A New Continuous-Time Scheduling Formulation for Continuous Plants under Variable Electricity Cost

Pedro M. Castro,[*,†,◊], *Iiro Harjunkoski*[‡] *and Ignacio E. Grossmann*[◊]

[†] Departamento de Modelação e Simulação de Processos, Instituto Nacional de Engenharia, Tecnologia e Inovação, 1649-038 Lisboa, Portugal

[‡]ABB Corporate Research Center, Wallstadter Str. 59, 68526 Ladenburg, Germany,

[◊] Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213.

Abstract

This work addresses the scheduling of continuous plants subject to energy constraints related to time-dependent electricity pricing and availability. Discrete and continuous-time formulations are presented that can address these issues together with multiple intermediate due dates. Both formulations rely on the resource-task network process representation. Their computationally performance is compared for the objective of total electricity minimization with the results favoring the discrete-time model due to the more natural way of handling such a wide variety of discrete events. In particular, it can successfully handle problems of industrial size. Nevertheless, the new continuous-time model is a major breakthrough since it is the first model of its type that is able to effectively incorporate time-variable utility profiles. When compared to a simple manual scheduling procedure, the proposed scheduling approaches can lead to potential electricity savings around 20% by switching production from periods of high to low electricity cost.

---

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

1. Motivation

Enterprises are currently under pressure to produce at the lowest possible cost within continuously changing economic constraints[1]. To achieve this goal, they must actively look at the best operating practices and optimize these both globally and locally. Within this overall goal, scheduling plays an important part.

This work is motivated by a real industrial problem that we cannot disclose for confidentiality reasons. It involves the final stage of a multiproduct plant with electricity intensive parallel equipment units, where scheduling involves deciding when each unit has to produce a certain product. Most of the times, this is made manually by the operator according to heuristic rules. The products are then sent to storage units, where they are stored until dispatching takes place. Meeting customer demands on time is vital, and for this reason, in some plants, no other factors are taken into account besides trying to keep the storage units full in order to be able to fulfill the orders. Plant scheduling is difficult due to the following factors: large combinatorial size arising from the number of equipment units, products and storage units; various operating and contractual constraints; liberalized electricity market with nontransparent billing practices. Due to the inherent complexity, the operator scheduling choices may be far from the optimal ones.

The most challenging aspect of plant scheduling is undoubtedly the incorporation of energy constraints related to electricity pricing and availability. We consider in this paper the case where in the planning stage, contracts are agreed between the electricity supplier and the plant, which often specifies maximum levels of power usage. If electricity consumption exceeds this threshold, the plant incurs in stiff penalties, whereas underproduction costs the same as planned production. Even for normal production, electricity cost varies significantly throughout the day, and this must be taken into consideration within the modeling framework. Another important aspect, concerns meeting the sales forecasts, which represent the minimum amount of products that has to be available in the storage units, during the time horizon. These are typically considered to occur at the end of each day.

Accounting for events that occur at predetermined points of time, for instance at a specific hour of a given day, being either a change in electricity cost level or the occurrence of a demand point, may be relatively easy or extremely complex, depending on the type of time representation employed. This is straightforward with a discrete-time approach, whenever a sufficiently fine time grid can be used. In such cases, the preselected time of some of the grid time points, will match exactly those of the points of change, and so the constraints are easier to enforce. In contrast, with continuous-time, the absolute time of all event points is determined by the solver and thus it is much harder to relate the events with the points of change.

This paper presents a new continuous-time formulation that effectively handles time dependent cost parameters and discrete demand points. Incorporation of the former aspect within a continuous-time formulation has not been reported before to the best of our knowledge, whereas the constraints used to model the latter aspect are conceptually similar to those used by Maravelias and Grossmann[2]. The proposed approach builds on the general multipurpose formulation of Castro et al.[3], which can address problems involving batch and continuous tasks, efficiently. Indeed, for batch plants, a study[4] has found it to be the best single time grid formulation. Nevertheless, the constraints that are given are only suitable for continuous plants, merely because batch tasks, unlike continuous tasks, cannot be divided in as many times as required. In other words, while the output material from a batch task is produced entirely at its end, after a specified period of time, the one from a continuous task is continuously being produced. Thus, the execution of a single instance of a continuous task is equivalent to the execution (at the same processing rate) of multiple instances producing the same total amount of material, in sequence, during the same time period. The new formulation is also more detailed with respect to the resource balances. In order to rigorously account for inventory constraints, balances need to be considered both at the start and end of time slots, so that the effect of resource consumption and generation occurring due to continuous tasks is considered in full, as explained in Schilling & Pantelides[5].

The new formulation is built on a unified framework for process representation, the Resource-Task Network (RTN) of Pantelides[6]. This means that the model variables and constraints are written is terms of abstract entities like resources, tasks and event points, so it has a much wider scope that the single stage industrial case study used for illustrative purposes.

2. Problem Definition

During the final stage of the process, an intermediate material is transformed into one of different final products, characterized by chemical composition and particle size distribution, through the use of electricity. These are then sent to storage units, where they wait until customer dispatch takes place. This process is illustrated in Figure 1. It emphasizes that a particular storage unit may be suitable for just a subset of the products. In fact, the product allocated to a storage unit normally never changes. Since it is straightforward to incorporate such constraints in the upcoming models, it is preferable to consider full connectivity between units to increase their flexibility. In other words, we will be assuming that every storage unit can handle all products (shared storage), but only one at a time.

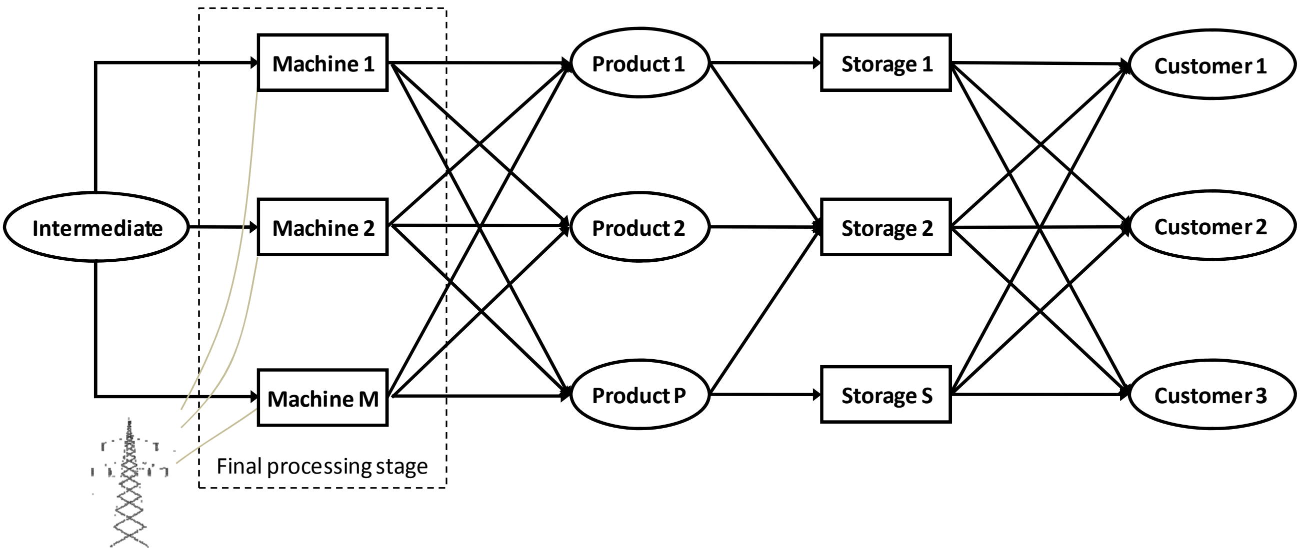

Figure 1. Final processing stage of industrial case study

Typical plant schedules are established over one week, so this will be the time horizon (H=168 h) assumed for the remainder of the paper. Naturally, it is straightforward to consider other values. Let M represent the set of machines, P the set of products and S the storage units. Machines are characterized by: (i) power requirements for each product $pw_{p,m}$ [MW]; (ii) processing rates, $\rho_{p,m}$ [ton/h]. Let DY and HR be sets whose elements are the days of the week and the hours of the day, respectively. Each product may have multiple demands over the week, and can occur at any hour of the day, $d_{p,dy,hr}$ [ton]. Finally, storage units have known maximum capacities, cap_s [ton].

The energy contract signed by the plant and electricity provider establishes a certain pricing policy. Electricity cost is typically lower during the night and higher during the day. Similarly to the product demands, it is generally assumed that the cost can change every hour of every day[19]. The maximum level of total power consumption is in turn specified through the parameter $pwx_{hr,dy}$ [MW]. For illustration purposes, the energy policy given in Duarte et al.[7], is used. It consists of three energy levels, *E*, with prices c_e of 0.0481, 0.0945 and 0.2162 [€/kWh]. The weekly distribution is given in Figure 2.

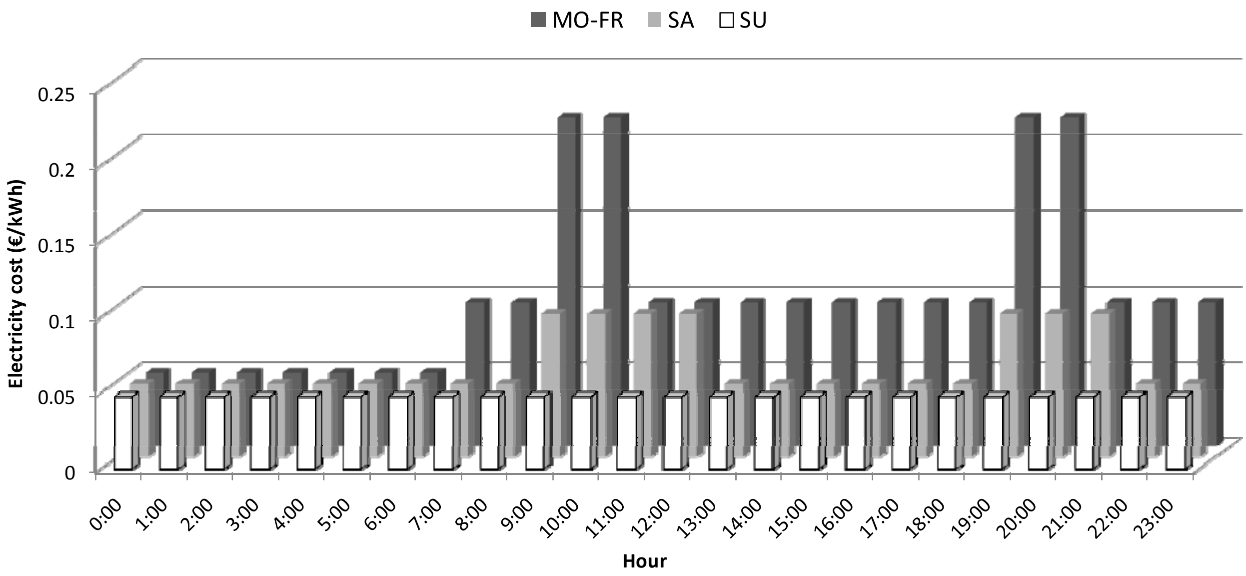

Figure 2. Electricity cost policy within a working week.

The objective will be to minimize the total energy cost subject to constraints on resource availability that includes processing and storage units, and utilities (electricity).

3. Resource-Task Network Representation of the Process

The models proposed in sections 5 and 6 are built on the Resource-Task Network to make them as general as possible. The next step is thus to identify the set of process resources and tasks, which will be identified as a member of one or more subtypes, in order to better understand the structural parameters generation process and the different terms in the model constraints. For that purpose, we use the guidelines given in Castro et al.[8], to define new subtypes such that the specifics of the industrial problem at hand can be incorporated. The final result is the RTN given in Figure 3. Note that it is helpful to generate the complete drawing even though it is not absolutely necessary.

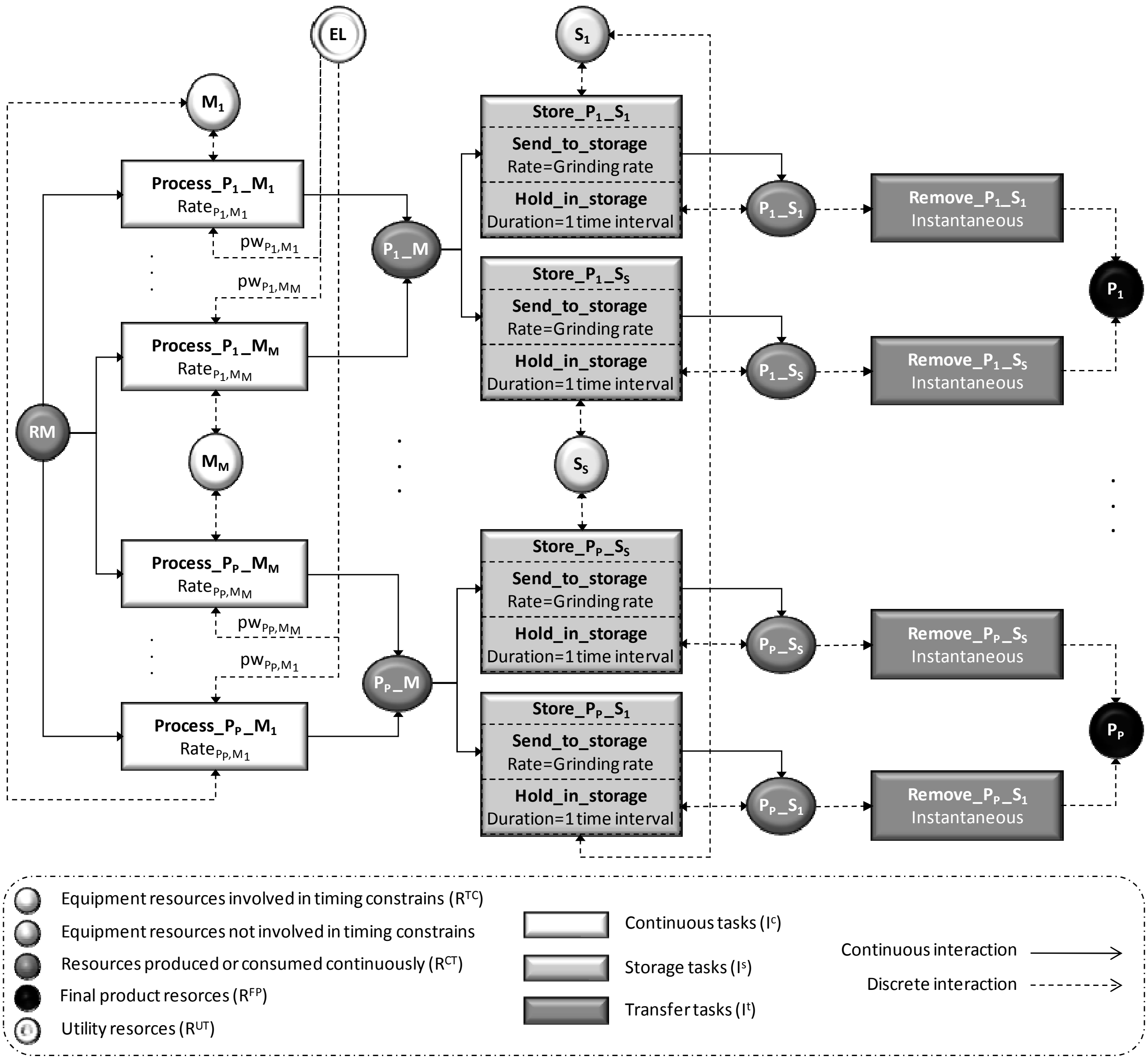

Figure 3. Resource-Task Network representation of final processing stage.

3.1. Resources Identification

In an RTN, resources are represented as circles. Going from left to right in Figure 3, we have the material input to the scheduling problem that will be called raw-material, and named RM (the sole member of set R^{RM}), despite the fact that it is in reality an intermediate process material. It is continuously consumed while a product is being produced. While already in the state of final product, it still has to go through storage and wait until dispatching. It is thus required to identify product location, which can be in three different places: (i) immediately after the machines, R^{LM}; (ii) contained in storage, R^{LS}; or (iii) inside the final transportation vessel, R^{FP}. In the first situation there is no need to differentiate the machine where the product is produced, so a total of $|P|$ resources are generated, from P_1_M to P_P_M, where the M stands for after the machines. In contrast, in the second possible place, we have to distinguish in which storage unit (or group of units working together) the product is located, since they can have different capacities. The result is the generation of additional $|P| \times |S|$ resources. Resources continuously produced and/or consumed, are the elements of set $R^{CT}=R^{RM}\cup R^{LM}\cup R^{LS}$. These are colored in dark grey to facilitate identification. The final location gives rise to the final product resources, P_1 to P_P, which are filled in black. And this ends the material resources.

Moving on to the equipment resources, set R^{EQ}, we have processing equipment, storage equipment and eventually transportation equipment. They are however handled differently, so subset R^{TC} is needed. This includes solely the former type, which are generally the most important resources in a scheduling problem. They will have a strong impact on the final form of the continuous-time grid due to their participation in the timing constraints. In this specific case, the machines are the members of R^{TC} and are filled in shadowed white, unlike storage ubits, which have a light grey background.

Finally there are the utilities, R^{UT}, the subject of this paper. Continuous-time formulations[9-12] have so far considered a maximum utility availability that remains constant throughout the time horizon. For steam and cooling water we are dealing with flowrate values, typically given in [ton/h], while for electricity we deal with power [MW]. However, as discussed earlier (section 2), we will generally assume a utility availability profile, where in the limit, the specified maximum availability can change every hour of the day during the entire week.

3.2. Tasks Identification

Tasks are represented as rectangles in an RTN, and must be characterized fully in terms of its resources, unlike in a State-Task Network (STN) representation[9,10,13]. More specifically, the STN representation of this process would show just $|P|$ processing tasks since there is no need to explicitly say in which unit the product is going to be executed. In contrast, in the RTN of Figure 3, the processing tasks must also be disaggregated into all the possible machines.

In Figure 3, there are three types of tasks. The processing tasks, filled in white, are performed continuously at a known rate, set I^c. The execution of a processing task involves four resources, the ones linked by arrows to the tasks. In order to differentiate between continuous and discrete interaction, solid and dashed lines are respectively used. Some lines will have arrows on both directions, denoting consumption of the resource at the start of the task and production at its end. As an example, *Process_P_1_M_1* continuously consumes *RM* while producing *P_1_M*. It also consumes power from electricity *EL* and machine M_1 at its start, only to regenerate M_1 at its end.

Storage tasks come next (I^s, in light grey). They are more complex than those described in Castro et al.[8], since now the storage units are shared rather than dedicated to a single material resource. These storage tasks are hybrid, consisting of two parts: (a) *Send_to_storage*, continuous; (b) *Hold_in_storage*, batch. Any part will trigger the consumption of the corresponding destination storage unit (e.g. S_1) such that it is made unavailable for other products during that time. Both can be active simultaneously, in order for a product to enter a partially filled storage unit with the same quality. The continuous part consumes a product-after-machines resource (e.g. P_1M) while it is being produced by *Process_P_1_M_1*,…, *Process_P_1_M_M*, and produces a product-at-storage resource (e.g. *P_1_S_1*). The batch part can be viewed as temporarily hiding the material resource (e.g. *P_1_S_1*) for one time interval.

The last subset of tasks, transfer the products from storage to customers, I^t (dark grey). They are assumed instantaneous, meaning that they last much shorter when compared to the processing tasks. They could have also been easily defined as batch or continuous. We are also assuming that no equipment resource is involved, which is again not in any way limiting.

3.2.1. Further Requirements for Continuous-time Model

Ideally, the RTN representation of the process should be independent on the model that is employed to solve the scheduling problem. However, this is not always the case. In the problem at hand, the electricity cost parameter is time dependent and we need to make sure that the correct parameter is considered. While in discrete-time this is not particularly difficult since the energy level can be directly linked to a subset of the time intervals, in continuous-time the link is at the task level.

In other words, we need to distinguish whether the task is executed at e_1, e_2, or e_E, with subset I_e^c indicating the processing tasks processed at energy level e. Further disaggregation of the processing tasks is thus required (Figure 4), leading to a total $|P|\times|M|\times|E|$ of such tasks. Nevertheless, the exact same resources are involved so Figure 3 is still very much useful.

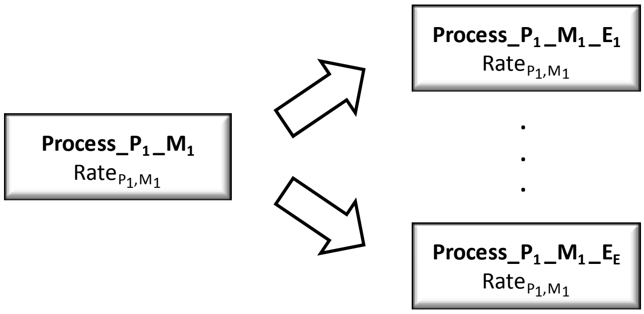

Figure 4. Due to time dependent electricity costs, processing tasks need to be further disaggregated for the continuous-time model.

4. RTN-based Model Entities

This section focus on the model entities that are used regardless of the time representation employed by the Resource-Task Network based model. In the case of the parameters, this does not necessarily mean that they have the same domain or take the same values. Additional variables for the continuous-time model are explained following the description of the underlying time grid, in section 5.1.

4.1. Variables

Tasks are generally characterized by extent variables, one binary, $N_{i,t}$, and one continuous set, $\xi_{i,t}$. The former identify the start of task i at time point t, while the latter normally give the amount handled by the task. We are implicitly assuming that the tasks are either instantaneous (start and end at t) or continuous, which can be made to last a single time interval without loss of generality[3]. In practice, this means that a few consecutive instances of the task will have to be performed to meet large demands. Batch tasks that last one time interval independently of its duration are also included. In fact, the storage tasks described in section 3.2 are hybrid, consisting of a continuous and a batch part that can occur simultaneously, so another set of continuous extent variables, $\xi^*_{i,t}$, is required for their full characterization. It is also important to highlight that no equipment resource is involved in the transfer tasks. Thus, binary variables are not necessary.

The breakthrough of the RTN[6] comes from the unified treatment of resources. The mathematical formulations keep track of resource availability over time through the excess resource variables $R_{r,t}$. The word excess is crucial, with non-zero values indicating that there is still some amount of resource r available at event point t beside the amount that is allocated to batch tasks that will start, or instantaneous tasks that will be executed at t. Equipment resources are mostly treated as individuals, so having $R_{r,t}$=0 indicates that there is one task starting at t that uses unit r. If one wants to force such equipment to be idle during t then it must be available in excess and the constraint to use is $R_{r,t}$=1.

In the presence of continuous tasks, another set of excess resource variables[5], $R_{r,t}^{end}$, is sometimes required. These give the excess amount of resource r immediately before the end of interval t. In this particular problem there were two reasons for their use: (i) the need to occupy a storage unit right from the start of a processing task, in order to ensure that there is one available; (ii) the need to know the amount in storage, to guarantee that the maximum capacity is not exceeded. Events triggered by the start or end of tasks will affect the two types of excess resource variables differently (see section 5.3).

Resource availability at the beginning of the time horizon, R_r^0, is often known for all resources and so it is normally defined as a parameter. The value for equipment units is given in Eq. (1). Nevertheless, the scope of the mathematical formulation is wider if such an entity is a variable. This is particularly useful if one wants to solve the simultaneous design and scheduling problem[8], where one of the goals is to select the most appropriate units for the plant. In this particular case, and even though the raw-material requirements can easily be determined from the product demands that are fixed, the initial availability of the raw-material resource $r \in R^{RM}$, will be a variable in the model. It will be the only input of raw-material to the system (it is straightforward to consider otherwise) and so it will be equal to the total amount required over the full time horizon. For all other resources it will be set to zero (Eq. 2).

$$R_r^0 = 1 \quad \forall r \in R^{EQ} \tag{1}$$

$$R_r^0 = 0 \quad \forall r \notin R^{EQ} \cup R^{RM} \tag{2}$$

4.2. Structural Parameters

The RTN representation of the process is brought into the model by the structural parameters. These can be slightly different depending on whether we are using a discrete or continuous-time formulation[14]. However, in problems with tasks lasting at most a single time interval, the discrete-time formulation can use the exact same parameters as for the continuous-time[17] model. Structural parameters give either the total resource consumption/production, or the proportion relative to the amount handled by the task.

There are five sets of structural parameters, corresponding to different types of interaction with the extent variables. Discrete interactions occur either at the start or end of the task, while a continuous interaction takes place throughout the execution of the task. Parameters $\mu_{r,i,}$ and $\overline{\mu}_{r,i}$ are the discrete interactions associated with variables $N_{i,t}$, respectively for the start and end of the task. They are used whenever the amount of resource consumed/produced is independent on the amount handled by the task, as for equipment units. For material resources one normally relies on the discrete interactions associated with the continuous extent variables $\xi_{i,t}$, i.e. parameters $\nu_{r,i}$ and $\overline{\nu}_{r,i}$. Finally, parameters $\lambda_{r,i}$ hold the continuous interactions associated with the continuous extent variables, either $\xi_{i,t}$ or $\xi_{i,t}^{*}$.

While there can be many parameters in total, the large majority will be equal to zero and those that are not will mostly have a value of either 1 or -1. After some practice, deriving them from the RTN becomes a relatively easy task. As an example, for the case study at hand we have:

$$\mu_{M1,i}\text{=-}1;\ \overline{\mu}_{M1,i} = 1;\ \mu_{EL,i}\text{=-}pw_{P1,M1};\ \lambda_{RM,i}\text{=-}1;\ \lambda_{P1\_M,i}\text{=}1\ \text{-}\forall i\text{=}Process\_P_1\_M_1 \tag{3}$$

$$\mu_{S1,i}\text{=-}1;\ \overline{\mu}_{S1,i} = 1;\ \nu_{P1\_S1,i}\text{=-}1;\ \overline{\nu}_{P1\_S1,i} = 1;\ \lambda_{P1\_M,i}\text{=-}1;\ \lambda_{P1\_S1,i}\text{=}1\ \forall i\text{=}Store\_P_1\_S_1 \tag{4}$$

$$\nu_{P1\_S1,i}\text{=-}1;\ \overline{\nu}_{P1,i} = 1\ \forall\ i\text{=}Remove\_P_1\_S_1 \tag{5}$$

4.3. Other Parameters

While structural parameters link tasks with resources, there are also parameters that are either related to tasks or to resources. These have already been given for the process entities such as machines and products, so now we just need to make the correspondence to the abstract entities of the RTN before assigning the appropriate values. An algorithm was devised for this purpose that simply generates the tasks and resources from the set of products, storage units, machines and electricity levels, while defining functions (f_x) to keep track of the correspondence. These may also depend on whether the problem is to be solved with a continuous or discrete-time formulation (*c* or *d* superscript, respectively).

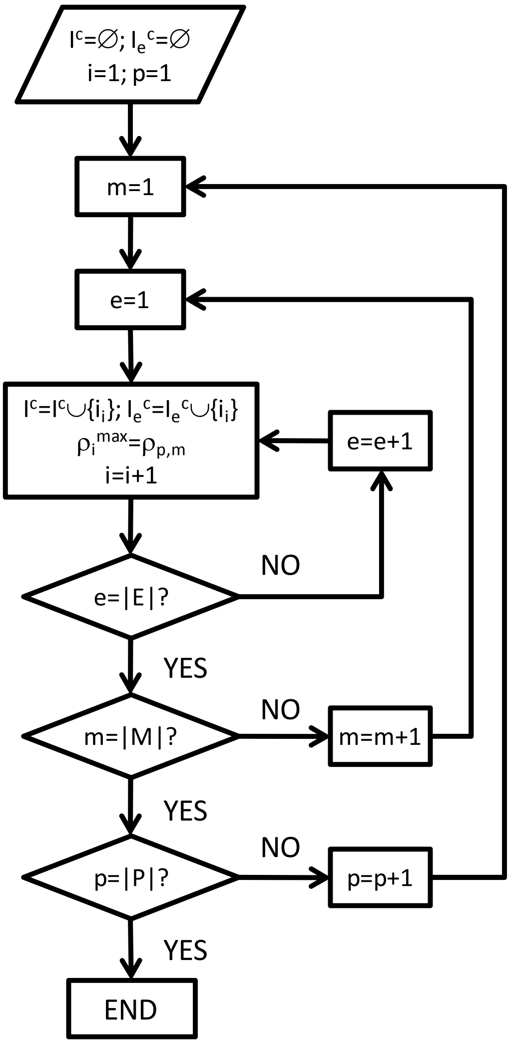

Figure 5. Part of the algorithm responsible for generation of sets I^c and I_e^c.

Figure 5 illustrates the part of the algorithm responsible for generating the continuous tasks (I^c) and the elements of set I_e^c. As an example for $|P|$=2, $|M|$=1 and $|E|$=3, we get I^c={i_1, …, i_6} and

$I_{e_1}^{c} = \{i_1, i_4\}$, $I_{e_2}^{c} = \{i_2, i_5\}$ and $I_{e_1}^{c} = \{i_3, i_6\}$. Along the way, the tasks maximum processing rates $\rho_i^{\max}$ are made equal to the processing rates of the corresponding product-machine pair, $\rho_{p,m}$. This can be generically represented by eq 6.

$$\rho_i^{\max} = f_1(\rho_{p,m}) \quad \forall i \in I^c \tag{6}$$

The algorithm also generates the different demand points $td \in TD$ and the time periods $tp \in TP_e$ for electricity levels e, together with their fixed, tfx_{td}, and starting and ending times, $lb_{e,tp}$ and $ub_{e,tp}$ [h]. Both the profile of cost and maximum power availability, influence the number of time periods. Taking as an example the profile of Figure 2 and assuming constant maximum power availability, there are a total of 8, 17 and 10 periods, for the cheap, medium and expensive cost levels, respectively.

For the discrete-time formulation, the electricity cost levels, c_e, can be easily associated to specific time intervals. It was assumed that the cost can change every hour so there is no problem if 1 is an integer multiple of the duration of every time interval, δ. For the other cases, a weighted mean can be used, even though the total cost will not be exact whenever there are partially executed tasks (see later on section 6.1). Equation (7) emphasizes that the cost for interval t, ce_t, is a function of δ.

$$ce_t = f_2^d(c_e, \delta) \quad \forall t \in T, t \neq |T| \tag{7}$$

Excess resource variables may be subject to given upper ($R_r^{\max}$, $R_r^{end\max}$) and lower bounds ($R_r^{\min}$, $R_r^{end\min}$). The latter are set to zero while one should also do the same for the former, whenever possible. In this way, solution degeneracy will be reduced by ensuring that tasks are executed sequentially without waiting periods in between. For instance, doing this for products located immediately after the machines will force sending to storage the exact same amount that is processed. On the other hand, products should be allowed to exist in storage vessels (R^{LS}), up to an amount equal to the maximum capacity, Eq. (8). Nevertheless, an excess of such resources is only temporary either because they will be transferred to the client, or because they need to be hidden temporarily so that other products do not occupy the same storage vessel. This is accomplished through Eq. (9). We ensure that the system receives the minimum amount of raw-material if there is none available at the last event point (Eq. 10).

$$R_{r,t}^{end\,\max} = 0 + f_3(cap_s)\big|_{r\in R^{LS}} \quad \forall r \in R^{CT} \setminus R^{RM}, t \in T, t \neq |T| \tag{8}$$

$$R_{r,t}^{\max} = 0 \quad \forall r \in R^{CT} \cup R^{FP} \setminus R^{RM}, t \in T \tag{9}$$

$$R_{r,|T|}^{\max} = 0 \quad \forall r \in R^{RM} \tag{10}$$

The system can exchange material with its surroundings at any time. We know exactly when a certain quantity of a resource becomes available to, or is removed from, the system. In a discrete-time model there is a direct link between real time and the time points of the grid, so parameters $\Pi_{r,t}^{in}$ and $\Pi_{r,t}^{out}$ can easily be derived, see Eqs. (11-12). In this case, they will be used to provide the machines with the maximum level of power consumption, and to meet the product demands. When using continuous-time formulations, however, the correspondence between the discrete inputs/outputs and the event points of the time grid will be accomplished through the use of two additional sets of binary variables. Thus, the time index of the input parameters needs to be changed to time period *tp* of energy level *e*, while that of the output parameters is changed to demand point *td* (Eqs. 13-14).

$$\Pi_{r,t}^{in} = f_4^d(pwx_{hr,dy}) \quad \forall r \in R^{UT}, t \in T, t \neq |T| \tag{11}$$

$$\Pi_{r,t}^{out} = f_5^d(d_{p,dy,hr}) \quad \forall r \in R^{FP}, t \in T, t \neq 1 \tag{12}$$

$$\Pi_{r,tp,e}^{in} = f_4^c(pwx_{hr,dy}) \quad \forall r \in R^{UT}, tp \in TP, e \in E \tag{13}$$

$$\Pi_{r,td}^{out} = f_5^c(d_{p,dy,hr}) \quad \forall r \in R^{FP}, td \in TD \tag{14}$$

5. New Continuous-Time Formulation (CT)

The new continuous-time formulation for short-term scheduling of continuous multiproduct plants under variable utility availability costs/profiles and multiple intermediate due dates is given below. The main focus will be on issues related to time modeling and the resource balances over time. The model entities are summarized in the Nomenclature section with all continuous variables being nonnegative.

5.1. Time Representation

The proposed continuous-time formulation uses a single time grid to keep track of events taking place, see Figure 6. It uses $|T|$ event points, also named global time points[15], that can be placed anywhere between the origin and end (H) of the time horizon. Traditionally[3,8-11], tasks starting at event point t have been assumed to start at the absolute time determined for that event point, T_t. For batch tasks with duration shorter than T_{t+1}-T_t, it is assumed that the output resources stay in the corresponding equipment unit for the remaining time. With continuous tasks, it is assumed that they last exactly the duration of the time interval while being processed at a rate lower than their predefined maximum rate. In reality, however, the tasks have total freedom to start anywhere provided that they end before T_{t+1}. Giménez et al.[16] highlight that the number of event points needed to represent a solution can be reduced if tasks are not required to start (end) exactly at a time point, while presenting a new continuous-time formulation for the short-term scheduling of multipurpose batch plants. The current case study is a perfect example for the advantages of such an approach as will be explained next.

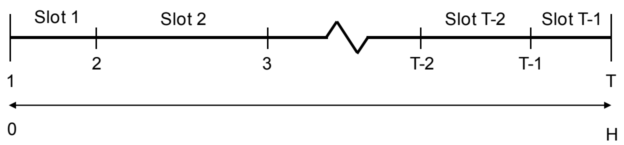

Figure 6. Continuous-time representation.

Consider a simple example involving the execution of a single processing task at slot/interval t, see Figure 7. The interval boundaries T_t and T_{t+1} can result from two consecutive demand points, in cases where the duration of the task is lower than the duration of the lowest cost period located inside the interval (shown in green). Variable Ts_t will give the starting time of the task executed during interval t (or the earliest starting time amongst all tasks executed). Note that no further event points are required.

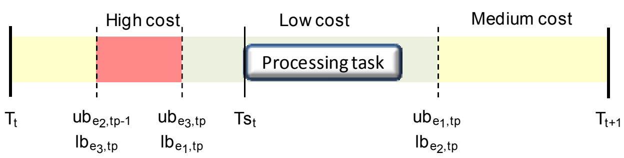

Figure 7. Continuous tasks executed at interval t do not necessarily start at event point t.

In the more general case, the amount to produce will require an extension larger than the duration of the cheapest time period. In such case, the processing task will be divided by the solver into as many times as the number of different energy cost levels, which may include consecutive time periods of the same energy level. Each such division will require an additional event point, as can be seen in Figure 8.

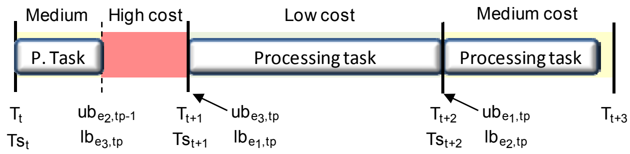

Figure 8. Multiple task instances might be required to meet the demands, one for every active energy cost level.

5.2. Timing Constraints

The fundamental timing constraint for a single time grid formulation[3] states that the difference in time between two consecutive event points must be greater than the duration of the task taking place. Equation (15) is written for all equipment resources involved in the timing constraints. The summation is used to reduce the integrality gap[3] and implicitly assumes that there can only be one task executed in such equipment unit at a certain time. This is ensured by the initial resource availability and the excess resource balances (see Eqs. (1) and (27)).

$$T_{t+1} - T_t \geq \sum_{i \in I^c} \frac{\bar{\mu}_{r,i} \xi_{i,t}}{\rho_i^{\max}} \quad \forall r \in R^{TC}, t \in T, t \neq |T| \tag{15}$$

A similar constraint can be used to guarantee that the tasks are fully executed within time interval *t*. Equation (16) together with Eq. (17) also satisfy Eq. (15). However, computational studies have shown that it is better to keep Eq. (15).

$$T_{t+1} - Ts_t \geq \sum_{i \in I^c} \frac{\bar{\mu}_{r,i} \xi_{i,t}}{\rho_i^{\max}} \quad \forall r \in R^{TC}, t \in T, t \neq |T| \tag{16}$$

$$Ts_t \geq T_t \quad \forall t \in T, t \neq |T| \tag{17}$$

We now define a new binary variable, $Y_{t,tp,e}$, which takes the value of one if, during interval *t*, tasks are processed within time period *tp* of energy level *e*. Similarly, we define binary variable $Y_{t,td}^{out}$, to identify whether or not event point *t* corresponds to demand point *td*. It can be assumed without loss of generality that there is: (i) one active time period of an energy level during t; (ii) one event point associated to due date td, eqs 18-19.

$$\sum_{e \in E} \sum_{tp \in TP_e} Y_{t,tp,e} = 1 \quad \forall t \in T, t \neq |T| \tag{18}$$

$$\sum_{t \in T \wedge t \neq 1} Y_{t,td}^{out} = 1 \quad \forall td \in TD \tag{19}$$

It time interval t is located within time period tp of energy level e, then the starting time of tasks must be greater than the time period lower bound, Eq. (20). Likewise, they must end before its upper boundary (see Figure 8). Equation (21) is a big-M constraint that is only active if there is a continuous task being executed that belongs to energy level e. This constraint is the one differentiating between energy levels for equivalent tasks (refer to Figure 4) so that the proper electricity cost is accounted for in the objective function, see Eq. (33).

$$Ts_t \geq \sum_{e \in E} \sum_{tp \in TP_e} lb_{e,tp} Y_{t,tp,e} \quad \forall t \in T, t \neq |T| \tag{20}$$

$$Ts_t + \sum_{i \in I_e^c} \frac{\overline{\mu}_{r,i} \xi_{i,t}}{\rho_i^{\max}} \leq \sum_{tp \in TP_e} ub_{e,tp} Y_{t,tp,e} + H \cdot (1 - \sum_{i \in I_e^c} \overline{\mu}_{r,i} N_{i,t}) \quad \forall r \in R^{TC}, t \in T, t \neq |T|, e \in E \tag{21}$$

If event point t corresponds to demand point td, then the time values must match, as expressed in Eqs. (22)-(23).

$$T_t \geq \sum_{td \in TD} tfx_{td} Y_{t,td}^{out} \quad \forall t \in T, t \neq 1 \tag{22}$$

$$T_t \leq \sum_{td \in TD} tfx_{td} Y_{t,td}^{out} + H \cdot (1 - \sum_{td \in TD} Y_{t,td}^{out}) \quad \forall t \in T, t \neq 1 \tag{23}$$

Finally, there are general bounds on the timing variables. According to Figure 6, $T_t \in [0,H]$, so Eqs. (24-25) apply. Equation (25) enforces that no task starts at the last event point. Since the model assumes that tasks can start after T_t, the time of the first event point can be set to zero without loss of generality (Eq. 26).

$$T_t \leq H \quad \forall t \in T \tag{24}$$

$$Ts_t \leq H \quad \forall t \in T, t \neq |T| \tag{25}$$

$$T_1 = 0 \tag{26}$$

5.3. Excess Resource Balances

Resource availability over the time grid is managed by the excess resource balances. These are multiperiod material balance expressions, in which the excess amount at event point t is equal to that at the previous event point (t-1) adjusted by the amounts discretely or continuously produced/consumed by all tasks starting or ending at t. Figure 9 illustrates the contributions to the values of the variables and supports the explanation of Eqs. (27-28).

In Eq. (27), the initial resource availability (first term on the RHS) is only to be considered at the first event point, t=1. From Eqs. (1-2), we know that it will be different from zero only for a subset of the resources, 1 for R^{EQ}, and a value to be determined by the solver, R_r^0, for $r \in R^{RM}$. For the remaining points, the value from the previous event point is used instead. For continuous resources R^{CT}, the ones involved in Eq. (28), $R_{r,t}$ receives the contribution of $R_{r,t-1}^{end}$, which gives the excess amount immediately before the end of interval t-1 (event point t). For the other resources but the utilities, one should use the value at the beginning of the previous interval, $R_{r,t-1}$. This third term on the RHS is not used for R^{UT}, since we want to prevent resource availability to propagate from one event point to the next. This is because there will be inputs from the exterior at every event point (sixth term on the RHS).

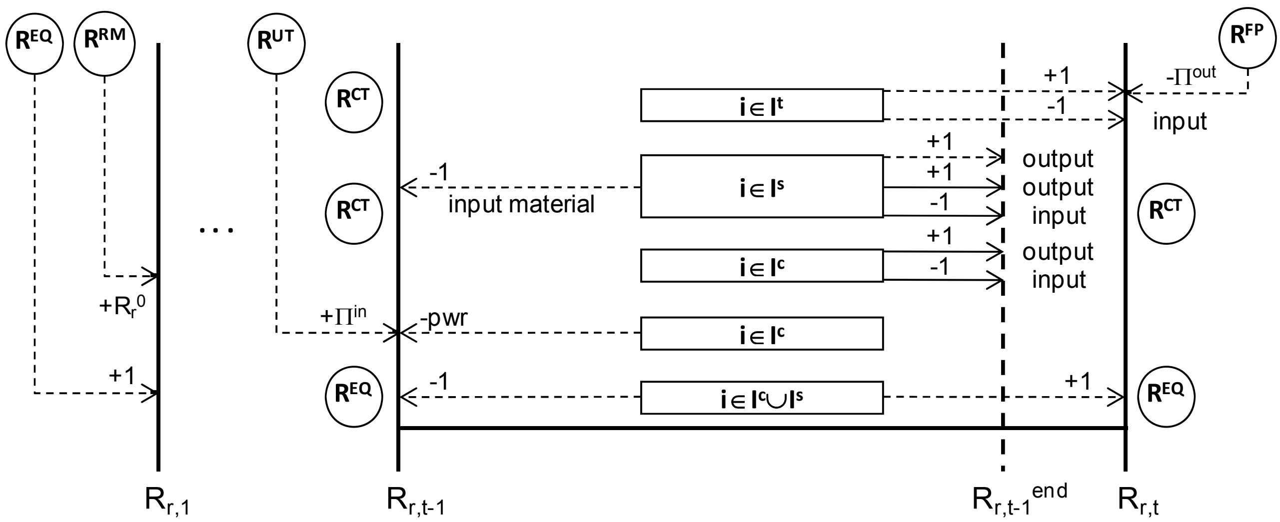

Figure 9. Events affecting the value of the excess resource variables.

$$R_{r,t} = R_r^0\big|_{t=1} + R_{r,t-1}^{end}\big|_{r\in R^{CT}} + R_{r,t-1}\big|_{r\notin(R^{CT}\cup R^{UT})} +$$
$$\sum_{i\in I}(\mu_{r,i}N_{i,t}\big|_{t\neq|T|} + \nu_{r,i}\xi_{i,t} + \bar{\mu}_{r,i}N_{i,t-1}) + \sum_{i\in I^t}\bar{\nu}_{r,i}\xi_{i,t} + \tag{27}$$
$$(\sum_{e\in E}\sum_{tp\in TP_e}\Pi_{r,tp,e}^{in}Y_{t,tp,e})\Big|_{r\in R^{UT}\wedge t\neq|T|} - (\sum_{td\in TD}\Pi_{r,td}^{out}Y_{t,td}^{out})\Big|_{r\in R^{FP}\wedge t\neq 1} \quad \forall r\in R, t\in T$$

$$R_{r,t}^{end} = R_{r,t} + \sum_{i\in I^c}\lambda_{r,i}\xi_{i,t} + \sum_{i\in I^s}(\bar{\nu}_{r,i}\xi_{i,t} + \lambda_{r,i}\xi_{i,t}^*) \quad \forall r\in R^{CT}, t\in T, t\neq|T| \tag{28}$$

In Eq. (27), the fourth and fifth terms deal with most of the discrete interactions of tasks with resources. The excess amount at event point t, decreases by tasks starting at t, and increases by tasks starting at t-1 (ending at t) or t (if they are instantaneous, $i\in I^t$). The discrete interaction at the end of storage tasks ($i\in I^s$) affects excess variable $R_{r,t}^{end}$ rather than $R_{r,t+1}$, see last term on the RHS of Eq. (28). The purpose is to get the amount in storage for every u so that the maximum capacity constraint can be enforced. Variable $R_{r,t}^{end}$ also deals with the continuous interactions of continuous (I^c) and storage tasks. Finally, the last term in Eq. (27) removes product resources from the system whenever the time of event point t is equal to a demand point td.

The excess resource variables should lie within given upper and lower bounds. The general constraints are given in Eqs. (29-30), while the values for this case study are discussed in section 4.3 and given in Eqs. (8-10).

$$R_r^{\min} \le R_{r,t} \le R_r^{\max} \quad \forall r\in R, t\in T \tag{29}$$

$$R_r^{end\min} \le R_{r,t}^{end} \le R_r^{end\max} \quad \forall r\in R, t\in T, t\neq|T| \tag{30}$$

5.4. Other Constraints

The remaining set of constraints ensures that the values of the continuous extent variables are set to zero whenever the task is not executed, i.e. when the binary extent variable is equal to zero. The upper bound in Eq. (31), U, can be given by the product of the time horizon times the sum of the maximum processing rate on every machine, see Eq. (32).

$$\xi_{i,t} + \xi_{i,t}^*\big|_{i\in I^s} \le U\cdot N_{i,t} \quad \forall i\in I^c\cup I^s, t\in T, t\neq|T| \tag{31}$$

$$U = H\cdot\sum_{m\in M}\max_{p\in P}\rho_{p,m} \tag{32}$$

5.5. Objective Function

The mathematical formulation is completed with the objective function. Equation (33) maximizes the total electricity cost [k€]. This is given by the sum over all electricity levels *e*, time intervals *t* and tasks *i*, of the product of electricity cost c_e [€/kWh], power consumption [MW], and duration of the task [h].

$$\min \sum_{e \in E} \sum_{i \in I_e^c} \sum_{r \in R^{UT}} \sum_{t \in T \wedge t \neq |T|} c_e \cdot (-\mu_{r,i}) \cdot \frac{\xi_{i,t}}{\rho_i^{\max}} \qquad (33)$$

6. Discrete-Time formulation (DT)

External interactions with the system at specific points in time, such as different electricity costs and due dates, are handled much more easily by a discrete-time formulation. This, of course, provided that the location of some of the grid's time points is within acceptable accuracy of the external events timing. After recalling a few important aspects related to the representation of time, the model constraints are given. These are analogous to the non-timing constraints of the continuous-time formulation, which is not surprising given the fact that the discrete-time formulation is in fact a sub-model of the more general continuous time formulation[18].

6.1. Time Representation

The single time grid employed by the discrete-time formulation has equal length time intervals of duration δ, see Figure 10. As a consequence, the absolute time of all event points is known a priori. Fixed length time intervals have important consequences for both batch and continuous tasks. For the former, the duration must often be rounded to a multiple of the interval length, leading to the consideration of an approximated version of the real problem. For continuous tasks, the extent can be made to vary within [0, $\delta \times \rho_i^{\max}$] to ensure that all possible demands can be met. However, there may be intervals where the task is being performed below its maximum processing rate, which in very constrained problems may compromise optimality or even feasibility since no other task can take advantage of the remaining capacity of the unit. This can be illustrated with a simple example.

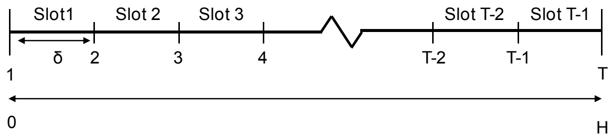

Figure 10. Discrete-time representation.

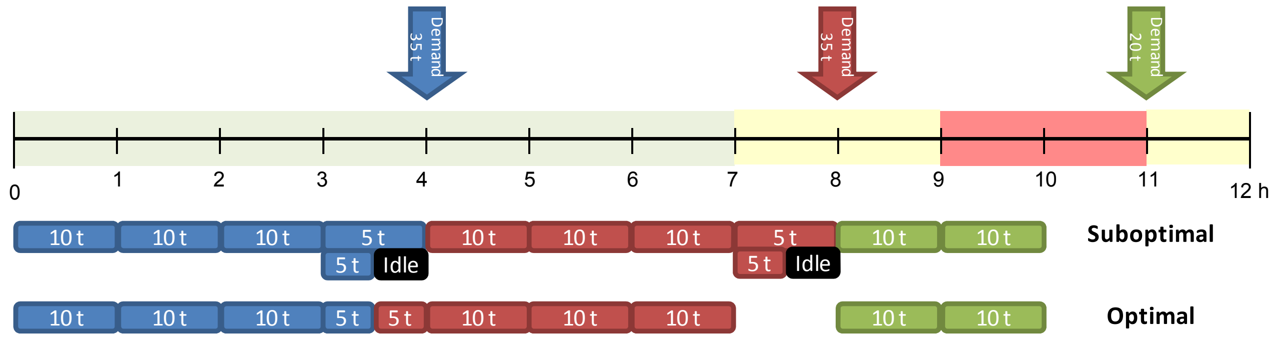

Figure 11. Limitation of discrete-time formulation when handling continuous tasks.

In Figure 11 it can be seen that for δ=1 h, we need to occupy the processing unit for the first eight hours of the time horizon to meet the blue and red products demands. The problem is that the last one is a medium-cost interval. Even though the objective function accounts only for the half an hour that the unit is active, optimality is compromised since the demands of both products can be met entirely in the low-cost period (first seven hours). Nevertheless, such solution cannot be obtained since the excess resource balances allow production of at most one product on each time interval. However, it can be achieved with δ=0.5 h (or with a continuous-time formulation), for which the intervals length will match exactly the time required to produce 5 t of product. Thus, the red product can start to be produced immediately after the blue ends, in order to finish before the medium-cost time period starts. More importantly, it would still be possible to product another 10 ton before the due date, in case of higher demands.

6.2. Model Constraints

The excess resource balances are given by Eqs. (28) and (34). The latter differs from Eq. (27) in the terms involving parameters Π^{in} and Π^{out}. Now their domain is exactly equal to that of the equation (recall the last paragraph of section 4.3) so there is no need to employ additional binary variables.

$$R_{r,t} = R_r^0\big|_{t=1} + R_{r,t-1}^{end}\big|_{r\in R^{CT}} + R_{r,t-1}\big|_{r\notin(R^{CT}\cup R^{UT})} + \\ \sum_{i\in I}(\mu_{r,i}N_{i,t}\big|_{t\neq|T|} + \nu_{r,i}\xi_{i,t} + \bar{\mu}_{r,i}N_{i,t-1}) + \sum_{i\in I^t}\bar{\nu}_{r,i}\xi_{i,t} + \\ \Pi_{r,t}^{in}\big|_{r\in R^{UT}\wedge t\neq|T|} - \Pi_{r,t}^{out}\big|_{r\in R^{FP}\wedge t\neq 1} \quad \forall r\in R, t\in T \tag{34}$$

Since there are no timing constraints, the capacity constraints need to ensure that the maximum amount of material handled by continuous tasks does not exceed the maximum processing rate times the duration of the time interval. For storage tasks, Eq. (35) is identical to Eq. (31).

$$\xi_{i,t} + \xi_{i,t}^*\big|_{i\in I^s} \le (U\big|_{i\in I^s} + \delta\cdot\rho_i^{\max}\big|_{i\in I^c})\cdot N_{i,t} \quad \forall i\in I^c\cup I^s, t\in T, t\neq|T| \tag{35}$$

6.3. Objective Function

The differences in comparison to the objective function of the continuous-time formulation, given in Eq. (33), result from the electricity cost parameters, which have a different domain. Because of this, Eq. (36) has one less summation.

$$\min \sum_{i\in I^c}\sum_{r\in R^{UT}}\sum_{t\in T\wedge t\neq|T|} ce_t\cdot(-\mu_{r,i})\cdot\frac{\xi_{i,t}}{\rho_i^{\max}} \tag{36}$$

7. Computational Studies

The performance of the models is illustrated through the solution of 10 test cases with data generated randomly from a real industrial problem. In all of them, the product demands lead to plant operation below their maximum capacity. Due to the differences in electricity cost among the energy levels, assigning production to the lower levels will have the biggest impact on the total cost. As we approach full plant capacity, all the energy levels become active, and the solver switches focus to finding the best combination of product-machine assignment according to the processing rates and power needs, which are product dependent. With the exception of EX5, the machines are occupied 41 to 56 % of the time. Directly related to the complexity of the problem is the number of products, machines and storage units, so the problems will be mainly characterized by (|P|, |M|, |S|).

The mathematical formulations give rise to mixed-integer linear programming (MILP) problems. They were implemented in GAMS 22.8 together with the algorithm that converts the problem data into the RTN format. All problems were solved by CPLEX 11.1 with default options, up to a relative optimality tolerance=10^{-6}, unless otherwise stated. The hardware consisted on a laptop with an Intel Core2 Duo T9300 processor running at 2.5 GHz, 4 GB of RAM and running Windows Vista Enterprise.

7.1. Illustrative Example

To illustrate the capabilities of the new continuous-time formulation we start with a very simple example involving two products, one machine and one storage unit (2,1,1). The machine power requirements is, in this case, product independent and equal to pw_{M1}= 5 [MW] and there are no constraints on maximum power consumption. The processing rates are $\rho_{P1,M1}$= 80, $\rho_{P2,M1}$=70 [ton/h], storage capacity is cap_{S1}=1200 [ton] and the product demands can be found in Table 1.

Table 1. Product Demands at the End of the Day for Examples EX1-EX3 [ton]

	MO (24 h)	TU (48 h)	WE (72 h)	TH (96 h)	FR (120 h)	SA (144 h)	SU (168 h)
P1	1000	-	500	-	-	1000	1200
P2	-	400	-	-	1200	-	-

The optimal solution for EX1 corresponds to a total electricity cost of €18,625. The resulting schedule and storage profiles are shown in Figure 12, where the numbers inside the boxes indicate the task length [h]. It requires a total of $|T|$=11 event points, which is the minimum number that ensures feasibility. No improvement in cost is observed for $|T|$=12. Such behavior is not common in continuous-time formulations[3,4,11], where typically the objective function increases with $|T|$ at least a couple of times. Therefore, it is an indication that the problem is highly constrained, which is a direct consequence of: (i) large variability of the electricity cost, leading to a total of 35 time periods with constant price; (ii) the tasks not being allowed to cross time periods. Thus, some demands can only be met by executing multiple instances of a particular task. More specifically, the Monday (MO) and Saturday (SA) demands of P1 need two tasks, while the Friday (FR) demand of P2 requires three.

In Figure 12, 8 out of 10 instances are executed in low-cost periods (green region) with the remaining being in medium-cost levels (yellow region). Ideally, all should be executed in the green region but that would only be possible for operation well below the maximum capacity of the plant. Power shortages may force tasks to switch to a higher cost level. In order to show that the model can cope with constraints of this type, EX2 was solved. It is uses the same data of EX1 but now the maximum power consumption during the first seven hours of Tuesday and Thursday ($pwx_{0\text{-}6,TU}$; $pwx_{0\text{-}6,TH}$) is set to zero.

The optimal solution for EX2 is shown in Figure 13 and corresponds to a total cost of €21,575. When compared to EX1 it translates into a 15.8% increase in cost, which is very significant considering such small changes in data. Such a schedule is obviously a feasible solution to EX1, which is in many aspects similar to the previous one. Hence, it also serves to illustrate the impact that advanced scheduling tools, like the models proposed in this paper, can have on plant profitability. We will return to this discussion in section 7.3.

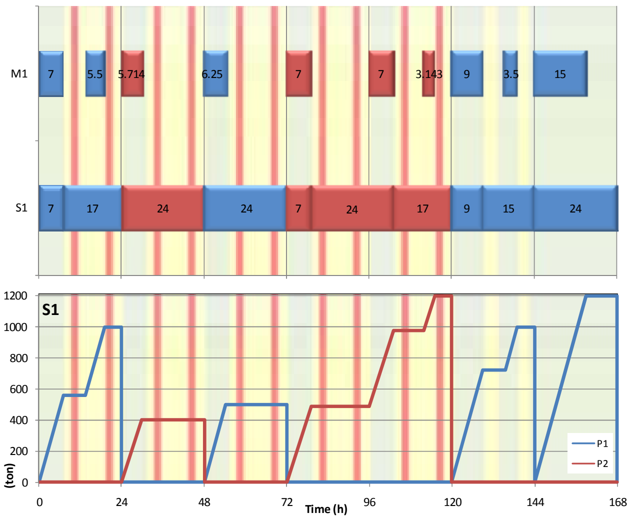

Figure 12. Optimal solution for EX1.

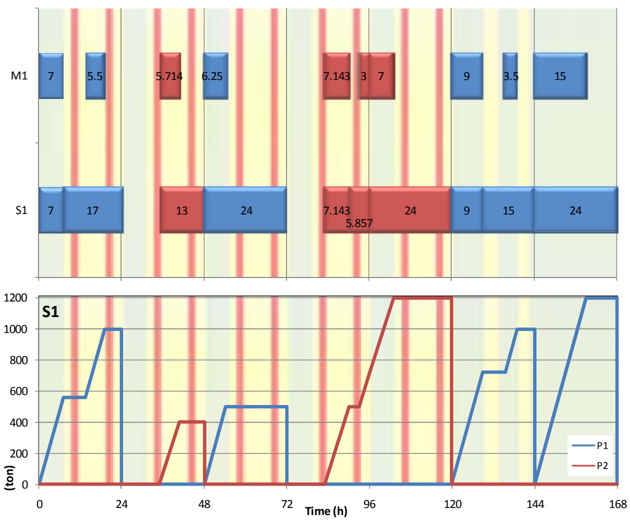

Figure 13. Optimal solution for EX2.

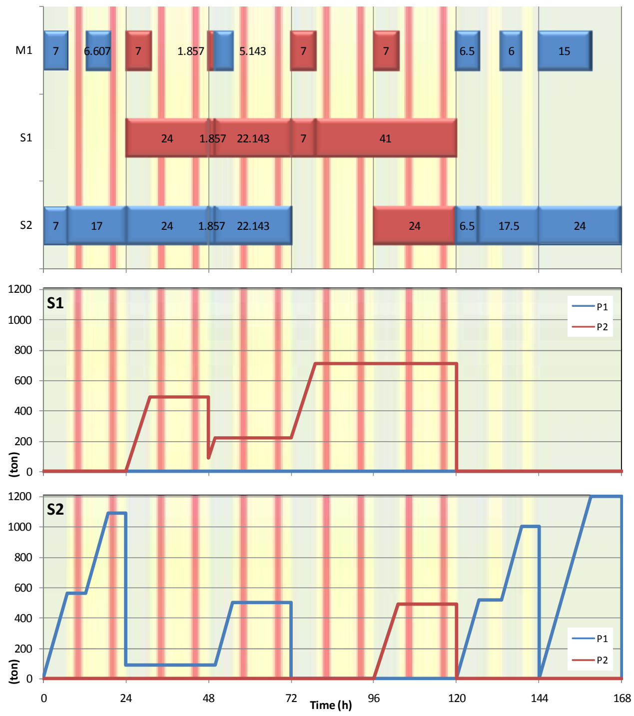

Figure 14. Optimal solution for EX3.

The amount of available storage is often limiting. The previous examples featured a shared storage unit that could only handle one product at a time. This made it impossible to meet demands of different products in the same day, which explains the format of such data. More importantly, production of *p* could only start after the total amount produced of *p*’ had been dispatched. In order to allow for more flexibility, EX3 considers two identical storage units and no maximum power constraints.

With the additional storage unit, the total cost can be reduced to €18,153 (2.5%). The optimal solution is given in Figure 14. It basically moves all the production of P2 (in red) to low-cost periods while more P1 (blue) gets directed to medium-cost periods. While one might be tempted to explain this behavior based on the different processing rates, there exists a degenerate solution that features P2 in both low- and medium-cost periods. On Monday, the production of P1 has increased from 1000 to 1089 ton, so silo S2 will remain partially filled until the end of Wednesday. S2 also holds P2 on Friday even though it could all be allocated to S1.

7.2. Computational Performance

The computational results of the 10 examples are given in Table 2. The example problems can be divided into two levels of complexity. EX1-5 can be addressed by the continuous-time model (CT), while for the others one has to rely on the discrete-time formulation (DT). This is the first major result: the continuous-time formulation is limited to small problems. Even for EX5, which features three products, two machines and two storage units, we had to specify low product demands (leading to a 33% capacity) to ensure that most tasks could fit into a single period of constant electricity cost. In this way, a good solution could still be found with a relatively small number of event points. However, it took already more than 1 hour to prove optimality for $|T|=11$ (€26,911), see Table 2, a value that is still 0.5% above the one from DT. For larger problems, CT is intractable due to the following facts: (i) the computational effort is strongly dependent on |T|, with experience telling us that we get typically a one order of magnitude increase for a single increase in |T|; (ii) it is not straightforward to find a number that ensures feasibility; (iii) it may be even difficult for the solver to find out that the value of $|T|$ is insufficient, i.e. finding that the problem is infeasible.

The last examples have more or less the complexity one expects to find in a real situation. The encouraging result is that the discrete-time formulation with 1-hour intervals, can find very good solutions to the problem in short computational time. In fact, from the results in Table 2, one can see that within up to 5 minutes, the optimal/best solutions returned are within a relative optimality gap of 0.78 %. Thus, DT provides an efficient decision-making tool for this problem. The gaps could be reduced most of the times by up to one hour of computational time, essentially due to the finding of better solutions since the value of the relaxed problem remained virtually unchanged. This is an indication of a high degree of degeneracy (more on this in section 7.4).

The results from Table 2 also show that the discrete-time formulation is significantly tighter than the continuous-time one. In EX1-4 and EX7 the integrality gap is even zero, and with the exception of EX8 it did not exceed 0.2%. The substantially large integrality gap for the latter (3.3%) is probably due to the use of harsher constraints on maximum power availability that prevented the machines to operate simultaneously in a significant part of the week. Nevertheless, CT still exhibited a reasonable integrality gap (maximum=23% for EX2, minimum=4.8% for EX5, average=11.6%).

Finally, it was surprising to find out that the number of binary variables of DT was of the same order of magnitude as that of CT in spite of the large difference between the $|T|$ numbers used in the two time grids. Recall that $t \in T$ is one of the indices of the binary extent variables $N_{i,t}$. There are two reasons for this: (a) there are three times as many tasks of the continuous type ($i \in I^c$) for CT due to the energy levels disaggregation, see section 3.2.1; but more importantly (b) CT employs two additional sets of binary variables, $Y_{t,td}^{out}$ and $Y_{t,tp,e}$, with the latter featuring three indices and being the major contributor whenever there is a large variability on electricity cost. Taking EX4 ($|T|$=12) as an example, such variables are responsible for 69% of the total number.

Table 2. Computational Results

	(|P|, |M|, |S|)	Model	$\|T\|$	binary variables	single variables	constraints	RMIP (€)	MIP (€)	CPU s	nodes
EX1	(2,1,1)	DT	169	672	4552	3203	18625	18625	0.32	0
		CT	11	490	797	347	16620	18625	8.53	4252
		CT	12	539	875	380	16620	18625	9.8	5442
EX2	(2,1,1)	DT	169	672	4552	3203	21575	21575	0.29	0
		CT	11	490	797	347	16620	21575	26	20971
		CT	12	539	875	380	16620	21575	21.6	15474
EX3	(2,1,2)	DT	169	1008	6743	4382	18153	18153	0.40	0
		CT	11	510	934	420	16620	18153	237	69046
		CT	12	561	1025	460	16620	18153	1372	467962
EX4	(2,1,2)	DT	169	1008	6743	4382	21175	21175	0.52	10
		CT	11	510	934	420	18896	21349	178	46512
		CT	12	561	1025	460	18896	21175	1470	237314
EX5	(3,2,2)	DT	169	2016	10784	6739	26738	26780	3600[a]	133242
		CT	9	528	1089	562	25625	27222	7.18	3989
		CT	10	594	1221	629	25625	27008	369	138426
		CT	11	660	1353	696	25625	26911	4131	1295540
EX6	(3,2,3)	DT	169	2520	13986	8423	43250	43259	3600[b]	85900
EX7	(3,3,4)	DT	169	3528	18365	10780	68282	68282	18.4	671
EX8	(3,3,5)	DT	169	4032	21567	12464	101139	104622	3600[c]	41252
EX9	(4,3,4)	DT	169	4704	23923	13810	87817	87868	3600[d]	24400
EX10	(5,3,4)	DT	169	5880	29481	16840	86505	86581	3600[e]	7500

[a]Integrality gap at 300s (IG300) [%]=0.06; Integrality gap at time of termination (IGT) [%]=0.04; [b]IG300=IGT=0.02; [c]IG300=0.61, IGT=0.18; [d]IG300=IGT=0.06; [e]IG300=0.78, IGT=0.09.

7.3. Potential Cost Savings

One important question is what are the potential savings that one may get by using such a comprehensive optimization scheduling tool? In other words, how important is it to account for the real electricity cost profile when deriving the schedule? Given: (i) the many constraints in terms of storage availability, demand satisfaction and maximum power availability; (ii) the need to provide a schedule fast not only for normal operation but, more importantly, as a response to unpredicted events, such as machine breakdown, or urgent orders that need to be satisfied; It is fair to assume that a feasible solution to the problem provides a valid basis for comparison.

A so called “blind” schedule can easily be derived by the discrete-time formulation under the assumption of constant electricity price throughout the time horizon. Under such scenario, and according to Eq. (36), the model will simply try to produce the products in the faster machines. If in terms of processing rate we get the same product rank on every machine, then the schedule will tend to a series production rather than a parallel one, which as we can see in Figure 15, is the preferred choice under a variable cost profile. Nevertheless, the intermediate product demands partially avoid this. Also, the effect becomes less important for higher plant capacities. Keeping this in mind, we stick to the plan. After obtaining the blind schedule, the real cost parameters are used to compute the true total cost.

Table 3. Potential Cost Savings as a Function of the Plant Capacity.

	EX1	EX2	EX3	EX4	EX5	EX6	EX7	EX8	EX9	EX10
Capacity (%)	41	41	41	47	33	46	40	55	51	56
Savings (%)	27	23	39	29	40	33	39	20	32	33

Table 3 lists the results obtained. Even after considering that an experienced scheduler by means of the heuristics in use at the plant could do a significantly better job than the “blind” schedule for the reasons mentioned above, these are very significant cost savings. The trend is that as the plant approaches full capacity, the potential savings diminish, which is not surprising given that there are fewer degrees of freedom. It is relevant to highlight the result for EX3 when compared to that for EX2. As discussed in section 7.1, they have the same data but EX3 has one more storage unit, which considerably increases the plant flexibility. This makes it much easier to generate a feasible schedule, but if one is not careful, one may make wrong decisions concerning the impact on cost (increase in potential savings from 23 to 39%).

Figure 15. Best solution found by continuous-time formulation for EX5.

7.4. Final Remarks

To end the analysis, we discuss the issue of solution degeneracy and its impact on the form of the schedule. For a schedule to be implemented in practice it should have as few events as possible since there are always factors that are not considered in the model. In this case study, the issue of sequence dependent changeovers is not relevant. However, that does not mean that we are allowed to change products on a given machine every hour! Therefore, we wish to first complete the production of a product and only then switch to another one. The reality is thus much closer to the concept of events used by time grid based continuous-time formulations, or of precedence used by sequencing based models[15]. The iterative procedure used by CT of increasing the number of event points one by one, implicitly ensures simplicity. In contrast, in DT, a much larger number of events are allowed, so there will typically be many more task instances being executed to achieve a particular amount of product. Thus, there will be a large number of solutions that correspond to the same value of the objective function, with the large majority of them being undesirable due to many changeovers, see

Figure 16. To overcome such solutions a post-processing procedure is required that removes the superfluous but this is beyond the scope of this paper.

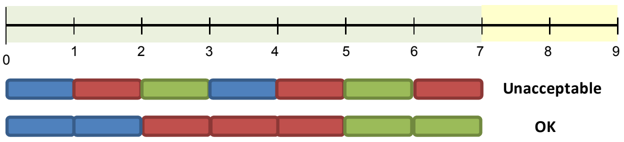

Figure 16. High solution degeneracy may lead to an unacceptable number of changeovers in solutions from the discrete-time formulation.

8. Conclusions

This paper has focused on the modeling of discrete events that occur at predetermined points in time with a continuous-time scheduling formulation. These included multiple intermediate due dates, utility availability and variable electricity costs. A conceptually new general model has been proposed that relies on the Resource-Task Network for process representation. It is nevertheless limited to continuous tasks. The validity of the approach has been demonstrated on a few test cases adapted from a real industrial problem. In such process, storage units are used for final products and may act as shared storage units. To efficiently model this issue, a novel hybrid batch/continuous task has been proposed.

Despite the major modeling breakthrough, the results have shown that only problems of small size can be handled effectively. This turned our attention to discrete-time models, where discrete events can be handled in a more natural and straightforward way. The only doubt was to whether a sufficiently fine time grid could be used to represent the problem data accurately. The goal was to ensure a weekly schedule that accounted for hourly changes in electricity cost, and this has been successfully accomplished. Problems of industrial significance were tackled down to optimality gaps below 1% with practical computational times of 5 minutes. This is mostly due to the characteristically low integrality gap of RTN discrete-time formulations, which were found to be significantly lower than their continuous-time counterparts.

The last part of the paper has highlighted the importance of taking variable electricity costs into consideration when deriving the schedule. State-of-the-art scheduling formulations have the potential to achieve major savings when compared to procedures that are mostly focused on feasibility. While accurate values are obviously dependent on problem data, particularly on the different cost levels agreed with the electricity provider, and on the scheduling practice at the plant, results have shown potential cost savings around 20%. Clearly, values of this order of magnitude provide enough motivation for the incorporation of the discrete-time mathematical formulation presented in this paper as a central element of the decision making tool used by industry.

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 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
TD/td=demand points
TP_e/tp=time periods of electricity 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]
$d_{p,dy,hr}$=demand of product p at the end of hour hr of day dy [ton]
H= time horizon [h]
$lb_{e,tp}$=starting time of time period tp of electricity level e [h]

$pw_{p,m}$=power requirement for product p in machine m [MW]
$pwx_{hr,dy}$=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
tfx_{td}=absolute time of demand point td [h]
$ub_{e,\mathrm{tp}}$=ending time of time period tp of electricity level e [h]
δ=duration of every interval on discrete-time grid [h]
$\lambda_{\mathrm{r,i}}$=continuous interaction of resource r during execution of task i
$\mu_{\mathrm{r,i}}$=discrete interaction of resource r with task i at its start, acting on binary extent variables
$\overline{\mu}_{r,i}$ =discrete interaction of resource r with task i at its end, acting on binary extent variables
$\nu_{\mathrm{r,i}}$= discrete interaction of resource r with task i at its start, acting on continuous extent variables
$\overline{\nu}_{r,i}$ = discrete interaction of resource r with task i at its end, acting on continuous extent variables
$\Pi_{r,t}^{in}$ =amount received into the system of resource r at event point t
$\Pi_{r,t}^{out}$ =amount removed from the system of resource r at event point t
$\rho_i^{\max}$ =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
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
$Y_{t,td}^{out}$ =binary variable identifying if event point t corresponds to demand point td
$\xi_{\mathrm{i,t}}$=amount handled by task i at event point/during interval t (continuous extent variables) [ton]
$\xi_{i,t}^*$ =amount continuously sent to storage by task i during interval t [ton]

References

(1) Castagnoli, D.; Gallestey, E. Technical Report CH-RD 2003-12. ABB Corporate Research, Dättwil, Switzerland, 2003.

(2) Maravelias, C.; Grossmann, I. 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), 2003, pp. 215.

(3) Castro, P.; Barbosa-Póvoa, A.; Matos, H.; Novais, A. Simple Continuous-time Formulation for Short-Term Scheduling of Batch and Continuous Processes. *Ind. Eng. Chem. Res.* **2004**, *43*, 105.

(4) Shaik, M.; Janak, S.; Floudas, C. Continuous-Time Models for Short-Term Scheduling of Multipurpose Batch Plants: A Comparative Study. *Ind. Eng. Chem. Res.* **2006**, *45*, 6190.

(5) Schilling, G.; Pantelides, C. Optimal Periodic Scheduling of Multipurpose Plants. *Comput. Chem. Eng.* **1999**, *23*, 635.

(6) Pantelides, C.C. 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, 1994; pp 253.

(7) Duarte, B.; Santos, L.; Mariano, J. Optimal sizing, scheduling and shift policy of the grinding section of a ceramic tile plant. *Computers Operations Research*. **2009**, *36*, 1825-1834.

(8) Castro, P.; Barbosa-Póvoa, A.; Novais, A. Simultaneous Design and Scheduling of Multipurpose Plants Using Resource Task Network Based Continuous-Time Formulations. *Ind. Eng. Chem. Res*. **2005**, *44*, 343.

(9) Maravelias, C.T.; Grossmann, I.E. New General Continuous-Time State-Task Network Formulation for Short-Term Scheduling of Multipurpose Batch Plants. *Ind. Eng. Chem. Res.* **2003**, *42*, 3056.

(10) Janak, S.L.; Lin, X.; Floudas, C.A. 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.

(11) Shaik, M.; Floudas, C. Novel Unified Modeling Approach for Short-Term Scheduling. Submitted to *Ind. Eng. Chem. Res*.

(12) Castro, P.; Barbosa-Póvoa, A.; Matos, H. Periodic Scheduling of an Industrial Batch System using a RTN Continuous-time Formulation. In *Chemical Engineering Transactions, Vol 3*. Editor: Sauro Pierucci, AIDIC, Italy, 2003; pp 1105.

(13) Kondili, E.; Pantelides, C. C.; Sargent, R. A General Algorithm for Short-Term Scheduling of Batch Operations - I. MILP Formulation. *Comput. Chem. Eng.* **1993**, *17*, 211.

(14) Castro, P.; Barbosa-Póvoa, A.; Matos, H. 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.

(15) Méndez, C.A.; Cerdá, J.; Grossmann, I.E.; 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.

(16) Giménez, D.; Henning, G.; Maravelias, C. A Novel Network-based Continuous-Time Representation for Process Scheduling: Part I. Mathematical Formulation. Submitted to *Comput. Chem. Eng*.

(17) Castro, P.; Westerlund, J.; Forssell, S. Scheduling of a Continuous Plant with Recycling of Byproducts: a Case Study from a Tissue Paper Mill. *Comput. Chem. Eng*, doi:10.1016/j.compchemeng.2008.10.004

(18) Maravelias, C.; Grossmann, I.E. On the Relation of Continuous- and Discrete-Time State-Task Network Formulations. *AIChE J.* **2006**, *52*, 843.

(19) Allen, E.; Ilic, M. *Price-Based Commitment Decisions in the Electricity Market*. Springer-Verlag: London, 1999.