aggregate planning model	(20)–(31)
DS	Detailed scheduling model	(9)–(19)
DS1	Detailed scheduling model with reformulation 1	(9)–(19), (37) and (38)
DSp1	Detailed scheduling model with reformulation 1 and branching priority on the variables NT_k^{DS}	(9)–(19), (37) and (38)
DS2	Detailed scheduling model with reformulation 2	(9)–(19) and (37)–(39)
RH1	Rolling horizon with variables fixing strategy 1	(9)–(18), (20)–(30) and (32)–(35)
RH2	Rolling horizon with variables fixing strategy 2	(9)–(18), (20)–(30) and (32)–(36)

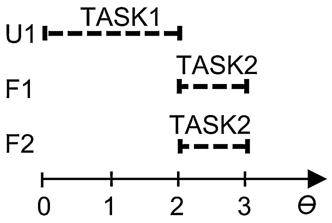


Fig. 8. Illustrative example: campaign task structure.

7.1. Illustrative example

Here we solve a scheduling problem of reduced size, where 3 products requiring each, one reaction task and one filtering task, are considered. The reaction tasks take 16 h and can only be executed by reactor U1, while filtering tasks take 8 h and have two suitable filters, F1 and F2 (see Fig. 7). Since the recipes of the products have a similar structure, they can be represented by similar campaign tasks, as depicted in Fig. 8. The maximum capacity of unit U1 is 5 tons, and of the filters is 3 tons. Raw materials and final products have finite intermediate storage (FIS) and intermediaries follow a zero-wait storage policy (ZW). Products P1, P2 and P3 economic values are 10, 20 and 15 monetary units (m.u.); the raw material costs are 5, 3 and 6 m.u.; and the storage costs 0.05, 0.08 and 0.04, respectively. The demand data is shown in Table 3. Since all the processing times of the tasks are multiple of 8, the time periods were assumed to have a fixed duration of 8 h. The sequence-dependent changeover is of 24 h and equal to all three products. The missing delivery costs a_m are twice the value of the products.

The numerical results shown in the tables present the following data: the model used; the scheduling time horizon; the number of integer and continuous variables and constraints; the number of nodes and iterations; the value of the linear relaxation of the MILP; the integrality gap; the objective function value and the computational time required for solving the instance. Regarding the RH approach the data shown is related to the last iteration, with exception of the CPU time column that displays the total time required by the algorithm.

7.1.1. 4 weeks scheduling

The solution statistics of the four weeks scheduling problem are given in Table 4. As can be seen, all models with the exception of the AP model obtained the optimal solution of 1962.3 m.u. The RH1 required less CPU time than the other models, obtaining the optimal solution in just 11.9 CPU seconds (assuming that no campaign tasks are used). Note that, RH1 and RH2 have rolling horizon windows equal to 2 delivery periods. Moreover, results show that DSp1 model required more than twice the CPU time of DS, DS1 and DS2 models, having also higher number of nodes and iterations.

Using campaign tasks for the three products, the instance size reduced as well as the CPU time needed to solve the problem (see Table 5). Again the RH1 had the best performance, obtaining the optimal solution in just 5.4 CPU seconds.

7.1.2. 8 weeks scheduling

With the increase of the scheduling horizon to eight weeks and assuming no campaign tasks, none of the DS models proved optimality in the time limit of 3600 CPU seconds (see Table 6). DS2 requiring just three more binary variables and constraints than DS1, performs better computationally.

Assuming campaign tasks the instance became easier to solve and DS, DS1 and DS2 models proved optimality within the 3600 CPU seconds. Additionally, results show that the reformulation DS2 had better performance than the DS, DS1 and DSp1 models, and that it seems preferable to use the default CPLEX branching priority, instead of giving priority to the NT_k^{DS} variables. In both instances DSp1 had the worst performance.

In Fig. 9, it is represented the CPU times and objective function values assuming that recipes are aggregated using campaign tasks. Concerning just the computational time, the RH is certainly the most competitive method. While DS2 required 956.2 CPU seconds to obtain a solution of 4121.5 m.u., RH2.2 (rolling horizon window equal to 4 delivery periods) just took 63.7 CPU seconds to obtain a solution with profit equal to 4112.2 m.u. (see Tables 7 and 8). Nevertheless, results indicate that RH approach is very dependent on the definition of the variables fixing strategy and the length of the rolling horizon window. As shown in Fig. 9, for the same rolling

Table 3
Demand in tons for the illustrative example.

Weeks	1		2		3		4		5		6	
Product	min	max	min	max	min	max	min	max	min	max	min	max
P1	5	20	20	30	5	40	0	0	5	15	0	0
P2	0	20	10	10	10	30	20	30	10	30	15	30
P3	0	30	0	0	10	30	10	20	5	10	5	20
Weeks	7		8		9		10		11		12	
Product	min	max	min	max	min	max	min	max	min	max	min	max
P1	0	0	20	30	0	20	15	25	20	40	10	20
P2	5	5	10	40	10	60	5	10	5	15	10	20
P3	0	0	15	40	0	0	20	30	0	30	10	20

Table 4
Four weeks schedule (4W = four weeks scheduling horizon).

Model/horizon	Int. variables/cont. variables/constraints	Nodes	Iterations	LP relaxation	Gap (%)	Objective	CPU time (s)
AP/4W	72/145/300	117	673	2397.36	0.01	2119.8	0.7
DS/4W	2295/3337/5646	12,259	2,183,279	2245.55	0.01	1962.3	64.5
DS1/4W	2304/3337/5664	16,731	2,292,919	2245.55	0.01	1962.3	60.6
DSp1/4W	2304/3337/5664	26,301	5,880,359	2245.55	0.01	1962.3	142.1
DS2/4W	2307/3337/5667	13,211	1,750,774	2245.55	0.00	1962.3	50.9
RH1/4W	2295/3337/5703	16	3787	2011.01	0.00	1962.3	11.9
RH2/4W/	2295/3337/5650	10,225	1,261,106	2192.70	0.00	1962.3	30.0

Table 5
Illustrative example: four weeks schedule with campaign tasks.

Model/horizon/aggregation	Int. variables/cont. variables/constraints	Nodes	Iterations	LP relaxation	Gap (%)	Objective	CPU time (s)
AP/4W/C	48/85/204	0	51	2303.09	0.00	2090.0	0.3
DS/4W/C	1785/2317/3600	22,606	1,409,338	2243.73	0.01	1962.3	21.3
DS1/4W/C	1788/2317/3606	10,508	693,224	2243.73	0.01	1962.3	16.9
DSp1/4W/C	1788/2317/3606	53,784	5,283,160	2243.73	0.01	1962.3	55.0
DS2/4W/C	1791/2317/3609	16,471	1,190,734	2243.73	0.01	1962.3	22.3
RH1/4W/C	1785/2317/3623	0	911	2009.23	0.00	1962.3	5.4
RH2/4W/C	1785/2317/3602	12,666	617,703	2191.95	0.01	1962.3	13.9

Table 6
Eight weeks schedule (8W = eight weeks scheduling horizon; C = campaign tasks used).

Model/horizon/aggregation	Int. variables/cont. variables/constraints	Nodes	Iterations	LP relaxation	Gap (%)	Objective	CPU time (s)
AP/8W	144/289/600	3537	27,943	4967.20	0.01	4553.9	1.0
DS/8W	4563/6625/11214	245,719	77,233,056	4631.96	2.78	4111.5	3601.4
DS1/8W	4572/6625/11232	323,250	101,340,287	4631.96	3.09	4104.0	3601.5
DSp1/8W	4572/6625/11232	196,492	88,934,784	4631.96	8.63	3940.7	3601.3
DS2/8W	4575/6625/11235	361,553	109,406,063	4631.96	1.79	4117.5	3601.5
DS/8W/C	3549/4597/7152	1,027,859	115,570,751	4630.48	0.01	4121.5	2165.0
DS1/8W/C	3552/4597/7158	755,217	64,563,162	4630.48	0.01	4121.5	1133.3
DSp1/8W/C	3552/4597/7158	694,648	160,640,734	4630.48	2.01	4121.5	3601.3
DS2/8W/C	3555/4597/7161	639,363	56,872,981	4630.48	0.01	4121.5	956.2

horizon window the RH2 had always better results than RH1 and solutions tend to improve with the increase of the rolling horizon window. The lower profit solution was obtained by RH1/8W/C which is by 5% less than the best solution found. The best solution among the RH methods was retrieved by RH2.2/8W/C that assumes a rolling horizon window of 4 delivery periods and is just by 0.2% inferior to the best solution found.

7.1.3. 12 weeks scheduling

The longest scheduling horizon this paper considers is of 12 weeks. The numerical results are shown in Table 9. The best solution found was obtained by the DS model in 3395 CPU seconds, assuming campaign tasks, with a profit of 6182.8 m.u. (see Fig. 10). The lower profit solution, among the DS models, was retrieved by DSp1. Although the use of campaign tasks leads to a reduction of more than 20% of the number of integer variables, the DS1 and DS2 models could not deliver solutions within an integrality gap of 5% and time limit of 3600 CPU seconds.

Once more the quality of the solutions delivered by the RH approaches strongly depends on the variables fixing strategy and on the length of the rolling horizon window (see Table 10). The lower profit solution was retrieved by RH2 and is by 6% inferior

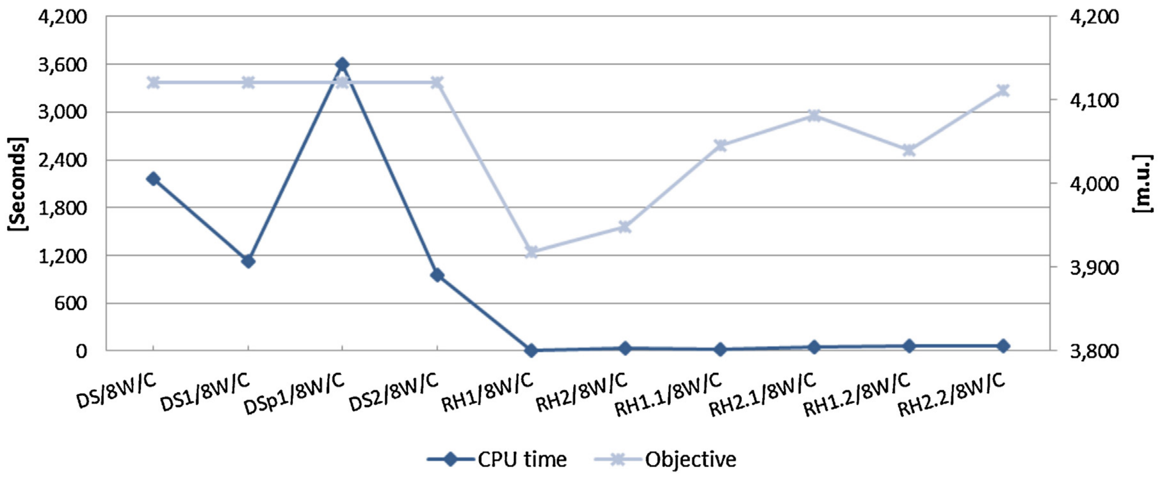


Fig. 9. Eight weeks schedule: models computational times and objective function values (RH1 and RH2 have rolling horizon windows equal to 2 delivery periods; RH1.1 and RH2.1 have rolling horizon windows equal to 3 delivery periods; RH1.2 and RH2.2 have rolling horizon windows equal to 4 delivery periods).

Table 7
Illustrative example: eight weeks schedule with campaign tasks.

Model/horizon/aggregation	Int. variables/cont. variables/constraints	Nodes	Iterations	LP relaxation	Gap (%)	Objective	CPU time (s)
AP/8W/C	96/169/408	0	120	4766.13	0.00	4397.5	0.3
DS/8W/C	3549/4597/7152	1,027,859	115,570,751	4630.48	0.01	4121.5	2165.0
DS1/8W/C	3552/4597/7158	755,217	64,563,162	4630.48	0.01	4121.5	1133.3
DSp1/8W/C	3552/4597/7158	694,648	160,640,734	4630.48	2.01	4121.5	3601.3
DS2/8W/C	3555/4597/7161	639,363	56,872,981	4630.48	0.01	4121.5	956.2
RH1/8W/C	3549/4597/7223	3953	114,599	3991.50	0.01	3918.2	9.7
RH2/8W/C	3549/4597/7202	15,207	749,948	4030.27	0.00	3949.2	36.4

Table 8
Illustrative example: eight weeks schedule with campaign tasks and different rolling horizon windows.

Model/horizon/aggregation	Int. variables/cont. variables/constraints	Nodes	Iterations	LP relaxation	Gap (%)	Objective	CPU time (s)
RH1.1/8W/C	3549/4597/7223	2370	82,848	4146.74	0.00	4046.4	21.9
RH2.1/8W/C	3549/4597/7203	10,083	650,502	4226.94	0.00	4081.3	47.3
RH1.2/8W/C	3549/4597/7198	46,449	2,734,607	4228.89	0.01	4040.5	57.4
RH2.2/8W/C	3549/4597/7180	23,631	2,245,720	4387.78	0.01	4112.2	63.7

Table 9
Illustrative example: twelve weeks schedule.

Model/horizon	Int. variables/cont. variables/constraints	Nodes	Iterations	LP relaxation	Gap (%)	Objective	CPU time (s)
AP/12W	216/433/900	10,528	149,206	7561.25	0.01	6973.1	2.0
DS/12W	6831/9913/16782	219,941	70,867,983	6971.51	6.27	6150.1	3602.0
DS1/12W	6840/9913/16800	185,686	67,646,537	6971.51	7.10	6156.4	3602.2
DSp1/12W	6840/9913/16800	89,491	49,004,502	6971.51	14.71	5805.0	3601.8
DS2/12W	6843/9913/16803	142,460	57,272,458	6971.51	8.16	6062.6	3601.9
RH1/12W	6831/9913/17085	10,610	595,273	6021.66	0.01	5806.1	89.9
RH2/12W	6831/9913/17033	13,688	1,384,723	6072.89	0.00	5806.1	177.4
AP/12W/C	144/253/612	191	1063	7142.23	0.01	6538.7	0.8
DS/12W/C	5313/6877/10704	507,264	87,286,615	6969.79	4.85	6182.8	3395.0
DS1/12W/C	5316/6877/10710	603,294	89,337,815	6969.79	5.64	6100.2	3602.1
DSp1/12W/C	5316/6877/10710	270,009	82,649,307	6969.79	7.77	6070.3	3601.8
DS2/12W/C	5319/6877/10713	402,645	69,102,174	6969.79	6.78	6135.3	3601.8
RH1/12W/C	5313/6877/10817	931	20,258	6045.70	0.01	5888.8	14.3
RH2/12W/C	5313/6877/10800	8708	368,738	6007.75	0.01	5808.4	28.9

to the best solution. The best solution among the RH approaches was obtained in just 132.9 CPU seconds by RH2.2 with a profit of 6162.1 m.u.

Overall, the DS models worked reasonably well. However, with the increase of the scheduling horizon the DS models could not prove optimality. In opposition, the AP model retrieved solutions in very short times but overestimated the production capacity. Looking into the 4, 8 and 12 weeks problems, we can conclude that using campaign tasks improves the computational performance of the models. The RH approaches ran quite fast and obtained good solutions or even optimal solutions. The variables fixing strategy and the length of the rolling horizon window strongly affect the quality of the solutions. In general, the variables fixing strategy 2 requires more CPU time, but obtains better solutions than strategy 1. This is related to the flexibility of constraints (36) that do not impose a fix task-unit assignment for the time intervals of the DS model that interface with the AP model. In other words, task-unit assignment is allowed to change in order to better accommodate the production

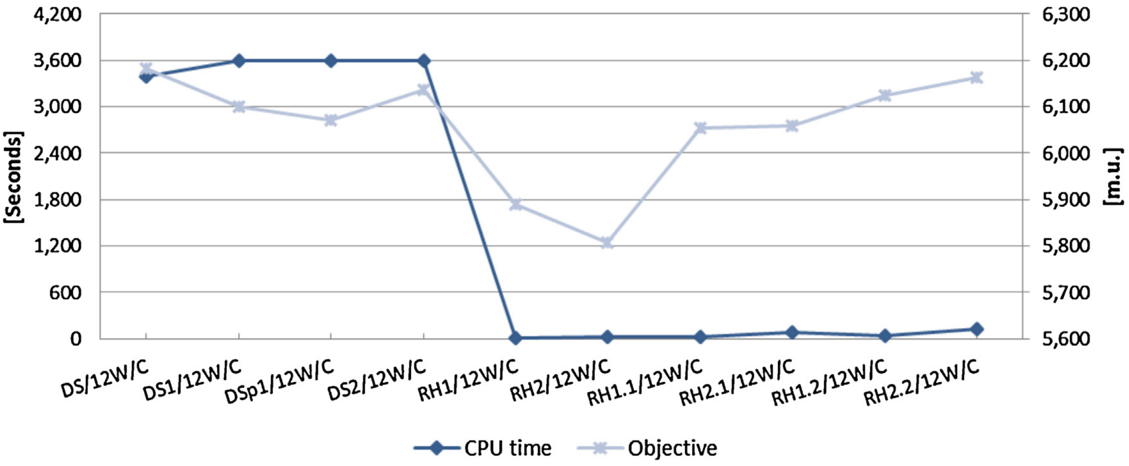


Fig. 10. Twelve weeks schedule: models computational times and objective function values (RH1 and RH2 have rolling horizon windows equal to 2 delivery periods; RH1.1 and RH2.1 have rolling horizon windows equal to 3 delivery periods; RH2.2 and RH2.2 have rolling horizon windows equal to 4 delivery periods).

Table 10
Illustrative example: twelve weeks schedule with campaign tasks and different rolling horizon windows.

Model/horizon/aggregation	Int. variables/cont. variables/constraints	Nodes	Iterations	LP relaxation	Gap (%)	Objective	CPU time (s)
RH1.1/12W/C	5313/6877/10809	10,834	636,166	6298.37	0.00	6053.7	28.4
RH2.1/12W/C	5313/6877/10793	18,487	1,448,855	6348.70	0.01	6059.2	83.4
RH1.2/12W/C	5313/6877/10796	7907	640,003	6397.02	0.00	6124.9	45.2
RH2.2/12W/C	5313/6877/10778	45,417	3,067,954	6441.88	0.01	6162.1	132.9

Table 11
Illustrative example: twelve weeks schedule without changeovers.

Model/horizon/aggregation	Int. variables/cont. variables/constraints	Nodes	Iterations	LP relaxation	Gap (%)	Objective	CPU time (s)
AP/12W	108/433/684	0	50	7651.25	0.00	7394.8	0.2
DS/12W	2277/8386/15261	0	3704	7161.65	0.00	7161.7	0.5

requirements in the next iteration of the RH approach, while the CPU time required to solve the DS model is kept low. We can expect better solutions if a larger RH window is considered, since scheduling decisions are taken considering more data. Nevertheless, it is important to note that with the increase of the rolling horizon window the scheduling problem becomes more difficult to solve. Therefore, RH window size must be defined taking into account the CPU time required to solve the scheduling problem. In order to emphasize the complexity of modeling sequence-dependent changeovers in scheduling problems, we note that the 12 weeks instance without changeovers can be solved to optimality in less than 1 s (see Table 11).

Table 13
Processing units' characteristics for the case study.

Unit	Max. volume	Min. volume
U1	4000	100
U2	6300	150
U3	10000	50
U4	1000	100
U5	1300	50
U6	1000	50
U7	7000	130
U8	4000	80
U9	6300	150
U10	4000	120
F1	800	50
F2	500	30
D1	900	100
D2	600	100
V1	1000	100

Table 14
Products value and raw material costs for the case study (m.u.–monetary units).

	Economic value [m.u.]	Raw material cost [m.u./kg]
PA	10	5
PB	20	3
PC	15	6
PD	30	11
PE	70	36

7.2. Real case study

In this section, we solve a real-world scheduling problem from a chemical–pharmaceutical industry. We consider a multipurpose batch plant producing the 5 products depicted in Fig. 1. These are to be scheduled in a time horizon of up to 3 months and the schedule must give the tasks-unit assignment and sequencing of the regular and non-regular products.

Product PA recipe has 11 tasks and one stable intermediary (PA_S4), requiring a production time of 304 h (sum of the tasks processing times required to manufacture one batch). Product PB has 5 tasks, one stable intermediary and a total production time of 72 h. Product PC has 6 tasks that require a total of 128 h. Product PD has 10 tasks and takes 184 h. Finally, Product PE has 11 tasks and takes 224 h. The objective is the profit maximization. The demand data, processing units' characteristics, and value of the products are depicted in Tables 12, 13, and 14, respectively. The scheduling horizon was discretized into time intervals of 8 h, since all task durations are assumed to be multiples of 8. The sequence-dependent changeover tasks take 24 h and the missing delivery costs a_m are twice the value of the products.

We have considered two different production types: non-regular and regular production. The products that are produced

Table 12
Demand in kg for the case study.

Weeks	1		2		3		4		5		6	
Product	min	max	min	max	min	max	min	max	min	max	min	max
PA	0	0	200	250	0	0	200	300	200	300	200	300
PB	200	360	200	360	200	360	200	360	0	0	200	360
PC	0	0	0	0	70	140	0	0	0	0	0	0
PD	0	0	0	0	180	260	0	0	0	0	0	0
PE	0	0	0	0	0	0	140	200	0	0	0	0

Weeks	7		8		9		10		11		12	
Product	min	max	min	max	min	max	min	max	min	max	min	max
PA	0	0	200	300	0	0	200	300	0	0	200	300
PB	200	360	200	360	0	0	360	480	0	0	360	480
PC	0	0	200	220	0	0	100	120	0	0	0	0
PD	0	0	200	220	0	0	0	0	0	0	200	220
PE	0	0	160	180	0	0	140	160	0	0	140	160

Table 15
Four weeks schedule (4W = four weeks scheduling horizon; C = campaign tasks used).

Model/horizon/aggregation	Int. variables/cont. variables/constraints	Nodes	Iterations	LP relaxation	Gap (%)	Objective	CPU time (s)
DS/4W	15045/18937/33525	198,489	180,675,428	39,524.40	5.52	36,441.2	14,402.9
DS1/4W	15110/18937/33655	92,545	94,107,274	39,524.40	5.06	36,684.7	14,414.2
DSp1/4W	15175/18937/33655	91,782	93,376,456	39,524.40	5.06	36,684.7	14,412.6
DS2/4W	15115/18937/33660	61,821	66,295,967	39,524.40	4.98	36,612.5	7958.6
RH1/4W	15045/18937/33618	8098	2,337,094	37,268.50	4.99	34,206.9	136.2
RH2/4W	15045/18937/33589	33,768	34,907,653	39,128.90	5.03	36,102.3	3610.4
DS/4W/C	13855/15707/27051	106,882	156,144,030	37,946.40	6.19	34,268.0	14,402.7
DS1/4W/C	13906/15707/27153	84,583	172,741,014	37,946.40	10.80	33,177.3	14,402.1
DSp1/4W/C	13957/15707/27153	83,771	170,967,461	37,946.40	10.83	33,177.3	14,402.6
DS2/4W/C	13911/15707/27158	70,537	111,335,068	37,946.40	5.00	34,179.9	9311.8
RH1/4W/C	13855/15707/27079	0	5244	33,470.90	4.78	31,329.4	22.6
RH2/4W/C	13855/15707/27076	1859	727,083	36,505.10	3.37	32,275.3	76.6

in a regular basis have been assigned to specific production lines, while the non-regular products have more flexibility regarding the task-unit assignment. Note that, in the course of the process development of a new drug, the set of alternative processing units available for each task tends to become smaller leading to stable and well-defined recipes. Thus, in its operation the company considers Products PA and PB as regular products that are represented here by the respective campaign tasks (see Fig. 5) and products PC, PD and PE as non-regular products, which are represented by their detailed recipe as depicted in Fig. 1. The case study is solved considering 4, 8 and 12 weeks scheduling horizons scenarios.

In the DS models we have assumed two stopping criteria, the integrality gap of 5% and time limit of 14,400 s, and in the RH approaches, we have considered the integrality gap of 5% and time limit of 3600 s.

7.2.1. 4 weeks scheduling

The results of the 4 weeks scheduling instance are shown in Table 15. The best solution, without campaign tasks, was obtained by DS1 and DSp1 models in 14,400 CPU seconds, with a profit equal to 36,684.7 m.u., while DS2 delivered a solution within 4.98% of the optimum in just 7958.6 CPU seconds. Overall, the RH approaches performed quite well. For example, the solution of RH2 was obtained in just 3610.4 CPU seconds and is by 2% inferior to the best solution found.

Modeling the regular products PA and PB with campaign tasks led to reduction of the profit by 7% to 34,268.0 m.u. The storage costs are higher when using campaigns since it is required keeping stock of the stable intermediaries. This can be interpreted as the cost of the cyclic operation for the regular products. Additionally, note that campaign tasks impose strict tasks sequencing for the regular products, which results in a loss of flexibility when performing scheduling. On the other hand, campaign tasks allow the definition of production lines with cyclic operation, and the control over the inventory of the stable intermediaries, leading to more responsive schedules. The DS model had the best performance among the detailed models, and the RH2 approach obtained a solution within 6% of the best solution, in just 76.6 CPU seconds. Again, results show that RH2 achieved better results when compared with RH1, but at cost of higher CPU time.

Fig. 11 depicts the schedule solution of approach RH2/4W/C, thus assuming that campaign tasks are used to model the regular products PA and PB. As can be seen, two campaigns of PA and four campaigns of PB are scheduled. The first campaign of PB starts in week 1 and runs three campaign cycles. At the end of this week 360 kg of PB are delivered. This campaign is then interrupted to produce one campaign cycle of PA that delivers 235 kg of this product, at the end of week 2. Then, the second campaign of PB starts, having also three cycles and delivering 360 kg of this product in week 2. The third campaign of PB is initiated in week 3 and has three cycles. At the end of this week, 360 kg of PB, 64 kg of PC and 208 kg of PE are delivered. In the last week, the second campaign of PA and the fourth campaign of PB are performed, delivering 235 kg of PA, 360 kg of PB and 200 kg of PE.

7.2.2. 8 weeks scheduling

In the 8 weeks scheduling the RH approaches performed better than the DS models, as can be seen in Fig. 12. The CPU times

Table 16
Case study: eight weeks schedule.

Model/horizon/aggregation	Int. variables/cont. variables/constraints	Nodes	Iterations	LP relaxation	Gap (%)	Objective	CPU time (s)
AP/8W	888/1673/3736	191,528	9,010,675	75,124.20	0.01	73,162.4	234.5
DS/8W	29913/37605/66577	40,970	41,645,010	74,239.90	39.33	52,082.6	14,439.9
DS1/8W	29978/37605/66707	30,153	52,636,580	74,239.90	32.30	54,873.5	14,410.6
DSp1/8W	29978/37605/66707	30,214	52,706,062	74,239.90	32.30	54,873.5	14,410.7
DS2/8W	29983/37605/66712	38,707	57,944,461	74,239.90	45.75	49,799.2	14,409.8
RH1/8W	29913/37605/66821	17,306	18,499,174	64,136.60	5.99	58,683.1	4174.9
RH2/8W	29913/37605/66875	10,514	17,955,601	67,495.30	6.06	62,723.5	7043.4
RH1.1/8W	29913/37605/66813	6190	4,710,039	64,804.60	4.99	60,731.5	920.7
RH2.1/8W	29913/37605/66815	10,820	13,144,695	56,595.80	6.07	52,366.7	7330.9
AP/8W/C	776/1225/3000	73,152	2,939,395	73,629.80	0.01	72,245.6	34.3
DS/8W/C	27547/31183/53719	31,175	51,628,895	70,684.50	34.92	51,748.6	14,434.8
DS1/8W/C	27598/31183/53821	27,502	58,879,606	70,684.50	42.33	49,140.8	14,405.4
DSp1/8W/C	27598/31183/53821	27,274	58,684,118	70,684.50	48.10	47,228.0	14,406.0
DS2/8W/C	27603/31183/53826	22,846	48,608,967	70,684.50	31.24	53,212.2	14,444.7
RH1/8W/C	27547/31183/53861	11,324	9,035,461	61,884.80	5.00	54,195.3	1648.2
RH1.1/8W/C	27547/31183/53878	24,206	19,340,491	63,654.80	4.80	58,025.2	5858.9
RH2/8W/C	27547/31183/53877	20,922	15,976,374	65,922.00	4.98	60,048.9	4020.3
RH2.1/8W/C	27547/31183/53883	27,601	26,477,756	64,161.40	5.99	58,641.7	7493.2

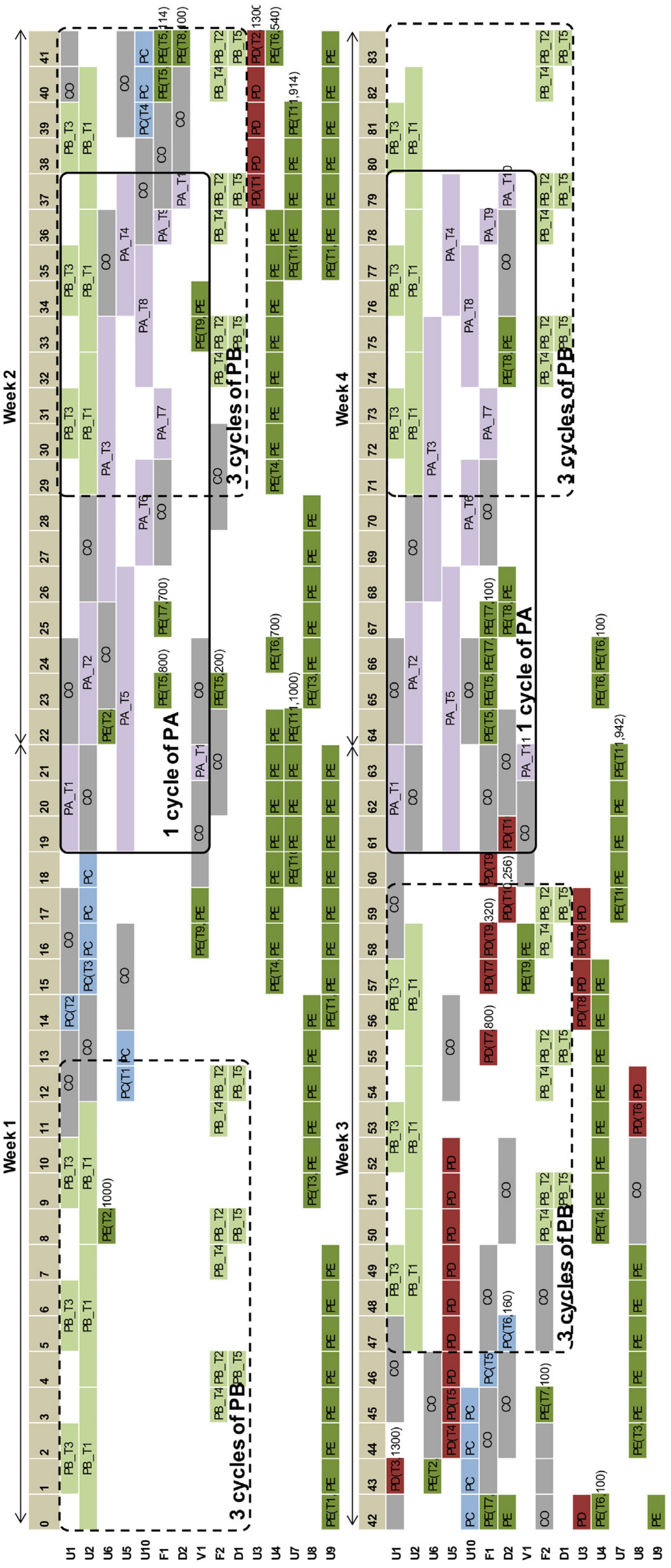


Fig. 11. Four weeks scheduling of regular and non-regular production (model RH2/4W/C).

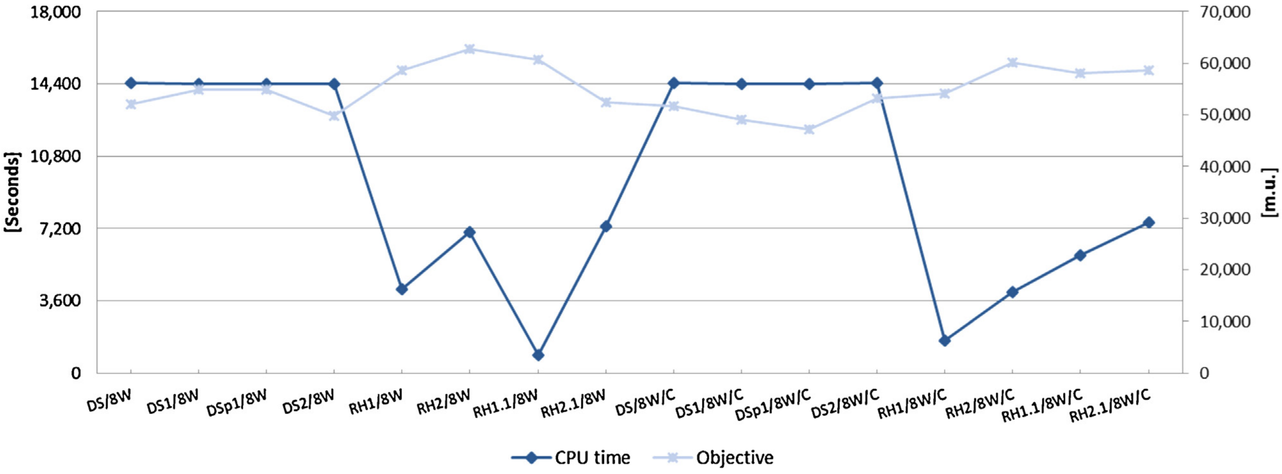


Fig. 12. Eight weeks schedule: models computational times and objective function values (RH1 and RH2 have rolling horizon windows equal to 2 delivery periods; RH1.1 and RH2.1 have rolling horizon windows equal to 3 delivery periods).

Table 17
Case study: twelve weeks scheduling.

Model/horizon/aggregation	Int. variables/cont. variables/constraints	Nodes	Iterations	LP relaxation	Gap (%)	Objective	CPU time (s)
AP/12W	1332/2509/5604	492,694	35,230,261	110,455.00	0.01	107,688.0	471.7
RH1/12W	44781/56273/100159	625	141,187	89,929.90	1.73	87,907.5	1473.9
RH1.1/12W	44781/56273/100106	7125	3,039,419	89,763.10	3.97	85,504.6	2587.2
RH2/12W	44781/56273/100181	1286	1,069,626	95,020.30	2.94	91,140.1	4653.4
RH2.1/12W	44781/56273/100061	8370	6,191,209	96,480.00	3.86	91,909.2	5491.0
AP/12W/C	1164/1837/4500	70,965	3,798,482	108,357.00	0.01	106,446.00	56.63
RH1/12W/C	41239/46659/80744	190	97,528	85,952.10	4.90	81,790.30	2788.05
RH1.1/12W/C	41239/46659/80667	3933	1,112,363	87,456.00	4.86	80,867.70	7555.55
RH2/12W/C	41239/46659/80713	2887	1,287,263	86,198.00	1.70	82,639.40	4419.79
RH2.1/12W/C	41239/46659/80645	5592	2,878,518	86,647.10	2.81	83,801.40	8203.44

of the RH approaches are significantly inferior to the CPU times required by the DS models and the best solutions found in the scenarios with and without campaign tasks were delivered by the RH2 model. Assuming campaign tasks, RH2 reached a profit of 60,048.9 m.u., which is by 5% inferior to the profit considering that no campaign tasks are used. Among the DS models, the time limit of 14,400 CPU seconds was not sufficient to obtain good quality solutions. The use of the reformulation and branching strategies presented in Section 6.3 were not advantageous in this instance, since the resultant integrality gaps were higher than 30% (see in Table 16).

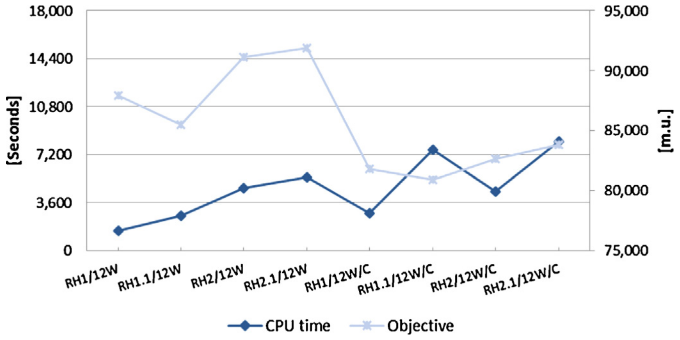


Fig. 13. Twelve weeks schedule: models computational times and objective function values (RH1 and RH2 have rolling horizon windows equal to 2 delivery periods; RH1.1 and RH2.1 have rolling horizon windows equal to 3 delivery periods).

7.2.3. 12 weeks scheduling

In the 12 weeks instance, we opted to just apply the RH approach (see Fig. 13 and Table 17), since in the 8 weeks scheduling horizon the DS models demonstrated to be computationally intractable. Without campaign tasks, the best solution found has a profit of 91,909.2 m.u. and was obtained by RH2.1 in 5491.0 CPU seconds. Assuming campaign tasks RH2.1 obtained as well the best solution with a profit of 83,801.40 m.u., which is by 9% inferior to the scenario that does not consider campaign tasks. The RH approaches demonstrated to be a good alternative when exact methods (as are the DS models presented in this paper) tend to obtain solutions with high integrality gaps. In practical terms, the CPU time required by the RH approaches to solve the 3 months scheduling problem has been considered acceptable by the company.

Generally, the definition of campaign tasks responds to one important requirement that we have found in the chemical–pharmaceutical industry: products with well-defined recipes are typically produced in the same processing units and follow predefined production sequences. Moreover, the number of binary and continuous variables and constraints decreased as a result of the task and resource aggregation done in the campaign tasks. Campaign tasks provide more responsive schedules by decreasing the lead time, but may have higher storage costs as a result of the storage policy for the stable intermediaries. The definition of these aggregate tasks allows as well a variation of the amounts being produced, limited by a minimum and maximum production lot, which is not possible to achieve if the typical periodic scheduling approach is applied.

8. Conclusions

This paper addresses the scheduling multipurpose of batch plants that simultaneously consider two different operating conditions – regular and non-regular production. The former encompasses the products that are manufactured regularly in predefined production lines and the latter includes under development products having no defined production lines.

A solution approach to solve such problem is proposed, which was developed along mathematical formulations based on RTN. The approach considers the integration of campaign and short-term scheduling in multipurpose batch plants, and proposes a three-step procedure that firstly determines the campaign schedule, secondly creates the campaign tasks and thirdly obtains a detailed schedule for the campaign and non-regular products. Campaigns are modeled as aggregate tasks that take into account the production resources determined previously, while the non-regular products are modeled using their detailed recipe. Campaign tasks proved to be an efficient concept in the cases where the definition of production lines requires cyclic operation mode, which is the procedure followed at the pilot company of this study. In the case study, the use of campaign tasks led to a reduction of the profit by 7%, 5% and 9% in the 4, 8 and 12 weeks schedules, respectively, when compared with the scenarios that do not consider campaign tasks. This profit reduction can be interpreted as the cost of the cyclic operation for the regular products.

To deal with the computational complexity of the larger instances, we have decided to compare the performances of a rolling horizon approach based on Dimitriadis et al. (1997) and Erdirik-Dogan and Grossmann (2007) with the reformulation and branching strategy proposed by Velez and Maravelias (2013). Moreover, we have performed several modifications in both methods in order to improve their performance. We propose a reformulation that considers new integer variables for the number of changeovers. Overall, the reformulation proposed by Velez and Maravelias (2013) together with the proposed reformulation improved the results of the base formulation. The combination of the two reformulations demonstrated better performance when compared with the reformulation of Velez and Maravelias (2013). Nevertheless, numerical results show that it is preferable to use the default CPLEX branching priority.

In the smaller instances, the DS models obtained the best solutions in very competitive time. Increasing the size of the scheduling problem, the DS models led to solutions with high integrality gaps (over than 30%) and required considerable CPU time, while the RH approaches obtained better solutions in very small CPU time. The performance of the RH approaches can be truly improved by adapting the variables fixing strategy and the length of the rolling horizon window to the problem. Additionally, it is important to note that the RH can naturally integrate the reformulation strategies for improving the performance of the algorithm.

For further study the authors aim to address other task-unit and temporal decomposition approaches inspired by current industrial practices. Moreover, improvements on the solutions obtained by the rolling horizon, while keeping this approach tractable for large instances, will be also explored.

Notation

Indices

l	campaign
d	delivery period
e	processing unit
k, k'	task
m	material
p	product
r	resource
t	time interval

Sets

AK	production tasks (without considering processing units)
A_K	alternative tasks to k
D_m^{DS}	delivery periods of product m of the detailed scheduling model
D^{AP}	delivery periods of the aggregate planning model
D_m^{AP}	delivery periods of product m of the aggregate planning model
E	processing units (equipments)
E^{fixed}	processing units that are fixed in the rolling horizon approach
H, H^{DS}	scheduling horizon
$H^{DSfixed}$	time horizon corresponding to the fixed tasks
I	intermediaries
I_l	intermediaries associated to campaign l
L	campaigns
M	materials (raw materials, intermediaries and final products)
K	tasks
K^{fixed}	tasks that are fixed in the rolling horizon approach
K_r	tasks that require resource r
K_m	tasks that consume or produce material m
K_e	tasks associated to unit e
K_l	tasks k associated to campaign l
K_e^{NR}	tasks of the non-regular products that require unit e
K^R	campaign tasks of the regular products
K_e^l	tasks k associated to campaign l and unit e
KP_e	tasks k of product p associated to unit e
P	products
P^{fixed}	products that are fixed in the rolling horizon approach
P_l	products associated to campaign l
P_e	products p that can be produced in unit e
R	production resources (processing units, intermediaries and final products)
R_l	production resources (processing units, intermediaries and final products) associated to campaign l

Parameters

τ_k	processing time of task k
τ_{ke}	processing time of task k in unit e (used in regular and non-regular products)
$\mu_{kr\theta}$	allocation/release coefficient of resource r in task k at time θ relative to the start of task
$\mu_{ke\theta}$	allocation/release coefficient of unit e in task k at time θ relative to the start of task
$\upsilon_{kr\theta}$	production/consumption of resource r in task k at time θ relative to the start of task
$\upsilon_{km\theta}$	production/consumption of material m in task k at time θ relative to the start of task
v_m	value of product m
$\alpha_{ep'p''p\theta}$	allocation/release changeover coefficient of unit e from product p' to product p'' being at product p and at time θ relative to the start of the changeover task
υ_{km}	production/consumption of material m in task k
a_m	non-delivery penalty factor for product m
$\Theta_r^{\min}, \Theta_r^{\max}$	minimum and maximum amounts for product r
C_m^{raw}	cost of materials for product m
C_m^{sto}	cost of storage of material m
chg^{AP}	changeover duration
c_e	changeover cost in unit e
$C_{ep'p''}$	changeover time between product p' and product p'' in unit e

it_d^{AP} length of delivery period d

$L_{rk}^{\min}$, $L_{rk}^{\max}$ minimum and maximum lot size of product r at campaign task k

NT_k^{fixed} number of times task k runs (used in the rolling horizon approach)

$Q_{md}^{DS\,\min}$, $Q_{md}^{DS\,\max}$, minimum and maximum amount of product m for delivery d

$R_{rt}^{\max}$ maximum resource availability of resource r at time interval t

$R_{mt}^{\max}$ maximum material m availability at time interval t

$R_{md}^{\max}$ maximum resource availability of material m at delivery d

R_m^{APinit} material m availability in the beginning of the planning horizon

R_r^{init} resource r availability in the beginning of the scheduling horizon

R_e^{init} unit e availability in the beginning of the scheduling horizon

R_m^{init} material m availability in the beginning of the scheduling horizon

T cycle time

T^{DS} length of the scheduling horizon

t_d time interval of the delivery period d

$V_{ke}^{\min}$, $V_{ke}^{\max}$ minimum and maximum capacity of unit e for task k

W raw materials

W_l raw materials associated to campaign l

Variables

Π_{mt}^{DS} continuous variables that define the delivery of product m at time interval t

$\Pi_{md}^{DSslack}$ continuous variables that define the slack of product m at delivery d

Π_{md}^{AP} continuous variable that define the amount of product m delivered at period d

$\Pi_{md}^{APslack}$ continuous variable that define the slack of product m at delivery d

Δ_{r} continuous variables that define the net production of resource r

$R_r^{init'}$ continuous variables that define the resource r availability in the beginning of the scheduling horizon

R_{rt} continuous variables that define the resource availability r at time interval t

R_{pet}^{DS} continuous variables that define the resource availability r of product p at time interval t

R_{pe}^{init} allocation of unit e at the beginning of the scheduling horizon

R_{mt}^{DS} continuous variables that define the material availability m at time interval t

R_{md}^{AP} continuous variables that define the availability of material m at delivery d

R_m^{APinit} continuous variables that define the material m availability in the beginning of the planning horizon (used in the rolling horizon)

ξ_{kt}, ξ_{kt}^{DS} continuous variables that define the batch size of task k at time interval t

ξ_{kd}^{AP} continuous variables that define the total amount of material processed by task k at delivery d

$C_{ep'p''t}^{DS}$ binary variables that define the changeover task in unit e between product p' and product p'' and at time interval t

N_{kt}, N_{kt}^{DS} binary variables that define if task k starts at time interval t

N_k^{DS} integer variables that define the number of times task k runs

NC_p^{DS} integer variables that define the number of changeovers associated to product p

N_{kd}^{AP} integer variables that define the number of occurrences of task k at delivery d

Y_{ped}^{AP} binary variables that define if product p is produced in unit e at delivery period d

λ_r binary variables that define the selection of the storable intermediaries

Acknowledgement

The authors gratefully acknowledge the financial support of Hovione FarmaCiencia SA, Fundação para a Ciência e Tecnologia, under the grant SFRH/BD/33970/2009, and the North Portugal Regional Operational Programme (ON.2-O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF).

References

Amaro AC, Barbosa-Póvoa APFD. Optimal supply chain management with detailed scheduling. Ind Eng Chem Res 2008a;47(1):116–32.

Amaro AC, Barbosa-Póvoa APFD. Planning and scheduling of industrial supply chains with reverse flows: a real pharmaceutical case-study. Comput Chem Eng 2008b;32:2606–35.

Barbosa-Povoa AP. A critical review on the design and retrofit of batch plants. Comput Chem Eng 2007;31:833–55.

Bassett M, Dave P, Doyle F, Kudva G, Pekny J, Reklaitis G, et al. Perspectives on model based integration of process operations. Comput Chem Eng 1996a;20:821–44.

Bassett MH, Pekny JF, Reklaitis GV. Decomposition techniques for the solution of large-scale scheduling problems. AIChE J 1996b;42:3373–87.

Castro P, Barbosa-Povoa A, Matos H. An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. Ind Eng Chem Res 2001;40:2059–68.

Castro P, Barbosa-Povoa AP, Matos HA. Optimal periodic scheduling of batch plants using RTN-based discrete and continuous-time formulations: a case study approach. Ind Eng Chem Res 2003;42:3346–60.

Castro P, Novais Q, Carvalho A. Optimal equipment allocation for high plant flexibility: an industrial case study. Ind Eng Chem Res 2008;47:2742–61.

Dimitriadis A, Shah N, Pantelides C. RTN-based rolling horizon algorithms for medium term scheduling of multipurpose plants. Comput Chem Eng 1997;21:S1061–6.

Erdirik-Dogan M, Grossmann IE. Planning models for parallel batch reactors with sequence-dependent changeovers. AIChE J 2007;53:2284–300.

Fumero Y, Corsano G, Montagna JM. Scheduling of multistage multiproduct batch plants operating in a campaign-mode. Ind Eng Chem Res 2012;51:3988–4001.

Ierapetritou M, Floudas C. Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. Ind Eng Chem Res 1998;37:4341–59.

Janak SL, Lin X, Floudas CA. Enhanced continuous-time unit-specific event-based formulation for short-term scheduling of multipurpose batch processes: resource constraints and mixed storage policies. Ind Eng Chem Res 2004;43:2516–33.

Kondili E, Pantelides C, Sargent R. A general algorithm for short-term scheduling of batch operations-I. MILP formulation. Comput Chem Eng 1993;17:211–27.

Li X, Gao L, Zhang C, Shao X. A review on integrated process planning and scheduling. Int J Manuf Res 2010;5:161–80.

Li Z, Ierapetritou M. Process scheduling under uncertainty: review and challenges. Comput Chem Eng 2008;32:715–27.

Lin X, Floudas CA, Modi S, Juhasz NM. Continuous-time optimization approach for medium-range production scheduling of a multiproduct batch plant. Ind Eng Chem Res 2002;41:3884–906.

Maravelias CT, Grossmann IE. New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants. Ind Eng Chem Res 2003;42:3056–74.

Maravelias CT, Sung C. Integration of production planning and scheduling: overview, challenges and opportunities. Comput Chem Eng 2009;33:1919–30.

Mauderli A, Rippin D. Production planning and scheduling for multi-purpose batch chemical plants. Comput Chem Eng 1979;3:199–206.

Mendez CA, Cerda J, Grossmann IE, Harjunkoski I, Fahl M. State-of-the-art review of optimization methods for short-term scheduling of batch processes. Comput Chem Eng 2006;30:913–46.

Méndez C, Henning G, Cerda J. An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities. Comput Chem Eng 2001;25:701–11.

Moniz S, Barbosa-Póvoa AP, Pinho de Sousa J. Regular and non-regular production scheduling of multipurpose batch plants. In: Proceedings of the 22nd european symposium on computer aided process engineering; 2012.

Pantelides CC. Unified frameworks for optimal process planning and scheduling. New York: Cache Publications; 1994. p. 253–74.

Papageorgiou LG, Pantelides CC. A hierarchical approach for campaign planning of multipurpose batch plants. Comput Chem Eng 1993;17:S27–32.

Papageorgiou LG, Pantelides CC. Optimal campaign planning/scheduling of multipurpose batch/semicontinuous plants. 1. Mathematical formulation. Ind Eng Chem Res 1996a;35:488–509.

Papageorgiou LG, Pantelides CC. Optimal campaign planning/scheduling of multipurpose batch/semicontinuous plants. 2. A mathematical decomposition approach. Ind Eng Chem Res 1996b;35:510–29.

Pinto T, Barbosa-Póvoa APFD, Novais AQ. Optimal design and retrofit of batch plants with a periodic mode of operation. Comput Chem Eng 2005;29:1293–303.

Pochet Y, Warichet F. A tighter continuous time formulation for the cyclic scheduling of a mixed plant. Comput Chem Eng 2008;32:2723–44.

Reklaitis G. Scheduling approaches for the batch process industries. ISA Trans 1995;34:349–58.

Schilling G, Pantelides C. A simple continuous-time process scheduling formulation and a novel solution algorithm. Comput Chem Eng 1996;20:S1221–6.

Schilling G, Pantelides C. Optimal periodic scheduling of multipurpose plants. Comput Chem Eng 1999;23:635–55.

Shah N, Pantelides C, Sargent R. Optimal periodic scheduling of multipurpose batch plants. Ann Oper Res 1993;42:193–228.

Shah N. Pharmaceutical supply chains: key issues and strategies for optimisation. Comput Chem Eng 2004;28:929–41.

Stefansson H, Sigmarsdottir S, Jensson P, Shah N. Discrete and continuous time representations and mathematical models for large production scheduling problems: a case study from the pharmaceutical industry. Eur J Oper Res 2011;215(2):383–92, http://dx.doi.org/10.1016/j.ejor.2011.06.021.

Sundaramoorthy A, Maravelias CT. Computational study of network-based mixed-integer programming approaches for chemical production scheduling. Ind Eng Chem Res 2011;50:5023–40, http://dx.doi.org/10.1021/ie101419z.

Sundaramoorthy A, Karimi I. Planning in pharmaceutical supply chains with outsourcing and new product introductions. Ind Eng Chem Res 2004;43:8293–306.

Velez S, Maravelias CT. Reformulations and branching methods for mixed-integer programming chemical production scheduling models. Ind Eng Chem Res 2013;52:3832–41.

Verderame PM, Elia JA, Li J, Floudas CA. Planning and scheduling under uncertainty: a review across multiple sectors. Ind Eng Chem Res 2010;49:3993–4017.

Vooradi R, Shaik MA. Improved three-index unit-specific event-based model for short-term scheduling of batch plants. Comput Chem Eng 2012;43:148–72, http://dx.doi.org/10.1016/j.compchemeng.2012.03.014.

Wu D, Ierapetritou M. Cyclic short-term scheduling of multiproduct batch plants using continuous-time representation. Comput Chem Eng 2004;28:2271–86.

Wu D, Ierapetritou M. Hierarchical approach for production planning and scheduling under uncertainty. Chem Eng Process Process Intens 2007;46:1129–40.

Wu D, Ierapetritou MG. Decomposition approaches for the efficient solution of short-term scheduling problems. Comput Chem Eng 2003;27:1261–76.

You F, Castro PM, Grossmann IE. Dinkelbach's algorithm as an efficient method to solve a class of MINLP models for large-scale cyclic scheduling problems. Comput Chem Eng 2009;33:1879–89.

Zentner M, Pekny J, Reklaitis G, Gupta J. Practical considerations in using model-based optimization for the scheduling and planning of batch/semicontinuous processes. J Process Control 1994;4:259–80.