

Rolling-Horizon Algorithm for Scheduling under Time-Dependent Utility Pricing and Availability

Pedro M. Castro,^a Iiro Harjunkoski,^b Ignacio E. Grossmann^c

^a*Energy Systems Modeling and Optimization Unit, LNEG, 1649-038 Lisboa, Portugal*

^b*ABB Corporate Research Center, Wallstadter Str. 59, 68526 Ladenburg, Germany*

^c*Dept. Chemical Engineering, Carnegie Mellon University, Pittsburgh 15213, USA*

Abstract

This work addresses the scheduling of continuous single-stage multiproduct plants with energy intensive processing tasks and time-dependent electricity cost and power supply. A new rolling horizon algorithm is proposed that consists of a planning model to predict the production levels and a continuous-time model for detailed scheduling. The results from a set of test problems from the literature show that the algorithm can generate global optimal solutions much more rapidly than standalone discrete or continuous-time formulations in problems involving unlimited power availability.

Keywords: Continuous-time, Resource-Task Network, Planning, Electricity

1. Introduction

Modern enterprises are complex global networks consisting of multiple business units and functions. In order to remain competitive in the global marketplace, companies need to optimize the various functions that comprise the supply chain. One of the focuses of Enterprise-Wide Optimization is the operation of manufacturing facilities, where essential operational items include planning and scheduling (Grossmann, 2005). A major challenge is the discovery of what type of models to use to render the effective solution of real-life problems.

This paper builds on our recent work (Castro et al. 2009), which has proposed a new continuous-time short-term scheduling formulation for continuous plants subject to energy constraints related to electricity pricing and availability. Despite the major breakthrough of effectively handling discrete events resulting from time-dependent utility cost/availability profiles and multiple intermediate due dates, the number of event points needed to represent a solution rapidly increases with the number of energy levels and demand points. As a consequence, the mixed-integer linear programming (MILP) formulation can only solve very small problems to optimality, when considering a one-week horizon with end-of-the-day demands and frequent energy changes.

Discrete events are handled more naturally with a discrete-time formulation but there are two important issues when addressing problems featuring continuous tasks. First, slightly suboptimal solutions may result since the task duration is approximated by a multiple of the prespecified interval length. Second, multiple instances will typically need to be executed to meet the daily demands, leading to high solution degeneracy. Thus, only by chance will one get solutions with a minimum number of changeovers that can actually be implemented in practice.

Clearly, discrete and continuous-time formulations have complementary strengths and the ideal approach should combine the advantages of both (Maravelias, 2005; Westerlund et al., 2007). In this paper, we propose a new rolling-horizon algorithm that

can tackle the full problem in a sequence of iterations. The scheduling horizon is divided into detailed and aggregate time blocks. The aggregate model comprises a non-uniform discrete-time model to predict the production levels without determining the timing of events. Detailed scheduling is achieved with the continuous-time model for a time window between consecutive demand points, while simultaneously considering the remaining horizon with the aggregate model.

2. Problem Definition

Consider the last processing stage of a continuous multiproduct plant. The intermediate material is transformed into one of a few possible products (P) in equipment units (M) through the use of electricity. These are then sent to storage units (S) where they wait for dispatch, see Fig. 1. Units are characterized by power requirements $pw_{p,m}$ [MW] and processing rates $\rho_{p,m}$ [ton/h]. Products may have multiple demands, occurring at any hour of the day, $d_{p,hr,dy}$ [ton]. The maximum capacity of storage units, which are shared, is cap_s [ton], while $im_{s,p}$ [ton] are the initial amounts in storage.

The objective will be to minimize total energy cost, for a given energy contract between the plant and electricity provider that specifies time-dependent electricity cost, $ec_{hr,dy}$ [€/kWh] and maximum power levels, $pw_{hr,dy}$ [MW].

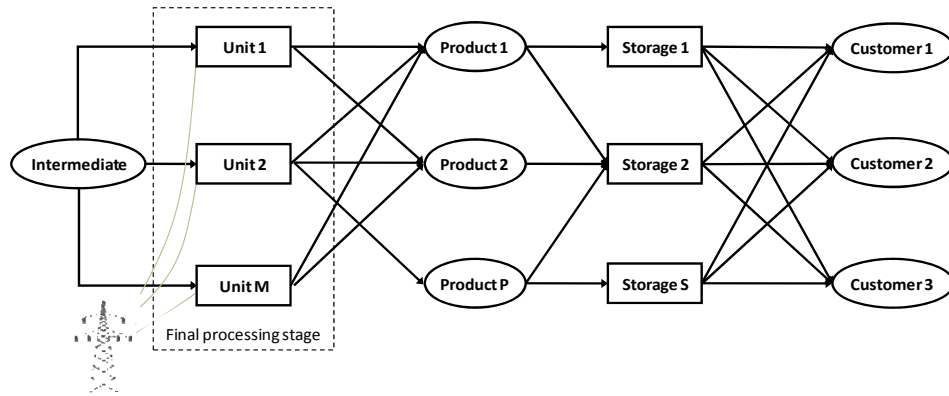


Figure 1. Generic representation of industrial case study

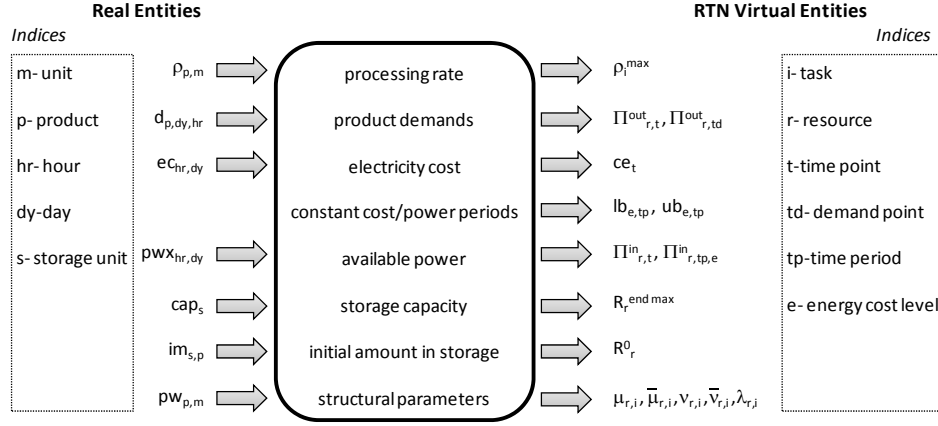


Figure 2. Process entities are converted into RTN model entities by a special purpose algorithm

3. From Process to Resource-Task Network Entities

The industrial process under study can be described as a Resource-Task Network (Castro et al., 2009). We can automatically generate the sets of tasks and resources as a function of problem data. Tasks can either be continuous (I^c), hybrid batch-continuous (I^h) or instantaneous (I^i). Resources include equipment units (R^{EQ}), divided into storage and processing units (R^{TC}), electricity (R^{UT}) and raw-material (R^{RM}). In addition, we distinguish the product location: immediately after processing (R^{LM}); in storage (R^{LS}); inside the clients' transportation vessels (R^{FP}). Resources continuously produced are given by $R^{CT} = R^{RM} \cup R^{LM} \cup R^{LS}$, whereas the final product (R^{FP}) is produced/consumed instantaneously. The RTN representation is then brought into the mathematical model by the structural parameters, see Fig. 2.

4. Mathematical Formulation

We propose a new combined aggregate/continuous-time model as the center of the rolling-horizon algorithm for scheduling. It uses a 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 element of $T^{c, last}$, a boundary time point also part of the set of discrete time points. The discrete part of the grid is non-uniform, meaning that the duration Δ_t is not the same for all time slots t . The number of slots and their exact location in the grid is calculated from the product demand and electricity cost and availability data.

Three sets of variables are used to characterize tasks. Binary variables $N_{i,t}$ identify the execution of task i during slot t ; continuous variables $\zeta_{i,t}$ give the amount processed; the amount continuously sent to storage by hybrid tasks is given by $\zeta_{i,t}^*$. We keep track of resource availability over time through excess resource variables $R_{r,t}$ and $R_{r,t}^{end}$. Slack variables $S_{r,t}$, representing the extra power to be purchased [MW], allow the violation of utility balances but are penalized in the objective function through parameter cs .

$$\min pcost + \sum_{r \in R^{UT}} \sum_{e \in E} \sum_{t \in T^c \wedge t \notin T^{c,last}} \sum_{i \in I_\xi^c} c_e \cdot (-\mu_{r,i}) \cdot \frac{\xi_{i,t}}{\rho_i^{\max}} \quad (1)$$

$$+ \sum_{r \in R^{UT}} \sum_{t \in T^{ag} \wedge t \notin T^{c,last}} \sum_{i \in I_\xi^c} c_{el} \cdot (-\mu_{r,i}) \cdot \frac{\xi_{i,t}}{\rho_i^{\max}} + cs \sum_{t \in T} S_{r,t}$$

$$R_{r,t} = R_r^0 \Big|_{t=1} + R_{r,t-1}^{end} \Big|_{r \in R^{CT}} + R_{r,t-1} \Big|_{r \notin (R^{CT} \cup R^{UT})} - (\Pi_{r,t}^{out} + \Pi_{r,td^*}^{out} \Big|_{t \in T^{c,last}}) \Big|_{r \in R^{FP}} +$$

$$(\Pi_{r,t}^{in} + \sum_{e \in E} \sum_{tp \in TP_e} \Pi_{r,tp,e}^{in} Y_{t,tp,e} \Big|_{t \in T^c \setminus T^{c,last}} + S_{r,t}) \Big|_{r \in R^{UT}} + \quad (2)$$

$$\sum_{i \in I} (\mu_{r,i} N_{i,t} \Big|_{i \in I_\xi^c \vee (i \notin I^c \wedge t \neq |T|)} + v_{r,i} \xi_{i,t} \Big|_{i \in I_\xi^c \vee i \notin I^c} + \bar{\mu}_{r,i} N_{i,t-1} \Big|_{i \in I_{t-1}^c \vee i \notin I^c}) + \sum_{i \in I^t} \bar{v}_{r,i} \xi_{i,t}$$

$$\forall t \in T, r \in R, [r \notin (R^{TC} \cup R^{UT}) \vee t \in T^c \setminus T^{c,last}]$$

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

$$Ts_t \geq T_t \Big|_{t \notin T^{first}} + tfx_{td^*-1} \Big|_{t \in T^{first}} \quad \forall t \in T^c \setminus T^{c,last} \quad (4)$$

$$tfx_{td^*} \Big|_{t+1 \in T^{c,last}} + T_{t+1} \Big|_{t+1 \notin T^{c,last}} - Ts_t \geq \sum_{i \in I_\xi^c} \frac{\bar{\mu}_{r,i} \xi_{i,t}}{\rho_i^{\max}} \quad \forall r \in R^{TC}, t \in T^c \setminus T^{c,last} \quad (5)$$

$$Ts_t \geq \sum_{e \in E} \sum_{tp \in TP_e} lb_{e,tp} Y_{t,tp,e} \quad \forall t \in T^c \setminus T^{c,last} \quad (6)$$

$$Ts_t + \sum_{i \in I_\xi^c} \frac{\bar{\mu}_{r,i} \xi_{i,t}}{\rho_i^{\max}} \leq \sum_{tp \in TP_e} ub_{e,tp} Y_{t,tp,e} + tfx_{td^*} \cdot (1 - \sum_{i \in I_\xi^c} \bar{\mu}_{r,i} N_{i,t}) \quad \forall r \in R^{TC}, e, t \in T^c \setminus T^{c,last} \quad (7)$$

$$\sum_{e \in E} \sum_{tp \in TP_e} Y_{t,tp,e} = 1 \quad \forall t \in T^c \setminus T^{c,last} \quad (8)$$

$$\Delta_t \geq \sum_{i \in I_\xi^c} \frac{\bar{\mu}_{r,i} \xi_{i,t}}{\rho_i^{\max}} \quad \forall r \in R^{TC}, t \in T^{ag}, t \neq |T| \quad (9)$$

$$\Delta_t \cdot (\Pi_{r,t}^{in} + S_{r,t}) \geq \sum_{i \in I_\xi^c} \frac{-\mu_{r,i} \xi_{i,t}}{\rho_i^{\max}} \quad \forall r \in R^{UT}, t \in T^{ag}, t \neq |T| \quad (10)$$

$$\xi_{i,t} + \xi_{i,t}^* \Big|_{i \in I^s} \leq H \cdot \sum_{m \in M} \max_{p \in P} \rho_{p,m} \cdot N_{i,t} \quad \forall t \in T, i \in I_\xi^c \vee (i \in I^s \wedge t \neq |T|) \quad (11)$$

In Eq. 1, $pcost$ is a parameter that accounts for the partial electricity cost [k€] in previously scheduled periods, the second term gives the electricity cost from processing tasks executed in the continuous part of the grid (sum over all electricity levels e of the product of electricity cost c_e [€/kWh], power consumption [MW] and task duration [h]) and the third, the cost from the discrete part of the grid.

The complexity of the model is reflected by the excess resource balances (Eqs. 2-3). They are very similar to those of the pure continuous-time model (Castro et al., 2009). Parameter Π_{r,td^*}^{out} holds the product demands at the end of the period td currently being scheduled in detail (superscript *). In the grid, such interaction occurs at $T^{c,last}$. There may be multiple energy pricing levels within a demand period and even different time

periods tp within the same energy level (TP_e). To identify during which time period tasks executed at slot t are processed at, binary variables $Y_{t,tp,e}$ are used.

Eqs. 4-8 are for the continuous part of the model. 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 slot t . The starting time must be greater than the event point's absolute time, which for the first point is given by the location of the previous demand point (Eq. 4). Likewise, tasks must end before $t+1$ (Eq. 5). Furthermore, tasks executed during slot t must lie within a single time period of an energy level (Eq. 8). Hence, the starting time must also be greater than the time period's lower bound ($lb_{e,tp}$) and there must be enough time for the tasks to end before its upper bound, $ub_{e,tp}$ (Eqs. 6-7).

Note that the domain of Eq. 2 does not include processing units or utilities for the discrete part of the grid. The purpose is to allow multiple products to be processed on a given time slot. Resource balances for units are replaced by timing constraints that ensure that the total processing time in slot t does not exceed its duration (Eq. 9). For the utility, the instantaneous balances on power are replaced by energy balances (Eq. 10). Finally, the last set of constraints activate the binary variables whenever there is material being handled by the task (Eq. 11).

5. Rolling-Horizon Algorithm

The discrete part of the grid is handled by an aggregate (planning) model that predicts the production levels without being concerned with the timing of events. It is completely accurate for cases with unlimited power availability, generating a lower bound on cost otherwise. More importantly, the output solution can be analyzed to predict how many event points are required by the continuous-time formulation to generate the detailed schedule. Consider an aggregate time slot t resulting from merging 3 time periods of length 2, 8 and 3 h for some energy level e (the lighter boxes in Fig. 3, which correspond to a medium cost level, see Castro et al. 2009 for further details). If the maximum processing time over all units M is 11 h, we know that the production will spread at least 2 slots (8+3 h) in the continuous time formulation. Step 0 of the rolling-horizon algorithm will thus involve solving the full problem with the aggregate model.

Fig. 3 illustrates the main components of the algorithm for a simple example. In step 0, $T^c = \emptyset$ and $T^{ag} = \{1, \dots, 13\}$, while in step1, we use the same 13 events: $T^c = \{1, \dots, 4\}$ and $T^{ag} = \{4, \dots, 13\}$. Notice in the latter that the exact location of the energy levels is being considered in the first demand period but not all are active. Provided that a processing task occupies a single level (for low production rates), a slot can span across multiple levels (Castro et al., 2009). If the objective function value (OBJ) remains constant between iterations, we can fix the schedule and proceed to the next demand period.

In step 2, the aggregate model predicts 3 events for demand period 2: $T^c = \{4, \dots, 6\}$ and $T^{ag} = \{6, \dots, 12\}$. Now OBJ is greater than the lower bound (LB) so we need to resolve the problem following a single increase in $|T|$. If OBJ remains unchanged, the available power is insufficient to meet the production levels predicted by the aggregate model in the cheaper cost periods (see step 3). The lower bound can thus be reset, the added event point removed, and the algorithm can proceed to the next demand period until eventually tackling all of them and generating the full schedule (final step).

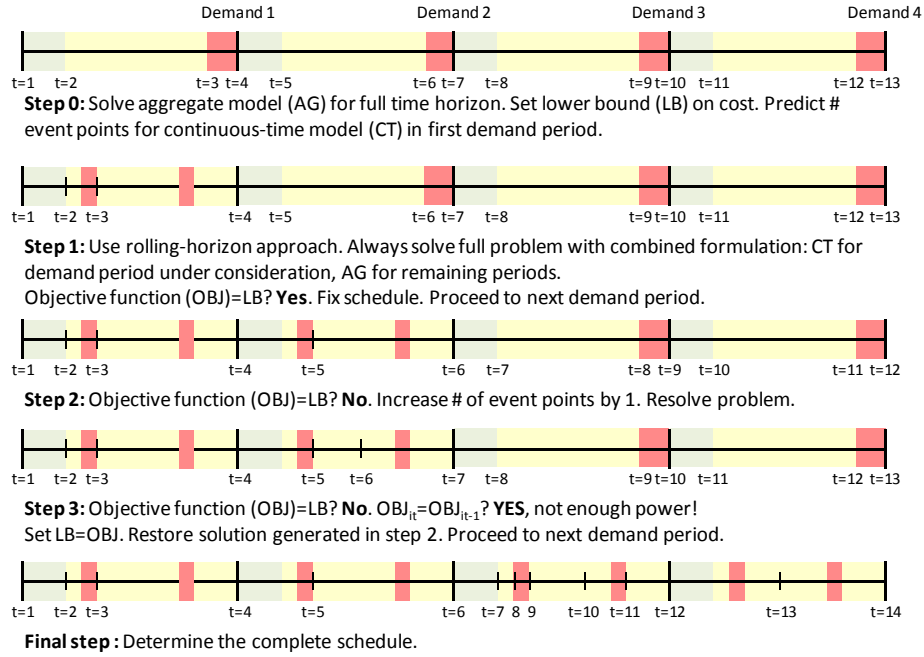


Figure 3. Illustration of rolling-horizon algorithm (boxes represent different energy cost levels)

6. Computational Results

The performance of the rolling-horizon (RH) algorithm and standalone aggregate model (AG) were evaluated through the solution of 7 example problems (Castro et al., 2009). We have used GAMS 22.8 (CPLEX 11.1) to solve the resulting MILP problems. The termination criteria were: relative tolerance= 10^{-6} ; maximum computational time=3,600 CPUs per iteration. 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. In order to prioritize schedules that meet the maximum power constraints, we set $cs=10$ k€MW. From the results in Table 1, it is clear that the aggregate model (AG) is very powerful. On the one hand, it can solve all problems in a few seconds, which represents orders of magnitude reduction in computational time when compared to a traditional discrete-time (DT) formulation employing 1-hour intervals (Castro et al., 2009). On the other hand, it gives very good total cost predictions, which are accurate for unrestricted power (U). However, the results for EX5a and EX8 are below those obtained by the rigorous DT model (29657 vs. 31798 and 14375 vs. 104622). In such problems, the RH algorithm generates solutions that violate the power constraints due to scheduling in detail a single demand period per iteration, rather than the full 1-week horizon. This is the reason for the large optimality gap. Besides EX8, RH could find solutions in less than half an hour, with the schedules for EX9-10 being slightly better than those obtained by DT in 2-hours time. Notice in the last column that we were able to go up to 30 event points, whereas with the full-space continuous-time approach the typical limit is roughly 10.

7. Conclusions

This paper has proposed a new rolling-horizon algorithm for the scheduling of single stage continuous multiproduct plants subject to energy constraints related to time dependent utility pricing and availability, and multiple due dates. The elements for decomposition are the demand periods, with the one period being tackled with a continuous-time scheduling model while simultaneously considering the remaining horizon with a non-uniform discrete-time planning model. The algorithm was shown to effectively solve real-life problems for the case of unrestricted power availability to global optimality. In the other cases, it may lead to suboptimal solutions due to the unnecessary violation of the power constraints, a consequence of the planning model underestimating the total electricity cost.

Table 1. Computational Statistics

Approach			AG	AG	AG	RH	RH	RH
Case	(P,M,S)	Power	T	CPUs	Total cost [€]		T	CPUs
EX2	(2,1,1)	R	17	0.12	21575		11	1.11
EX5	(3,2,2)	U	19	0.33	26758		13	2.26
EX5a	(3,2,2)	R	20	0.24	29657	41124	17	7.06
EX7	(3,3,4)	U	18	0.7	68282		12	3.12
EX8	(3,3,5)	R	19	2.05	104375	151257	31	17330
EX9	(4,3,4)	U	19	0.71	87817		25	917
EX10	(5,3,4)	U	19	3.57	86550		23	1508

References

- P. Castro, I. Harjunkoski, I. Grossmann, 2009. Ind. Eng. Chem. Res. 48, 6701.
 I. Grossmann, 2005. AIChE J., 51, 1846.
 C. Maravelias, 2005. Ind. Eng. Chem. Res. 44, 9129.
 J. Westerlund, M. Hastbacka, S. Forssell, T. Westerlund, 2007. Ind. Eng. Chem. Res. 46, 2781.