

# Optimal Scheduling of Continuous Plants with Energy Constraints

*Pedro M. Castro,<sup>\*,†,◇</sup> Iiro Harjunkoski<sup>‡</sup> and Ignacio E. Grossmann<sup>◇</sup>*

<sup>†</sup> Unidade de Modelação e Optimização de Sistemas Energéticos, Laboratório Nacional de Energia e Geologia, 1649-038 Lisboa, Portugal

<sup>‡</sup> ABB Corporate Research Center, Wallstadter Str. 59, 68526 Ladenburg, Germany

<sup>◇</sup> Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, United States

## Abstract

This work addresses the scheduling of continuous single stage multiproduct plants with parallel units and shared storage tanks. Processing tasks are energy intensive and we consider time-dependent electricity pricing and availability together with multiple intermediate due dates, handled as hard constraints. A new discrete-time aggregate formulation is proposed to rapidly plan the production levels. It is combined with a continuous-time model for detailed scheduling as the essential part of a rolling horizon algorithm. Their computational performance is compared to traditional discrete and continuous-time full-space formulations with all models relying on the Resource-Task Network (RTN) process representation. The results show that new models and algorithm can generate global optimal schedules much more efficiently than their counterparts in problems involving unlimited power availability. Under restricted power, the aggregate model

---

\* To whom correspondence should be addressed. Tel.: +351-210924643. Fax: +351-217167016. E-mail: [pedro.castro@ineti.pt](mailto:pedro.castro@ineti.pt).

underestimates the electricity cost, which may cause the rolling-horizon approach to converge to a suboptimal solution, becoming the discrete-time model a better approach.

## **1. Introduction**

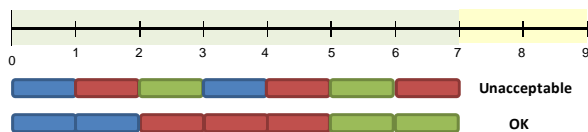
Modern enterprises form complex global networks of multiple business units and functions. In order to remain competitive in the global marketplace, companies need to optimize the various functions within their supply chain. One of the focuses of enterprise-wide optimization (EWO) is the operation of manufacturing facilities, where essential operational items include planning and scheduling (Grossmann, 2005). A major challenge is to discover what type of models to use to render effective solutions for real-life problems.

Scheduling has been receiving considerable attention by the Process Systems Engineering (PSE) community, particularly for batch plants. The review paper by Méndez et al. (2006) provides a classification and conceptual overview of the most important approaches. In the last 15 years researchers have focused on continuous-time formulations. With the advent of unified modeling for batch and continuous processes (STN, Kondili et al., 1993, and RTN, Pantelides, 1994), work concerning model generality is mostly complete. Relevant recent contributions have extended the scope by considering: (i) variable recipe tasks with flexible rather than fixed proportion of input materials (Castro et al., 2009b); (ii) non zero transfer times to enforce synchronization between consecutive processing stages (Ferrer-Nadal et al., 2008; Giménez et al., 2009); (iii) tasks to continue over multiple events when using unit-specific formulations in problems with resource constraints other than equipment (Shaik & Floudas, 2009); (iv) time dependent utility pricing and availability when dealing with continuous plants (Castro et al., 2009a). However, there is still much work to be done concerning model efficiency and to find out the best approach for a particular type of problem.

Our recently proposed continuous-time short-term scheduling formulation for continuous plants (Castro et al., 2009a) effectively handles energy constraints related to electricity pricing and availability. It is a Resource-Task Network (Pantelides, 1994) formulation evolving from the one by Castro et al. (2004), and takes advantage of the ability of continuous tasks to be split in as many time

slots as required so as to adapt to the duration of the time intervals where they take place. Such duration may result from a pre-specified utility cost/availability profile with binary variables being used to identify whether a certain interval belongs to a particular period of a given energy level. The other set of discrete events whose exact location is known a priori are the multiple intermediate due dates with binary variables identifying if a certain event point on the continuous-time grid corresponds to a given demand point as suggested by Maravelias & Grossmann (2003). As the number of different energy levels and demand points increase, so does the number of time intervals needed to represent a solution, which is known to have a profound effect on the complexity of the mathematical formulation. Thus, it was no surprise to observe that this approach could only solve very small problems to optimality when considering a 1-week horizon with end-of-the-day demands and frequent energy cost changes (6 times on week days and 4 times on Saturdays, from Duarte et al., 2009).

Discrete events are handled more naturally with a discrete-time formulation (Kondili et al., 1993; Pantelides, 1994). Indeed, the results in Castro et al. (2009a) have shown superior performance over its continuous-time counterpart, with the RTN discrete-time formulation being able to tackle problems of industrial relevance using 1-hour time intervals to near optimality (<1%) in a few minutes of computational time. There are however two important issues when addressing problems featuring continuous tasks with discrete-time formulations. First, slightly suboptimal solutions may result since the task durations are approximated by a multiple of the prespecified interval length. Second, multiple instances will typically need to be executed to meet the daily demands, where each instance lasts a single interval, leading to high solution degeneracy, see Figure 1. Thus, one will only by chance obtain a solution with a minimum number of changeovers that can immediately be implemented in practice.

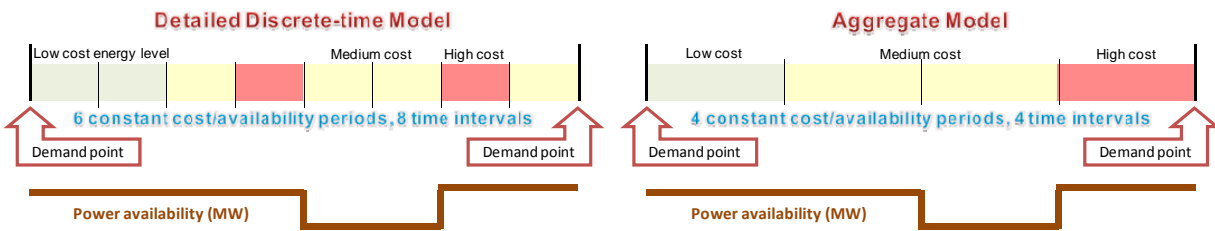


**Figure 1.** High solution degeneracy may lead to a solution from the discrete-time formulation with an unacceptable number of changeovers.

Clearly, discrete and continuous-time formulations have complementary strengths and the ideal approach should combine the advantages of both. Researchers have started looking into this problem by proposing mixed-time approaches (Maravelias, 2005; Westerlund et al., 2007) where the main idea is to allow product changeovers to occur between interval boundaries instead of only at the exact boundaries of a time grid. Yet, the authors acknowledge that improved solutions methods are required for real-world applications. Rather than going for a full-space formulation, a new rolling-horizon algorithm is proposed in this paper. We follow the basic ideas of the original paper (Dimitriadis et al., 1997), where the scheduling horizon is divided into detailed and aggregate time blocks with the overall problem being solved in a sequence of iterations. In successive iterations, the previous solution of the detailed time block is fixed and part of the remaining aggregate time block gets to be scheduled in detail.

In rolling-horizon algorithms, the most critical issue is finding a simplified (aggregate) model resulting into decisions that can actually be implemented in practice. In this paper, a novel aggregate model is proposed for single stage plants. In many ways, it is similar to a detailed discrete-time model but it uses fewer time intervals in order to reduce solution degeneracy. The reduction in the number of time intervals is possible since the relative position of the different cost/availability levels between two consecutive demand points is not important whenever instantaneous product withdrawals from the final storage units (at the demand points) can be assumed. In such cases, the inventory will steadily increase up to a maximum, occurring at the latest immediately before the later demand point. An illustration is given in Figure 2, where the aggregate model considers half of the time intervals of the discrete-time approach due to the merging of time periods with constant power availability belonging to the same energy cost level. The difference is that instead of applying the resource balances at the interval boundaries, we now do balances over the time intervals for processing equipment and electric power. For the former resources the new balances are rigorous since we ensure that the total processing time does not exceed the duration of the time interval. Note that we are no longer concerned with the actual timing of events so as to focus solely on the most

relevant information (planning level). The instantaneous power balances are approximated by energy balances over a given time, which give an average power consumption.

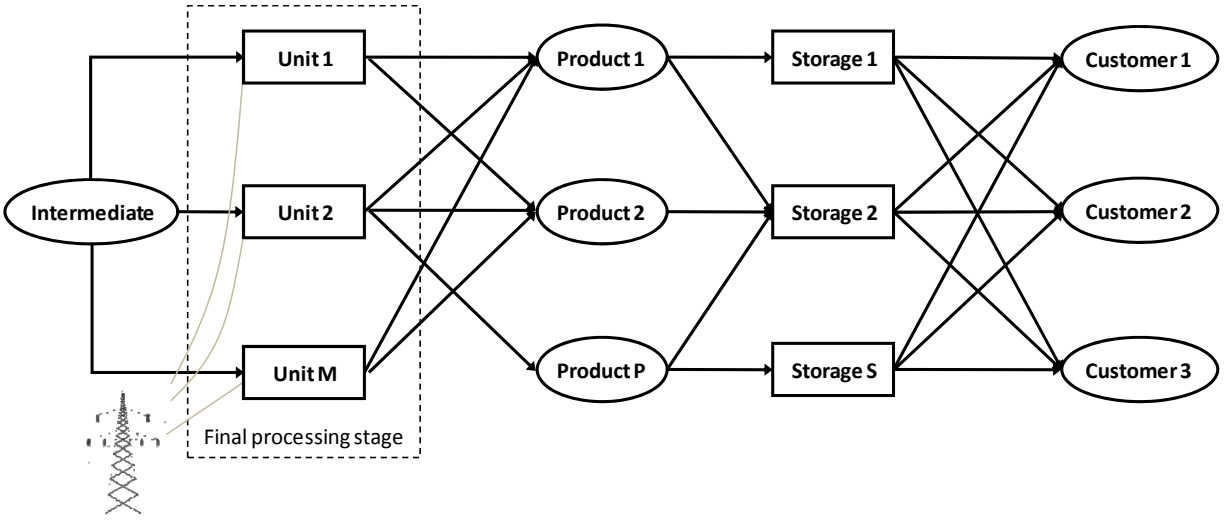


**Figure 2.** Detailed discrete-time vs. aggregate model

The underlying detailed model in the new rolling horizon algorithm is the continuous-time formulation presented in Castro et al. (2009a). It is applied to the period between two consecutive demand points (e.g. one day) rather than the full scheduling horizon (one week, typically) so as to keep complexity at a manageable level. On the one hand, within a demand period, binary variables linking event points with demand points are no longer needed. On the other hand, significantly fewer event points are required to represent the partial solution. The actual number to employ is predicted from the solution of the aggregate model, which will act as the starting point in the standard iterative search procedure for the global optimal solution (Méndez et al., 2006). The drawback of decomposition techniques over full-space methods is that optimality, sometimes even feasibility, may be compromised. In this paper, this is caused by the approximate power balances on the aggregate model, which may suggest production in periods of cheap electricity cost where there simply is not enough power available (as can be found by the rigorous detailed model). To overcome this issue, slack variables are added to allow exceeding the maximum power availability so that the product demands are met. These are severely penalized in the objective function to first generate feasible schedules.

## 2. Problem Definition

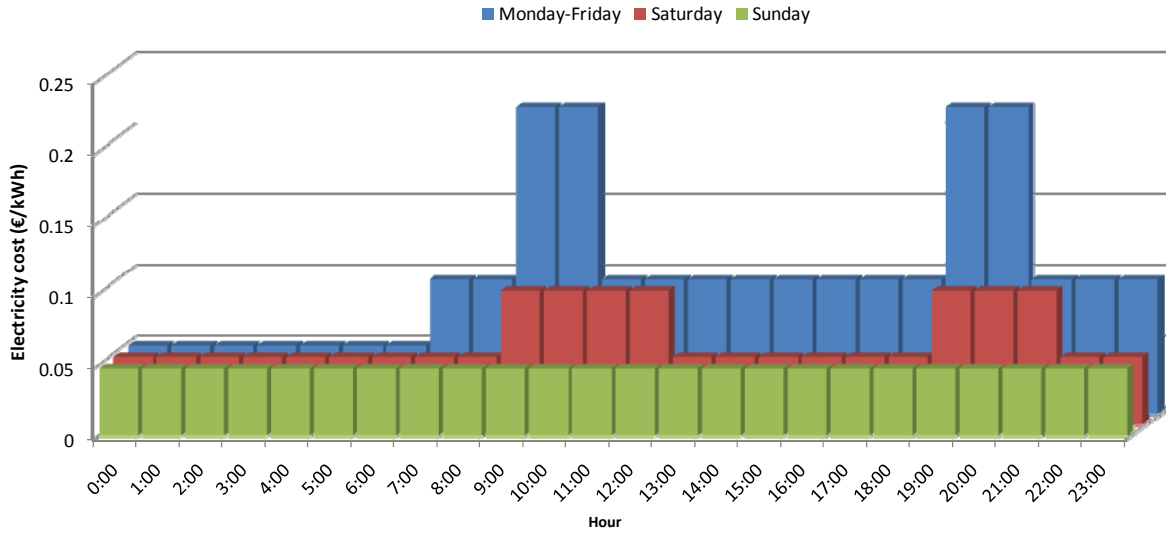
Consider the last processing stage of a continuous multiproduct plant, where an intermediate material is transformed through the use of electricity into one of a few different products. The products are characterized by chemical composition and sent to storage units until dispatch takes place. This process is illustrated in Figure 3.



**Figure 3.** Final processing stage of industrial case study

Typical plant schedules are established over one week, so this will be the time horizon assumed ( $H=168$  h). Given are a set of units  $M$ , products  $P$  and storage tanks  $S$ . Machines are characterized by: (i) power requirements for each product  $pw_{p,m}$  [MW]; (ii) processing rates,  $\rho_{p,m}$  [ton/h]. Products may have multiple demands over the week, which can occur at any hour  $hr$  of day  $dy$ ,  $d_{p,dy,hr}$  [ton]. Storage units have known maximum capacities,  $cap_s$  [ton], and the initial amounts in storage are given by  $im_{s,p}$ . We assume that all products can be stored on a given unit (shared storage) provided that is one at a time.

The objective will be to minimize the total energy cost subject to constraints on resource availability that includes processing and storage units, materials and utilities (electricity). We assume an energy contract between the plant and electricity provider that specifies the time-dependent electricity cost,  $ec_{hr,dy}$  [€/kWh], and maximum power levels,  $pw_{x,hr,dy}$  [MW], over a one-week period. Throughout the paper we will be using the cost values given in Duarte et al. (2009) that consist of three distinct energy pricing levels,  $E$ , with prices  $c_e$  equal to 0.0481, 0.0945 and 0.2162 [€/kWh], see Figure 4. The maximum power levels are treated as soft constraints, i.e. they can be violated at a harsh penalty cost,  $cs$  [k€/MW], so that the product demands are always met (hard constraint). This is a reasonable assumption since the plant normally operates well below maximum capacity.

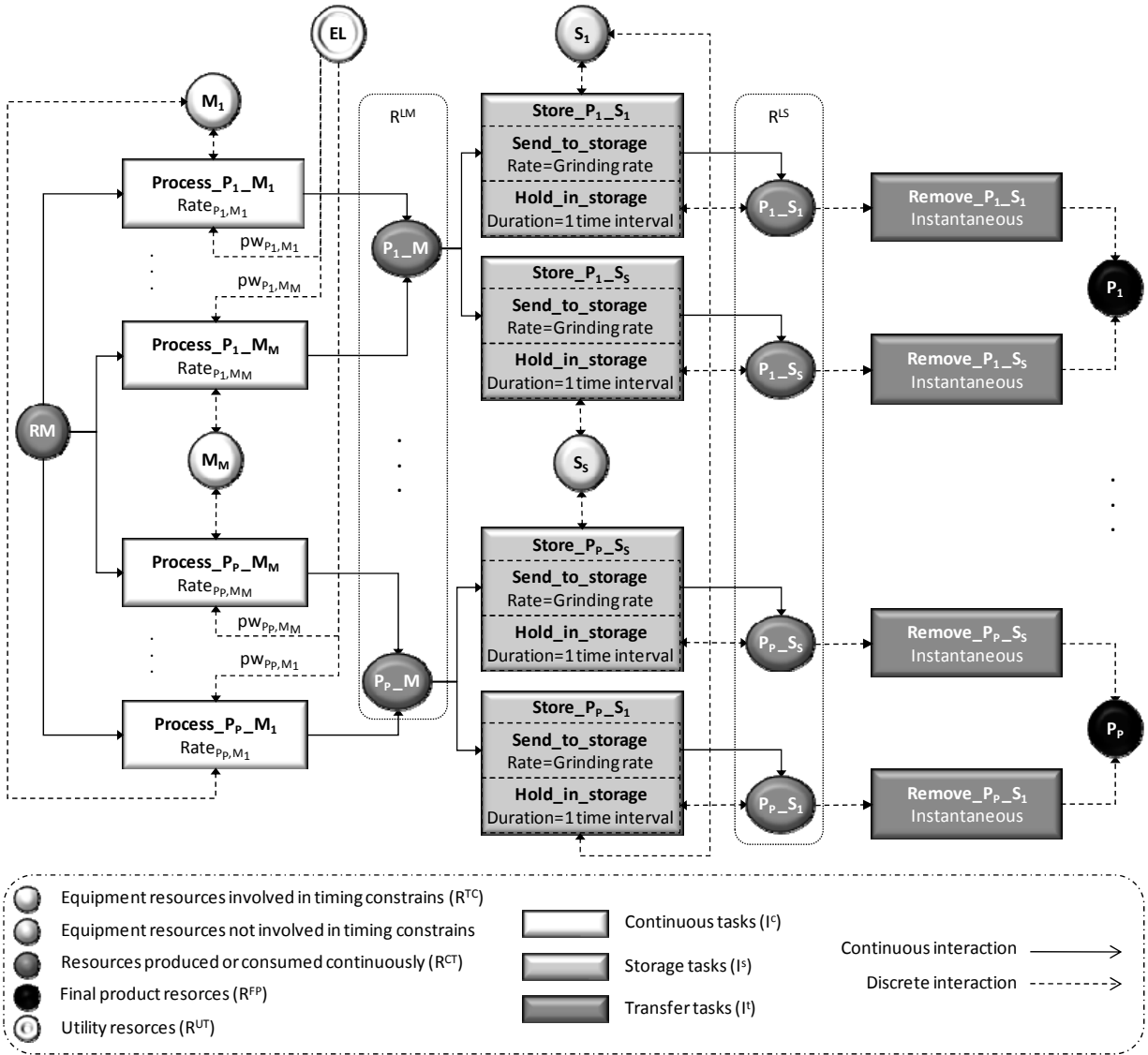


**Figure 4.** Electricity cost policy within a working week.

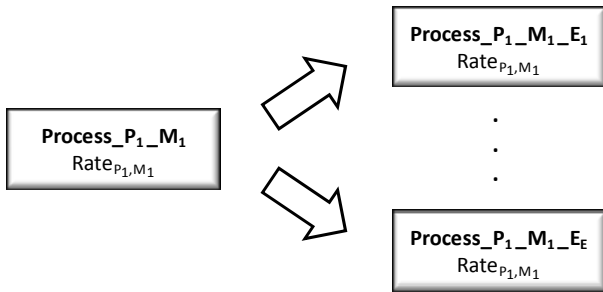
### 3. RTN Process Representation

The proposed models assume a Resource-Task Network representation of the process (Pantelides, 1994) to make them as general as possible. In order to tackle the scheduling problem at hand, the first step involves describing the process as a network of resources and tasks. There are several possibilities for deriving the network, so some creativity may be involved. Some will naturally be better than others as noted by Chen & Chang (2009) when addressing a heat-integrated multipurpose batch plant. For the problem under consideration, the proposed network can be seen in Figure 5. The reader is referred to Castro et al. (2009) for further details. Here, we limit ourselves to highlight the most important features.

While the concept of resource is entirely general, it is convenient to define a few subtypes that will be treated differently by the mathematical formulation. Equipment resources,  $R^{EQ}$ , include processing and storage units, with the former appearing in the timing constraints,  $R^{TC}$ . The electricity is the utility,  $R^{UT}$ , while the raw-material is the sole member of set  $R^{RM}$ . The material output from the processing task is already in final product condition but we need to distinguish its location (Castro et al., 2005). It can either be: (i) immediately after the units,  $R^{LM}$ ; (ii) inside the storage tanks,  $R^{LS}$ ; (iii) inside the client's transportation vehicle,  $R^{FP}$ . The resources continuously produced/consumed are given by  $R^{CT}=R^{RM}\cup R^{LM}\cup R^{LS}$ , whereas the final product ( $R^{FP}$ ) is produced/consumed instantaneously.



**Figure 5.** Resource-Task Network representation.



**Figure 6.** Disaggregation of continuous tasks for combined aggregate/continuous-time model.

In Figure 5 there are three types of tasks: (a) continuous processing tasks,  $I^C$ ; (b) hybrid storage tasks consisting of a continuous send-to-storage part and a batch hold-in-storage part,  $I^S$ ; (c) instantaneous remove-from-storage tasks,  $I^I$ . The processing tasks need to be further disaggregated when solving the problem with the combined aggregate/continuous-time formulation, to distinguish



between energy pricing levels,  $I_e^c$  (see Castro et al., 2009a). Note also that hybrid storage tasks allow considering shared storage, where a particular unit can handle multiple products, one at a time.

#### 4. RTN Variables and Parameters

The large majority of variables and parameters are independent of the time representation employed by the mathematical formulation. A brief description of such entities is the focus of this section.

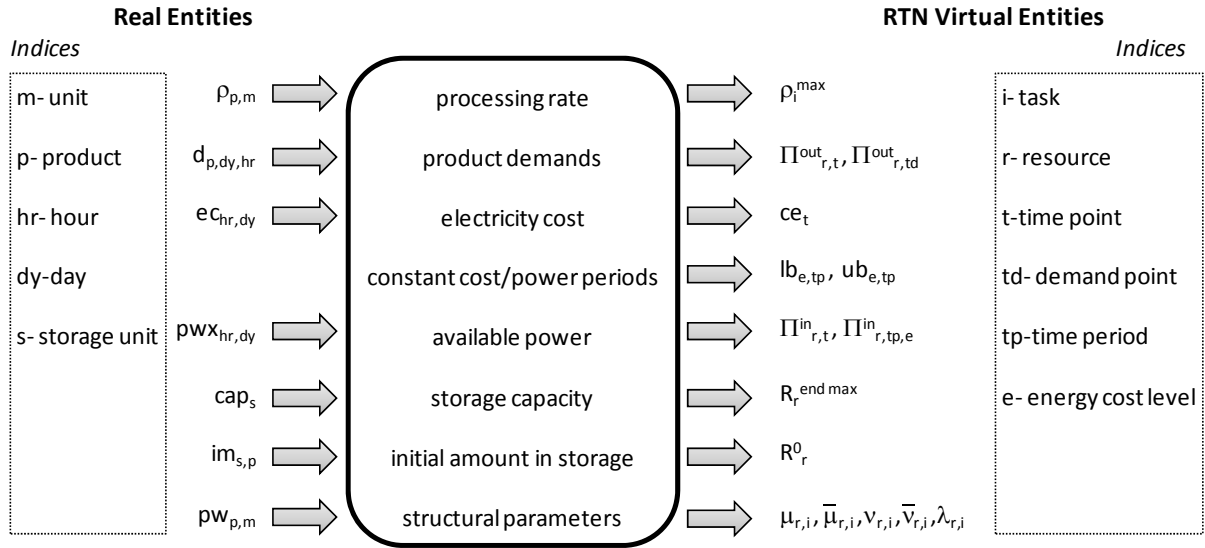
Normally, two sets of variables are used to characterize tasks. Binary variables  $N_{i,t}$  identify the execution of task  $i$  during interval  $t$ , while continuous variables  $\zeta_{i,t}$  give the amount processed. In the case of hybrid tasks, an extra set of continuous variables is required to determine the amount continuously sent to storage,  $\xi_{i,t}^*$ .

Resource availability over time is accounted for through excess resource variables  $R_{r,t}$  and  $R_{r,t}^{end}$ . The amount available at the beginning of the time horizon is often known for all resources and so it can be defined as a parameter,  $R_r^0$ . As an example, it is equal to 1 for equipment resources ( $R^{EQ}$ ). The upper and lower bounds on resource availability are specified through the parameters  $R_r^{\max}$ ,  $R_r^{end\max}$ ,  $R_r^{\min}$  and  $R_r^{end\min}$ . For resource interaction with the system surrounding environment, the aggregate model uses parameters  $\Pi_{r,t}^{in}$  and  $\Pi_{r,t}^{out}$ , with the indices changing for the continuous-time model ( $\Pi_{r,tp,e}^{in}, \Pi_{r,td}^{out}$ ). Slack variables  $S_{r,t}$  are included to allow the balances for the utility resources ( $R^{UT}$ ) to be violated.

The RTN representation is brought into the model by the structural parameters, which give the total resource consumption/production, or the proportion relative to the amount handled by the task. Furthermore, task interaction with resources can either occur discretely at the start/end of the task, or continuously throughout its execution. The parameters  $\mu_{r,i}$  and  $\bar{\mu}_{r,i}$  handle discrete interactions at the start (end) of tasks involving a known total consumption and are linked to the binary variables  $N_{i,t}$ . They are typically used for equipment resources. On the other hand, for material resources it is preferable to relate resource consumption to the amount handled by the task, through parameters  $\nu_{r,i}$

and  $\bar{v}_{r,i}$ . Continuous interactions are modeled using  $\lambda_{r,i}$ , which are associated to either  $\xi_{i,t}$  or  $\xi_{i,t}^*$ . It is important to highlight that there can be a large number of structural parameters but most will be equal to 0, and the remaining will almost always have a value of either 1 or -1.

Ideally, the parameters should be generated systematically so that the model can handle changes in the number of real process entities (e.g. units, products). For special network structures, like those in single or multistage plants, it is straightforward to derive an algorithm that converts the real process entities into the virtual entities of the RTN. It can be viewed as a black box, requiring process data inputs, as well as information linked to the model type in terms of time representation. The procedure is summarized in Figure 7, with the reader being directed for Castro et al. (2009a) for further details.

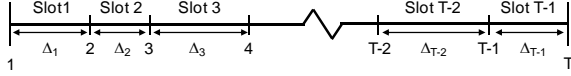


**Figure 7.** RTN parameter generation algorithm

## 5. New Aggregate Formulation (AG)

The proposed aggregate formulation uses a single, non-uniform, discrete-time grid to keep track of the events taking place, see Figure 8. The number of time points ( $|T|$ ) and their exact location on the grid is determined after analyzing the product demand, and the electricity cost and availability data, as illustrated in Figure 2. The difference with respect to a traditional discrete-time formulation (given in Castro et al., 2009a), is that the duration is no longer the same for all intervals. Instead, the width of interval  $t$  is given by  $\Delta_t$ . Furthermore, since we are not concerned with the rigorous timing

of events in the aggregate model, time varies continuously within an interval without the need to specify event points, like in a typical continuous-time formulation.



**Figure 8.** Non-uniform, discrete time grid for aggregate model

The model constraints follow. The objective function [k€], eq. 1, minimizes the total electricity cost plus a penalty to account for power violations. In the first term,  $ce_t$  is the electricity cost in time interval  $t$  [€/kWh],  $-\mu_{r,i}$  gives the power consumption [MW], and the quotient returns the duration of the electricity intensive task [h]. In the second term, the slack variables  $S_{r,t}$  represent the extra power to purchase [MW].

$$\min \sum_{i \in I^c} \sum_{r \in R^{UT}} \sum_{t \in T \wedge t \neq |T|} ce_t \cdot (-\mu_{r,i}) \cdot \frac{\xi_{i,t}}{\rho_i^{\max}} + cs \sum_{r \in R^{CT}} \sum_{t \in T} S_{r,t} \quad (1)$$

The excess resource balances are given by eqs. 2-3. For resources interacting discretely with tasks, the excess amount at event point  $t$  ( $R_{r,t}$ ) is equal to that at the previous event point ( $t-1$ ) adjusted by the amounts produced/consumed by all tasks starting or ending at  $t$ . For those interacting continuously ( $R^{CT}$ ), it is the excess amount immediately before the end of the previous interval ( $R_{r,t-1}^{end}$ ) that matters in eq. 2. This is in turn related to the amount available at the start of the interval through eq 3. A more thorough explanation of the two sets of constraints for the continuous-time formulation, which are very similar to eqs. 2-3, can be found in Castro et al. (2009a).

$$R_{r,t} = R_r^0 \Big|_{t=1} + R_{r,t-1}^{end} \Big|_{r \in R^{CT}} + R_{r,t-1} \Big|_{r \notin R^{CT}} - \Pi_{r,t}^{out} \Big|_{r \in R^{FP}} + \sum_{i \in I} (\mu_{r,i} N_{i,t} \Big|_{t \neq |T|} + \nu_{r,i} \xi_{i,t} + \bar{\mu}_{r,i} N_{i,t-1}) + \sum_{i \in I^c} \bar{\nu}_{r,i} \xi_{i,t} \quad \forall r \notin (R^{TC} \cup R^{UT}), t \in T \quad (2)$$

$$R_{r,t}^{end} = R_{r,t} + \sum_{i \in I^c} \lambda_{r,i} \xi_{i,t} + \sum_{i \in I^s} (\bar{\nu}_{r,i} \xi_{i,t} + \lambda_{r,i} \xi_{i,t}^*) \quad \forall r \in R^{CT}, t \in T, t \neq |T| \quad (3)$$

The novelty in the aggregate model is that the domain of eq. 2 does not include processing equipment or utilities. The purpose is to allow multiple products to be processed on a given unit on a certain interval. Since for every product, one unit of the corresponding equipment resource is consumed and their maximum (initial) availability is equal to 1, we are forced to remove processing equipment from the resource balances to meet our goal. These are replaced by timing constraints that

ensure that the total processing time in interval  $t$  does not exceed its duration, as given by eq. 4. For the utilities, the instantaneous balances on power are only possible through the resource balances, so the best that can be done is to write energy balances. Eq. 5 ensures that the total energy in interval  $t$  (LHS) is greater than the amount consumed (RHS), with the equality holding whenever the slack variables are active (recall that they are penalized in the objective function, eq. 1).

$$\Delta_t \geq \sum_{i \in I^c} \frac{\bar{\mu}_{r,t} \xi_{i,t}}{\rho_i^{\max}} \quad \forall r \in R^{TC}, t \in T, t \neq |T| \quad (4)$$

$$\Delta_t \cdot (\Pi_{r,t}^{in} + S_{r,t}) \geq \sum_{i \in I^c} \frac{-\mu_{r,i} \xi_{i,t}}{\rho_i^{\max}} \quad \forall r \in R^{UT}, t \in T, t \neq |T| \quad (5)$$

Finally, the last set of constraints forces the activation of the binary extent variables  $N_{i,t}$  whenever there is material being handled by the task. We have used the product of the time horizon,  $H$ , times the sum over all machines of the maximum processing rates, as the upper bound in eq. 6.

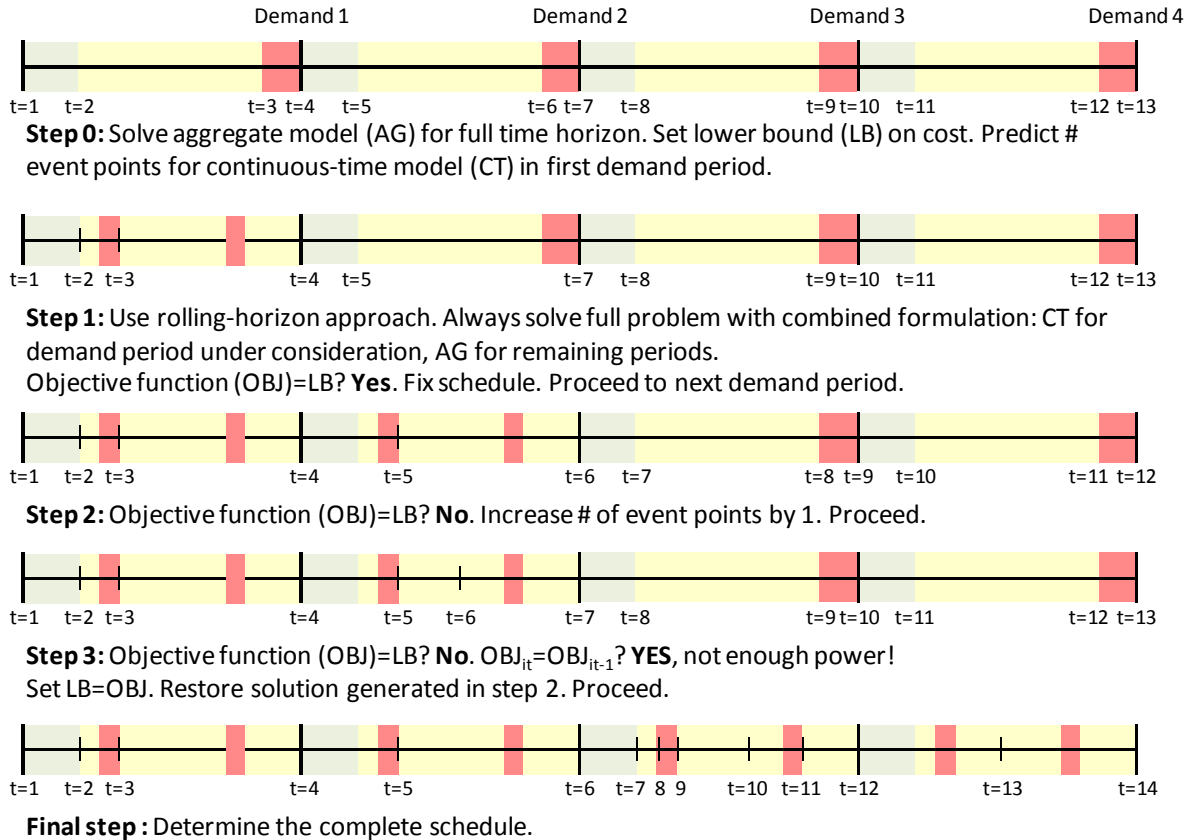
$$\xi_{i,t} + \xi_{i,t}^* \Big|_{i \in I^s} \leq H \cdot \sum_{m \in M} \max_{p \in P} \rho_{p,m} \cdot N_{i,t} \quad \forall i \in I^c \cup I^s, t \in T, t \neq |T| \quad (6)$$

## 6. Rolling-Horizon Algorithm

The aggregate model is one of the two components of the combined formulation that is used in the rolling-horizon algorithm. The most important property of the aggregate model is that it solves a relaxed version of the problem by replacing instantaneous power balances with energy balances over the time slots that comprise the grid (see also section 8.1). Thus, it generally provides a lower bound on the total electricity cost. However, in cases with unlimited power availability, it provides the exact minimum cost that can be achieved after detailed scheduling. The other advantage of the aggregate model is that its solution gives the planned production levels for the different demand periods, information that can be analyzed to predict the number of event points for the continuous-time formulation. Hence, it is logical that the initial step of the rolling-horizon algorithm involves solving the full problem with the aggregate model.

The main ideas of the rolling-horizon algorithm are illustrated in Figure 9. The combined formulation described in the next section uses a single time grid that is partly continuous and partly discrete,  $T = T^c \cup T^{ag}$ . The first time points will be continuous, with the last being the sole element of

set  $T^{c\ last}$ . It is a boundary time point that is also part of the set of aggregate time points, thus binding together the two parts of the grid. Both step 0 and 1 involve 13 event points, but the sets elements differ. In the former,  $T^c=\emptyset$  and  $T^{ag}=\{1,\dots,13\}$ , while in the latter,  $T^c=\{1,\dots,4\}$  and  $T^{ag}=\{4,\dots,13\}$ . In the illustration, the outcome of step 1 is a solution with the same value of the objective function as step 0. Since we cannot do better, we can fix the schedule for the first demand period and proceed to the next. By fixing we mean removing the time points belonging to the first demand period from the grid, with the exception of the one at the boundary. The aggregate model predicts 3 event points for the second demand period, so in step 2 we will have  $T^c=\{4,\dots,6\}$  and  $T^{ag}=\{6,\dots,12\}$ . Notice that the set elements are changing dynamically. Other important dynamic sets that need to be defined are the first and last points on the grid, e.g.  $T^{first}=\{4\}$  and  $T^{last}=\{12\}$ .



**Figure 9.** Main ideas implemented in rolling-horizon algorithm

The continuous-time model considers the exact location of the different electricity cost levels, leading to the disaggregation of the time periods from the aggregate model. As a consequence, processing tasks that could be executed in a single time interval may need two or more intervals in

the continuous grid. In other words, the aggregate model may predict an insufficient number of event points. This is one of the two possible causes for the value of the objective function to be higher than the lower bound. If that is the case, then the value of the objective function will improve following a single increment in  $|T|$ . If no improvement is observed (step 3), then the cause is insufficient power to process an amount at least as much as that predicted by the aggregate model in the cheaper cost periods. The lower bound is reset to the new value, the added continuous-time point removed (note that the event point corresponding to  $t=6$  in step 3 is not in the final solution) and the algorithm proceeds. Eventually, all demand periods are solved and the full detailed schedule is generated in the final step.

### 7. New Combined Aggregate/Continuous-time Formulation (RH)

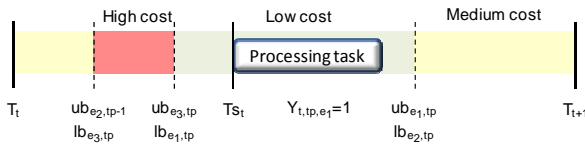
The new combined aggregate/continuous-time model, the most important part of the rolling-horizon algorithm, is presented next. Despite the natural complexity resulting from mixing models with alternative representations of time, the use of a unified framework for process representation (RTN) and dynamic sets, allows the generation of a relatively compact formulation. Since the aggregate part of the model has already been thoroughly discussed, we focus on the continuous-time part of the model.

The objective function is given in eq. 7. The first term,  $pcost$ , is a parameter that accounts for the partial electricity cost in demand periods already scheduled. Inside the square brackets, the first term is the electricity cost associated to processing tasks executed in the continuous part of the grid. It is given by the sum over all electricity levels  $e$ , of the product of electricity cost  $c_e$ , power consumption and duration of the task. Notice that contrary to the aggregate model, we do not know a priori the cost level associated to time interval  $t$ . Instead, we need a set of timing constraints to ensure that the correct parameter is activated, see eq. 16. Due to the disaggregation of tasks amongst energy levels (recall Figure 6), the number of continuous tasks to consider is larger than for the aggregate model. Nevertheless, not all such tasks need to be considered in time interval  $t$ , in the aggregate part of the combined model (only the elements of set  $I_t^c$ ).

$$\min pcost + \sum_{r \in R^{UT}} [\sum_{e \in E} \sum_{t \in T^c} \sum_{i \in I_i^c} c_e \cdot (-\mu_{r,i}) \cdot \frac{\xi_{i,t}}{\rho_i^{\max}} + \sum_{t \in T^{0g}} \sum_{i \in I_i^c} ce_i \cdot (-\mu_{r,i}) \cdot \frac{\xi_{i,t}}{\rho_i^{\max}} + cs \sum_{t \in T} S_{r,t}] \quad (7)$$

The complexity of the model is reflected in the number of terms present in the first set of the excess resource balances, eq. 8. The idea is the same as in eq. 2 but now we have to include some entities of the continuous-time model given in Castro et al. (2009a). The first is parameter  $\Pi_{r,t}^{out}$ , which holds the product demands at the end of the demand period  $td$  currently being scheduled in detail (superscript \*). In the time grid, such interaction occurs at the boundary time point. Concerning the utility input, there may be multiple energy pricing levels within a demand period and even different time periods  $tp$  within the same energy level ( $TP_e$ ). It is important to mention at this time that utility availability remains constant throughout the duration of any time period, which are characterized by given lower and upper bounds ( $lb_{e,tp}$ ,  $ub_{e,tp}$ ). To identify during which time period tasks executed at interval  $t$  are processed at (see Figure 10), the binary variables  $Y_{t,tp,e}$  are used. Whenever the power availability is insufficient to meet the demand, the slack variables  $S_{r,t}$  are activated. Notice in the constraint domain that the balance for utility and process equipment resources is only done for the continuous time points but the last.

$$\begin{aligned} R_{r,t} = & R_r^0 \Big|_{t=1} + R_{r,t-1}^{end} \Big|_{r \in R^{CT}} + R_{r,t-1} \Big|_{r \notin (R^{CT} \cup R^{UT})} - (\Pi_{r,t}^{out} + \Pi_{r,t}^{out*} \Big|_{t \in T^{c, last}}) \Big|_{r \in R^{FP}} + \\ & (\Pi_{r,t}^{in} + \sum_{e \in E} \sum_{tp \in TP_e} \Pi_{r,tp,e}^{in} Y_{t,tp,e} + S_{r,t}) \Big|_{t \in T^c \setminus T^{c, last}} \Big|_{r \in R^{UT}} + \\ & \sum_{i \in I} (\mu_{r,i} N_{i,t} \Big|_{i \in I_i^c \vee (i \notin I_i^c \wedge t \neq |T|)} + \nu_{r,i} \xi_{i,t} \Big|_{i \in I_i^c \vee i \notin I_i^c} + \bar{\mu}_{r,i} N_{i,t-1} \Big|_{i \in I_i^c \vee i \notin I_i^c}) + \\ & \sum_{i \in I'} \bar{\nu}_{r,i} \xi_{i,t} \quad \forall t \in T, r \in R, [r \notin (R^{TC} \cup R^{UT}) \vee t \in T^c \setminus T^{c, last}] \end{aligned} \quad (8)$$



**Figure 10.** Time interval  $t$  may include a few time periods  $tp$  but processing tasks can be executed in at most one.

The last terms in eq. 8 are identical to those in eq. 2 besides the stricter domain of the variables associated to the processing tasks. The same can be said for the rest of the constraints of the aggregate part of the model, see eqs. 9-12.

$$R_{r,t}^{end} = R_{r,t} + \sum_{i \in I_t^c} \lambda_{r,i} \xi_{i,t} + \sum_{i \in I^s} (\bar{v}_{r,i} \xi_{i,t} + \lambda_{r,i} \xi_{i,t}^*) \quad \forall r \in R^{CT}, t \in T, t \neq |T| \quad (9)$$

$$\Delta_t \geq \sum_{i \in I_t^c} \frac{\bar{\mu}_{r,i} \xi_{i,t}}{\rho_i} \quad \forall r \in R^{TC}, t \in T^{ag}, t \neq |T| \quad (10)$$

$$\Delta_t \cdot (\Pi_{r,t}^m + S_{r,t}) \geq \sum_{i \in I_t^c} \frac{-\mu_{r,i} \xi_{i,t}}{\rho_i} \quad \forall r \in R^{UT}, t \in T^{ag}, t \neq |T| \quad (11)$$

$$\xi_{i,t} + \xi_{i,t}^* \Big|_{i \in I^s} \leq H \cdot \sum_{m \in M} \max_{p \in P} \rho_{p,m} \cdot N_{i,t} \quad \forall t \in T, i \in I_t^c \vee (i \in I^s \wedge t \neq |T|) \quad (12)$$

As for the continuous part of the model, eqs. 13-17 apply. The timing variables  $T_t$  and  $Ts_t$  provide the absolute time of event point  $t$  and the earliest starting time amongst all processing tasks executed during interval  $t$ , respectively. Clearly, the starting time must be greater than the event point's absolute time (eq. 13), which, for the first point, is given by the location of the previous demand point. Likewise, tasks must end before the next event point. Eq. 14 is written for all processing equipment resources and implicitly assumes that there can be a single task executed on such unit at a certain time, which is ensured by the initial resource availability and excess resource balances. It is crucial to understand that tasks executed during interval  $t$  must lie within a single time period of an energy level, see eq. 17 and also Figure 10. Hence, the starting time must also be greater than the time period's lower bound (eq. 15), and there must be enough time for the tasks to end before the period's upper bound, eq. 16.

$$Ts_t \geq T_t \Big|_{t \notin T^{first}} + t f x_{id^*-1} \Big|_{t \in T^{first}} \quad \forall t \in T^c \setminus T^{clast} \quad (13)$$

$$t f x_{id^*} \Big|_{t+1 \in T^{clast}} + T_{t+1} \Big|_{t+1 \notin T^{clast}} - Ts_t \geq \sum_{i \in I_t^c} \frac{\bar{\mu}_{r,i} \xi_{i,t}}{\rho_i} \quad \forall r \in R^{TC}, t \in T^c \setminus T^{clast} \quad (14)$$

$$Ts_t \geq \sum_{e \in E} \sum_{tp \in TP_e} lb_{e,tp} Y_{t,tp,e} \quad \forall t \in T^c \setminus T^{clast} \quad (15)$$

$$Ts_t + \sum_{i \in I_t^c} \frac{\bar{\mu}_{r,i} \xi_{i,t}}{\rho_i} \leq \sum_{tp \in TP_e} ub_{e,tp} Y_{t,tp,e} + t f x_{id^*} \cdot (1 - \sum_{i \in I_t^c} \bar{\mu}_{r,i} N_{i,t}) \quad \forall r \in R^{TC}, e \in E, t \in T^c \setminus T^{clast} \quad (16)$$

$$\sum_{e \in E} \sum_{tp \in TP_e} Y_{t,tp,e} = 1 \quad \forall t \in T^c \setminus T^{clast} \quad (17)$$



## 8. Computational Results

The performance of the new aggregate and combined formulations is illustrated through the solution of nine example problems. EX2-EX10 are taken from Castro et al. (2009a) and two new cases, EX2a and EX5a, which are slightly modified versions of EX2 and EX5, are added to illustrate that the aggregate formulation may underestimate the total electricity cost in cases of limited power availability. The models are compared to the discrete (DT) and continuous-time (CT) models from the same reference. Nevertheless, they have been slightly altered to incorporate the slack variables, which explains the small differences in computational times. In order to prioritize schedules that meet the maximum power constraints, we have used  $cs=10$  [k€/MW].

**Table 1.** Computational Statistics

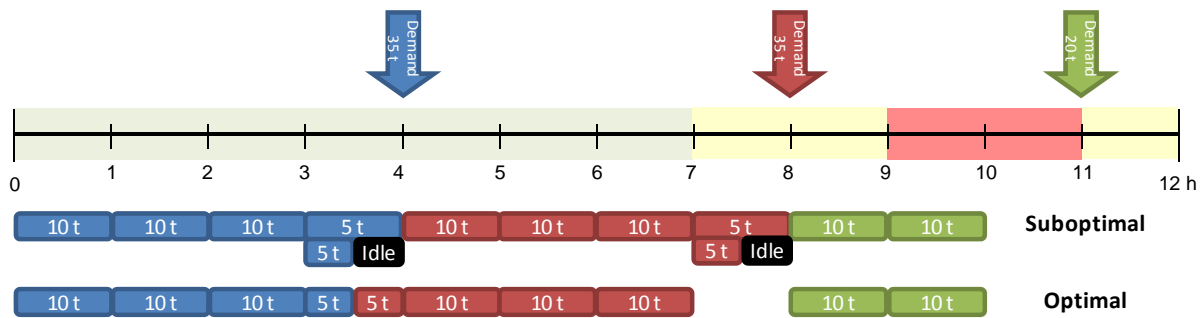
Case	(P,M,S)	Power	Model	T	DV	SV	EQ	RMIP	MIP [€]	CPUs	Gap (%)
Ex2	(2,1,1)	R	DT	169	672	4721	3203	21575	<b>21575</b>	0.24	0
			AG	17	64	431	313	21575	<b>21575</b>	0.12	0
			RH	11	9	54	39	21575	<b>21575</b>	1.11	0
			CT	11	490	808	347	16620	<b>21575</b>	25.3	0
Ex2a	(2,1,1)	R	DT	169	672	4721	3203	21575	<b>21575</b>	0.35	0
			AG	17	64	431	313	18977	<i>18977</i>	0.12	0
			RH	12	9	54	39	21575	<b>21575</b>	1.59	0
			CT	11	490	808	347	16620	<b>21575</b>	16.8	0
EX5	(3,2,2)	U	DT	169	2016	10953	6739	26738	26780	7200	0.04
			AG	19	216	1146	736	26757	<b>26758</b>	0.33	0
			RH	13	50	214	152	26758	<b>26758</b>	2.26	0
			CT	11	660	1364	696	25625	26911	4241	0
EX5a	(3,2,2)	R	DT	169	2016	10953	6739	31351	<b>31798</b>	7200	0.02
			AG	20	228	1208	776	29657	<i>29657</i>	0.24	0
			RH	17	50	214	142	41124	41124	7.06	0
			CT	10	603	1240	629	25625	94901	9829	0
EX6	(3,2,3)	U	DT	169	2520	14155	8423	43250	43259	7200	0.02
			AG	19	270	1498	920	43250	<b>43250</b>	0.37	0
			RH	21	56	262	166	43250	<b>43250</b>	5.57	0
			CT	9	552	1260	646	35517	Inf.	2811	-
EX7	(3,3,4)	U	DT	169	3528	18534	10780	68282	<b>68282</b>	19.9	0
			AG	18	357	1852	1112	68282	<b>68282</b>	0.7	0
			RH	12	44	203	121	68282	<b>68282</b>	3.12	0
			CT	12	880	2134	1156	48852	no sol.	7200	-
EX8	(3,3,5)	R	DT	169	4032	21736	12464	101139	<b>104622</b>	7200	0.22
			AG	19	432	2310	1360	104375	<i>104375</i>	2.05	0
			RH	31	336	1271	777	-	151257	17330	0.16
EX9	(4,3,4)	U	DT	169	4074	24092	13810	87817	87868	7200	0.06
			AG	19	504	2566	1506	87817	<b>87817</b>	0.71	0
			RH	25	53	258	151	87817	<b>87817</b>	917	0
EX10	(5,3,4)	U	DT	169	5880	29650	16840	86505	86582	7200	0.09
			AG	19	630	3174	1836	86505	<b>86550</b>	3.57	0
			RH	23	66	317	181	86550	<b>86550</b>	1508	0

The mathematical formulations give rise to mixed-integer linear programming (MILP) problems. They were implemented in GAMS 22.8 together with the rolling-horizon algorithm. All problems were solved by CPLEX 11.1 with default options. The termination criteria were either a relative optimality tolerance= $10^{-6}$  or a maximum computational time equal to: (i) 7,200 CPUs for the discrete-time model; (ii) 3,600 CPUs per iteration for the problems generated by the rolling horizon algorithm. The hardware consisted on a laptop with an Intel Core2 Duo T9300 processor (2.5 GHz) with 4 GB of RAM running Windows Vista Enterprise.

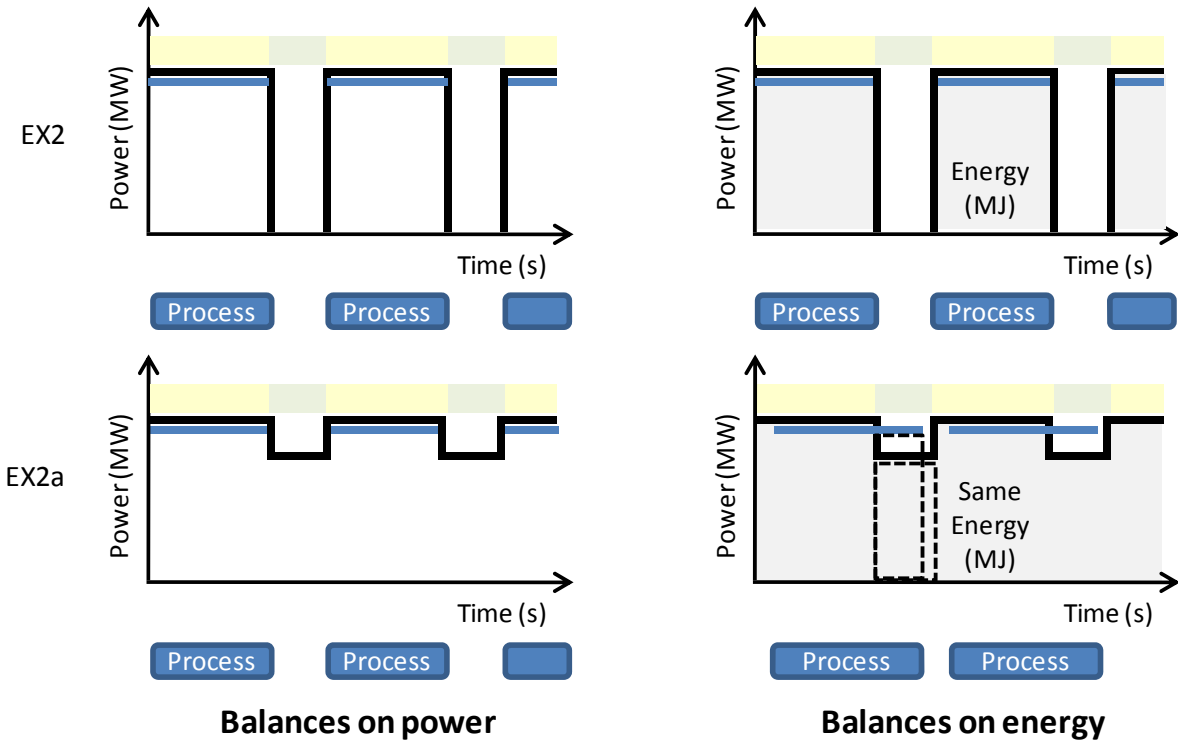
The computational results are given in Table 1 in increasing order of problem size in terms of number of products, units and storage tanks ( $|P|, |M|, |S|$ ). The third column indicates if the power availability is limited (R) or unrestricted (U). The fifth column indicates the composition of the time grid in terms of pre-specified (DT and CT), pre-calculated (AG) or final number of event points in the rolling-horizon algorithm (RH). The model statistics from GAMS come next in terms of discrete (DV), single/total variables (SV) and equations (EQ), while the last column gives the optimality gap.

### **8.1. Uniform Discrete-time vs. Non-uniform Aggregate Approach**

The results in Table 1 clearly show that the new aggregate approach is very powerful. The use of a discrete-time grid with variable interval lengths, instead of a uniform time grid (DT), has a strong impact on problem size and leads to orders of magnitude reduction in computational effort. It adapts to the problem data to generate time grids with roughly 20 time points, instead of the 169, 1-hour intervals of DT. This considering a rather complex electricity cost/availability profile, since simpler profiles would generate even smaller, easier to solve problems. The aggregate planning approach is rigorous for unrestricted power availability and so it was no surprise to observe that it could accurately predict the global optimal cost (shown in bold). In EX5, EX6, EX9 and EX10, the discrete-time model was unable to reach the optimal solution, primarily due to its inherent limitation for handling continuous tasks, which allows meeting the exact product demands at the compromise of having partially occupied time intervals, see Figure 11. This can be verified for EX5 for which the best bound at the time of termination, €26768, is already above the real optimum of €26758.



**Figure 11.** Limitation of discrete-time formulation when handling continuous tasks.



**Figure 12.** Aggregate model allows production in periods with insufficient power availability. The solid lines give the instantaneous power availability, in black, or consumption, in blue. The dashed rectangles give the energy consumption.

For restricted power availability, the solution of the aggregate model provides just a lower bound on the optimal cost (shown in italics in Table 1). Nevertheless, it could still find the optimal value for EX2 but not for EX2a. The differences in the problem data of these two cases are limited to the maximum power availability on the early hours of Tuesday and Thursday (0 in EX2 and 4 MW in EX2a). The single machine instantaneous power consumption is equal to 5 MW so either way there can be no production in such time periods, see Figure 12 on the left. However, while there is no energy available for 0 MW (top right), there is 80% of the amount required for 4 MW. Since the aggregate model cannot distinguish between the energy contained in a 7-hour supply at 4 MW or a

5.6-hour supply at 5 MW (areas within the wider and taller dashed rectangles in the bottom right of Figure 12) it incorrectly directs production to such low cost periods. As a consequence, the solution from AG (€18977) has a lower cost than the global optimal (€21575).

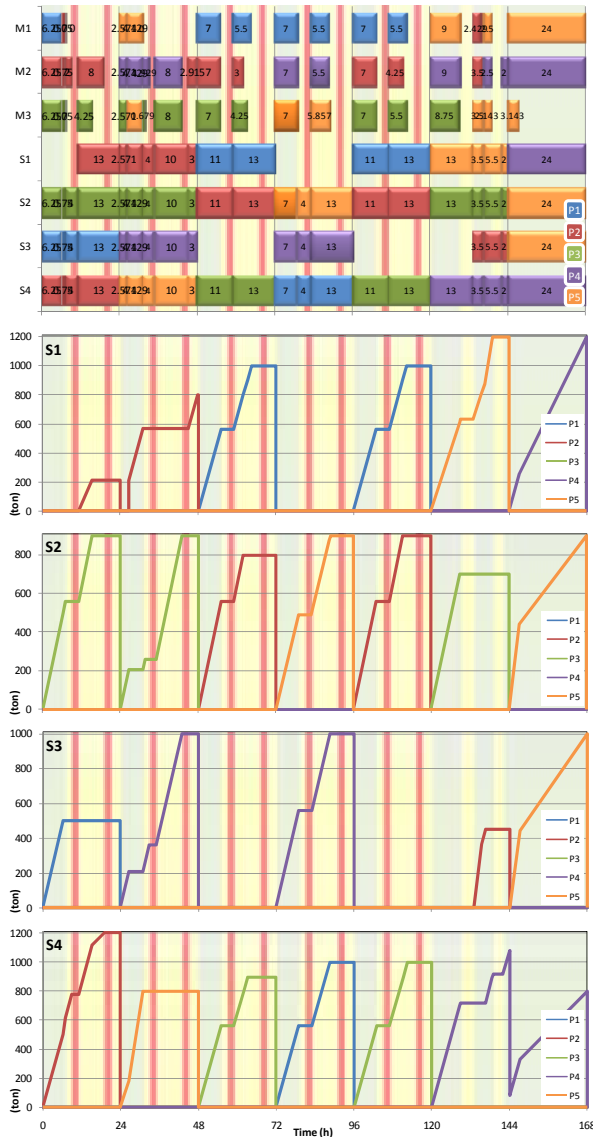
Under restricted power availability the discrete-time formulation is the best approach. It is capable of finding very good solutions with few computational resources even though it cannot fully close the optimality gap. The solutions for EX5a and EX8 are at least 20% better than those resulting from the rolling-horizon algorithm due to the fact that the latter is unable to generate solutions that meet the maximum power constraints throughout the duration of the time horizon. Still some schedule refinement procedure may be required to generate practical solutions (recall Figure 1).

## **8.2. Rolling-horizon vs. Full-space Continuous-time Approach**

The new combined aggregate/continuous-time formulation can, as a part of the rolling-horizon algorithm, successfully find the global optimal solution under unrestricted power availability. This is because it is always looking at the full time horizon with a rigorous model, even though its aggregate part does not provide all the required information for scheduling. Nevertheless, the planned production levels from the aggregate part can always be accomplished, with the problem residing on having enough event points on the time grid. A couple of iterations per demand period were usually enough. Since the lower the number of event points the fewer the changeovers, the outcome from the rolling-horizon algorithm are practical solutions which is not necessarily the case for the discrete-time approach. More importantly, a few seconds of computational time were enough to solve EX5-EX7, while the larger EX9-EX10 could be solved to optimality in half an hour.

The full-space continuous-time model is significantly more complex overall, primarily due to the added variables and constraints that relate event points with demand points (see Castro et al., 2009a). As a consequence, the number of event points in the grid has a stronger influence on the computational time making the iterative search procedure more time consuming, sometimes prohibitively so. This is clear from the following results. For EX5, the optimal solution for  $|T|=11$  takes 4241 CPUs to find and is not the global optimum, which can be found by RH in just 2.26 CPUs. For EX6 it takes 2811 CPUs for CT to find out that 9 event points are not enough to even find

a feasible solution to the problem, while RH finds the optimal solution in 5.57 CPUs (21 event points). For EX7, 12 event points suffice to find the global optimum (RH in 3.12 CPUs) but CT cannot find a feasible solution in 2 hours. Overall, we only get the full potential of the continuous-time formulation when combined with the aggregate model in a rolling-horizon algorithm.



**Figure 13.** Optimal schedule and storage profiles for EX10.

For power restricted problems the performance of the rolling-horizon approach is worse, primarily in terms of the quality of the solution returned as already mentioned. Due to the eventually not so good solution from AG, production may be delayed for future demand periods with cheaper electricity where there simply is not enough power to perform the processing tasks. When the algorithm reaches such periods, it finds out that it needs to violate the maximum power constraints

(by activating the slack variables, which are severely penalized in the objective function) to meet the hard constraints on the demands. It would have been better to produce earlier at a medium or high cost period. This is what happens in EX7 (1 extra MW on Thursday), which is still solved relatively fast, and EX8 (5 extra MW on Thursday), the hardest problem with 4 iterations of RH reaching the maximum resource limit for a total time short of 5 hours. Recall from the discrete-time model that it is possible to meet the demands with the specified maximum power profile.

### **8.3. Optimal Solution for EX10**

To complete the analysis we show in Figure 13 the optimal solution for the largest problem, EX10. One can see that the processing tasks are never executed in the periods where the electricity cost is at its maximum. In fact, with the exception of the last two hours of Saturday, the units are always working in the low-cost periods. The drawback of generating a minimum cost schedule with such electricity cost profile, is that the production stops frequently. Another aspect that adds to the complexity of the schedule results from having one less storage tank than the number of products. Furthermore, they have different capacities, which make the ideal storage tank for a certain product to change throughout the week (Monday-Sunday). It is even possible for a certain product to occupy multiple units at a particular time (e.g. P2 on Monday, P4 and P5 on Sunday). Using shared instead of dedicated storage units increases degeneracy and it is the primary reason why it is difficult to close the optimality gap for the discrete-time formulation. Evidence of degeneracy can be seen for P1 on Thursday, which is sent for storage tank S4 when it would have been more practical to keep sending it to S1 (it is used in the days before and after).

## **9. Conclusions**

This paper has focused on the optimal scheduling of single stage continuous multiproduct plants subject to energy constraints related to time dependent electricity pricing and power availability, and multiple intermediate due dates. A new aggregate mixed integer linear programming model has been developed to plan the production levels for a weekly horizon. It employs a non-uniform discrete time grid where interval duration is calculated following analysis of the problem data. More specifically,

time periods with constant electricity pricing and availability between two successive demand points are grouped together to form the grid's time intervals. Within two consecutive time points, time is allowed to vary continuously to allow production of multiple products in a machine, provided that the total production time does not exceed the interval length. The advantage when compared to a closely related, traditional uniform discrete-time formulation featuring 1-hour intervals, is a lower degree of degeneracy and one tenth of the problem size, which is reflected in up to four orders of magnitude reduction in computational effort. The drawback is that the aggregate model is only accurate for unrestricted power availability. In the other cases, it underestimates the total electricity cost due to the replacement of the instantaneous power balances by energy balances over time, which are just an approximation.

While the aggregate model can efficiently address the full problem it does not provide the actual timing of events, required for detailed scheduling. To overcome this limitation it has been combined with a single time grid continuous-time formulation and incorporated into a new rolling-horizon algorithm. Scheduling is performed sequentially, one demand period after the next, with the current period being tackled by the continuous-time part and the remaining demand periods by the aggregate part of the model. Besides allowing to consider the whole remaining problem simultaneously, the solution from the aggregate part of the model can also be used to predict the number of event points for the continuous part of the time grid, effectively reducing the number of iterations and computational time in the search for the global optimal solution. The rolling horizon algorithm has been shown to efficiently solve real-life problems with unlimited power availability to global optimality. Under limited power availability, the inaccuracy of the aggregate part of the model may lead to suboptimal solutions, with the unnecessary violation of the power constraints (severely penalized in the objective function) being particularly relevant.

## **Acknowledgments**

The authors gratefully acknowledge financial support from Fundação Luso-Americana and the Center for Advanced Process Decision-making at Carnegie Mellon University.

## Nomenclature

### Sets/Indices

$DY/dy$ =days of the week

$E/e$ =Electricity cost levels

$HR/hr$ =hours of the day

$I/i$ =tasks

$I^c$ =continuous tasks

$I_e^c$ =continuous tasks executed during energy cost level  $e$

$I^s$ =storage tasks

$I^t$ =instantaneous transfer tasks

$M/m$ =machines

$P/p$ =products

$R/r$ =resources

$R^{CT}$ =resources continuously produced/consumed

$R^{EQ}$ =equipment resources

$R^{FP}$ =final product resources

$R^{LM}$ =products at a location after the machines

$R^{LS}$ =products at a location inside the storage units

$R^{RM}$ =raw-material resources

$R^{TC}$ =equipment resources involved in timing constraints

$R^{UT}$ =utility resources

$S/s$ =storage units

$T/t$ =event points

$T^{ag}$ =discrete time points of the aggregate part of the time grid

$T^{first}$ =first event point of the time grid

$T^c$ =event points of the continuous part of the time grid

$T^{c\ last}$ =last event point of the continuous part of the grid and first element of aggregate part

$T^{last}$ =last event point of the time grid

$TD/td$ =demand points

$TP/tp$ =time periods of electricity cost level  $e$

### Parameters

$c_e$ =electricity cost at level  $e$  [€/kWh]

$cap_s$ =capacity of storage unit  $s$  [ton]

$ce_t$ =electricity cost during time interval  $t$  [€/kWh]

$cs$ =penalty cost for violating the given maximum power levels [k€/MW]

$d_{p,dy,hr}$ =demand of product  $p$  at the end of hour  $hr$  of day  $dy$  [ton]

$ec_{hr,dy}$ = electricity cost during hour  $hr$  of day  $dy$  [€/kWh]

$H$ = time horizon [h]

$im_{s,p}$ =initial amount in storage  $s$  of product  $p$  [ton]

$lb_{e,tp}$ =starting time of time period  $tp$  of electricity cost level  $e$  [h]

$pcost$ =partial cost in demand periods already scheduled [k€]

$pw_{p,m}$ =power requirement for product  $p$  in machine  $m$  [MW]

$pw_{x,hr,dy}$ =maximum power consumption during hour  $hr$  of day  $dy$  [MW]

$R_r^{\max}$ =upper bound on availability of resource  $r$  at event points

$R_r^{end\max}$ = upper bound on availability of resource  $r$  during intervals

$R_r^{\min}$  = lower bound on availability of resource  $r$  at event points

$R_r^{end\min}$  = lower bound on availability of resource  $r$  during intervals

$R_r^0$  = initial amount of resource  $r$



$tf_{x_{id}}$ =absolute time of demand point  $td$  [h]  
 $ub_{e,tp}$ =ending time of time period  $tp$  of electricity cost level  $e$  [h]  
 $\Delta_t$ =duration of interval  $t$  on non-uniform discrete-time grid [h]  
 $\lambda_{r,i}$ =continuous interaction of resource  $r$  during execution of task  $i$   
 $\mu_{r,i}$ =discrete interaction of resource  $r$  with task  $i$  at its start, acting on binary extent variables  
 $\bar{\mu}_{r,i}$ =discrete interaction of resource  $r$  with task  $i$  at its end, acting on binary extent variables  
 $v_{r,i}$ = discrete interaction of resource  $r$  with task  $i$  at its start, acting on continuous extent variables  
 $\bar{v}_{r,i}$  = discrete interaction of resource  $r$  with task  $i$  at its end, acting on continuous extent variables  
 $\Pi_{r,t}^{in}$ =amount received by the system of resource  $r$  at event point  $t$   
 $\Pi_{r,tp,e}^{in}$ =amount received by the system of resource  $r$  at time period  $tp$  of electricity level  $e$   
 $\Pi_{r,t}^{out}$ =amount removed from the system of resource  $r$  at event point  $t$   
 $\Pi_{r,td}^{out}$ =amount removed from the system of resource  $r$  at demand point  $td$   
 $\rho_i^{\max}$ =maximum processing rate of task  $i$  [ton/h]  
 $\rho_{p,m}$ =processing rate of product  $p$  in machine  $m$  [ton/h]

### *Variables*

$N_{i,t}$ =execution of task  $i$  during interval  $t$  (binary extent variables)  
 $R_{r,t}$ =excess amount of resource  $r$  at event point  $t$   
 $R_r^0$  = initial availability of resource  $r$  (can be a parameter for some resources)  
 $R_{r,t}^{end}$ =excess amount of resource  $r$  immediately before the end of interval  $t$   
 $S_{r,t}$ = slack variable on utility resource  $r$  at event point  $t$   
 $T_t$ =absolute time of event point  $t$  [h]  
 $Ts_t$ =starting time of tasks executed during interval  $t$  [h]  
 $Y_{t,tp,e}$ =binary variable identifying if during interval  $t$  tasks are executed within period  $tp$  of level  $e$   
 $\zeta_{i,t}$ =amount handled by task  $i$  at event point/during interval  $t$  (continuous extent variables) [ton]  
 $\zeta_{i,t}^*$ =amount continuously sent to storage by task  $i$  during interval  $t$  [ton]

## References

- Castro, P.; Barbosa-Póvoa, A.; Matos, H.; Novais, A. (2004). Simple Continuous-time Formulation for Short-Term Scheduling of Batch and Continuous Processes. *Ind. Eng. Chem. Res.* *43*, 105.
- Castro, P.; Barbosa-Póvoa, A.; Novais, A. (2005). Simultaneous Design and Scheduling of Multipurpose Plants Using Resource Task Network Based Continuous-Time Formulations. *Ind. Eng. Chem. Res.* *44*, 343.
- Castro, P.; Harjunkski, I.; Grossmann, I. (2009a). New Continuous-Time Scheduling Formulation for Continuous Plants under Variable Electricity Cost. *Ind. Eng. Chem. Res.* *48*, 6701.
- Castro, P.; Westerlund, J.; Forssell, S. (2009b). Scheduling of a continuous plant with recycling of byproducts: A case study from a tissue paper mill. *Comput. Chem. Eng.* *33*, 347.
- Chen, C.-L.; Chang, C.-Y. (2009). A resource-task network approach for optimal short-term/periodic scheduling and heat integration in multipurpose batch plants. *Applied Thermal Engineering* *29*, 1195.
- Dimitriadis, A.D.; Shah, N.; Pantelides, C.C. (1997). RTN-Based Rolling Horizon Algorithms for Medium Term Scheduling of Multipurpose Plants. *Comput. Chem. Eng.* *21*, S1061.
- Duarte, B.; Santos, L.; Mariano, J. (2009). Optimal sizing, scheduling and shift policy of the grinding section of a ceramic tile plant. *Computers Operations Research* *36*, 1825-1834.
- Ferrer-Nadal, S.; Capón-García, E.; Méndez, C.; Puigjaner, L. (2008). Material transfer operations in batch scheduling. A critical modeling issue. *Ind. Eng. Chem. Res.* *47*, 7721.
- Giménez, D.; Henning, G.; Maravelias, C. (2009). A novel network-based continuous-time representation for process scheduling: Part I. Main concepts and mathematical formulation. *Comput. Chem. Eng.* *33*, 1511.
- Grossmann, I.E. (2005). Enterprise-wide optimization: A new frontier in Process Systems Engineering. *AIChE J.* *51*, 1846.
- Kondili, E.; Pantelides, C. C.; Sargent, R. (1993). A General Algorithm for Short-Term Scheduling of Batch Operations - I. MILP Formulation. *Comput. Chem. Eng.* *17*, 211.
- Maravelias, C.T. (2005). Mixed-time Representation for State-Task Network Models. *Ind. Eng. Chem. Res.* *44*, 9129.

Maravelias, C.; Grossmann, I. (2003). A General Continuous State Task Network Formulation for Short Term Scheduling of Multipurpose Batch Plants with Due Dates. *Proceedings ESCAPE-13* (Eds. A. Kraslawski and I. Turunen), pp. 215.

Méndez, C.A.; Cerdá, J.; Grossmann, I.E.; Harjunkoski, I.; Fahl, M. (2006). State-of-the-art Review of Optimization Methods for Short-Term Scheduling of Batch Processes. *Comput. Chem. Eng.* 30, 913.

Pantelides, C.C. (1994). Unified Frameworks for the Optimal Process Planning and Scheduling. In *Proceedings of the Second Conference on Foundations of Computer Aided Operations*; Cache Publications: New York; pp 253.

Shaik, M.; Floudas, C. (2009). Novel Unified Modeling Approach for Short-Term Scheduling. *Ind. Eng. Chem. Res.* 48, 2947.

Westerlund, J.; Hastbacka, M.; Forssell, S.; Westerlund, T. (2007). Mixed-Time Mixed-Integer Linear Programming Scheduling Model. *Ind. Eng. Chem. Res.* 46, 2781.