DESIGN AND SCHEDULING OF MULTIPURPOSE PLANTS USING A RTN CONTINUOUS-TIME FORMULATION

Pedro M. Castro†, Ana P. Barbosa Póvoa‡ and Augusto Q. Novais†
†DMS/INETI and ‡CEGIST/IST
†1649-038 Lisboa and ‡1049-001 Lisboa, Portugal

Abstract

This paper considers the problem of designing a plant according to its operational constraints. A uniform time grid, continuous-time formulation is proposed that considers the design and scheduling aspects simultaneously. Under the condition of linear task durations and equipment cost, the Resource Task Network (RTN) based formulation gives rise to Mixed Integer Linear Programs (MILPs) for total cost minimization. Its performance is illustrated through the solution of two batch-plant example problems taken from the literature and compared to the design and scheduling, non-uniform time grid, State Task Network (STN) based continuous-time formulation of Lin and Floudas (2001).

Keywords

Resource-Task Network, Batch Plants, Short-term, Uniform time grid, Event points.

Introduction

Multipurpose plants are general purpose facilities where a variety of products can be produced by sharing the available plant resources (raw materials, equipment, utilities and manpower) properly in time. The production of a particular product involves a sequence of operations that can be batch, semi-continuous or continuous in nature, where an equipment unit is usually suitable for more than a single operation. As a result, multipurpose plants are more flexible and more suitable for the production of small quantities of high value-added products with short life cycles, the current trend of consumer's demands in this competitive global market. However, these same special characteristics introduce extra degrees of complexity into the design and operation of such plants. In particular, it is not possible to design a plant without considering how it will be operated, neither it is possible to schedule all the required operations without knowing the plant configuration. Hence, design and scheduling must be considered simultaneously to avoid over or under design.

Mathematical formulations for design and scheduling problems can be classified in two groups based on the time representation. An example of a discrete-time formulation is the work of Barbosa-Póvoa and Macchietto (1994), while the formulation of Lin and Floudas (2001) is an example of a recent continuous-time formulation. Both are based on the State Task Network process representation.

This paper extends the work of Castro et al. (2004) by considering design and scheduling aspects simultaneously. Due to the lack of space, focus is set on the solution of short-term problems involving batch plants, although it can be easily adapted to periodic problems involving both batch and continuous operations.

Fundamental Concepts

In the proposed formulation, the time horizon (H) is divided into a fixed number of time intervals. The interval boundaries are called event points (set T) and their exact location (T_t) is unknown a priori. A single time grid keeps track of all events taking place (uniform time grid). The formulation of Lin and Floudas (2001) uses a different time grid for each equipment resource. Non-uniform time grids have the advantage of requiring fewer event points to model a particular problem and the disadvantage of requiring more complicated material balances and storage constraints.

The quality of the solution returned depends greatly on the number of event points considered (|T|). For MILP models, the returned solution is a global optimum solution only if the pre-specified number of |T| is sufficient and does not act as a hidden constraint. The search for the global optimum (or the best possible solution, if the problem becomes intractable) usually involves solving the problem for different values of |T| in succession until no increment is found in the objective function.

To account for the fact that the majority of process tasks does not span across the whole time horizon, the maximum number of event points allowed between the beginning and end of a task is defined through parameter Δt. Like before, the use of an exceedingly small value of Δt works also as a hidden model constraint so a similar procedure to that of |T| should be used.

One important breakthrough in continuous-time formulations in recent years came from allowing materials to reside in the equipment unit that produces them past their processing time. This added time has been called inherent waiting period or buffer time and can be allowed for tasks not producing unstable materials. For tasks producing materials subject to zero-wait policies (I^{ZW}), an additional set of constraints is used.

Concerning the process representation used, the RTN regards all processes as bipartite graphs comprising two types of nodes: resources (set R) and tasks (set I), the latter being operations that transform a certain set of resources into another set. Two variables are used to characterize the instance of task *i* starting at event point *t* and ending at *t*'. The binary variable $\overline{N}_{i,t,t'}$ identifies the occurrence of the task, while the nonnegative continuous variable $\overline{\xi}_{i,t,t'}$ gives the total amount of material processed between *t* and *t*'. The other required variables, which are all nonnegative, are the excess resource variables $R_{r,t}$, and the initial amounts R_r^0, which for equipment resources become the existence variables ($R_r^{\max}=1$, and through Eqs. 7 and 9, $R_r^0 \le 1\,\forall r\in R^{EQ}$).

The task processing time is assumed to be given by a constant (α_i) plus a term proportional to the amount of material being processed (β_i), whereas the amounts of each resource consumed/produced at the start/end of a task are assumed to be proportional to the binary ($\mu_{r,i}/\overline{\mu}_{r,i}$) and/or continuous ($\nu_{r,i}/\overline{\nu}_{r,i}$) extents of the task. The former parameters are usually linked with equipment resources, while the latter are typically linked with material resources. The other required parameter (sf_i) is used to allow the amount processed by task *i* to be lower than the minimum design capacity of the equipment.

To better understand the model constraints note that a 1:1 correspondence is assumed between tasks and equipment resources, so if a particular task can be performed in more than one equipment, one task will need to be defined for each unit.

$$\sum_{r\in R}\mu_{r,i} = \sum_{r\in R}\overline{\mu}_{r,i} = 1\,\forall i\in I \tag{1}$$

Mathematical Formulation

$$T_{t'} - T_t \ge \sum_{i\in I}\overline{\mu}_{r,i}(\alpha_i \overline{N}_{i,t,t'} + \beta_i\overline{\xi}_{i,t,t'}) \\ \forall r\in R^{EQ}, t\in T, t'\in T, t<t'\le \Delta t + t, t\ne |T| \tag{2}$$

$$T_{t'} - T_t \le H + \sum_{i\in I, I^{ZW}}\overline{\mu}_{r,i}(\alpha_i \overline{N}_{i,t,t'} + \beta_i\overline{\xi}_{i,t,t'}) - \\ H\cdot \sum_{i\in I,I^{ZW}}\overline{\mu}_{r,i}\overline{N}_{i,t,t'}\ \forall r\in R^{EQ}, t,t'\in T, t'\le \Delta t + t, t\ne |T| \tag{3}$$

$$T_1 = 0 \wedge T_{|T|} = H \tag{4}$$

$$\overline{N}_{i,t,t'} sf_i \sum_{r\in R}\overline{\mu}_{r,i}V_r^{\min} \le \overline{\xi}_{i,t,t'} \le \overline{N}_{i,t,t'}\sum_{r\in R}\overline{\mu}_{r,i}V_r^{\max} \\ \forall i\in I, t\in T, t'\in T, t<t'\le \Delta t + t, t\ne |T| \tag{5}$$

$$\overline{\xi}_{i,t,t'} \le \sum_{r\in R}\overline{\mu}_{r,i}V_r\ \forall i\in I, t\in T, t'\in T, t<t'\le \Delta t+t, t\ne |T| \tag{6}$$

$$R_{r,t} = R_r^0\big|_{t=1} + R_{r,t-1}\big|_{t>1} + \sum_{i\in I}\sum_{\substack{t'\in T\\ t<t'\le \Delta t+t}}(\mu_{r,i}\overline{N}_{i,t,t'} + \nu_{r,i}\overline{\xi}_{i,t,t'}) + \\ \sum_{i\in I}\sum_{\substack{t'\in T\\ t-\Delta t\le t'<t}}(\overline{\mu}_{r,i}\overline{N}_{i,t',t} + \overline{\nu}_{r,i}\overline{\xi}_{i,t',t})\ \forall r\in R, t\in T \tag{7}$$

$$R_r^0 V_r^{\min} \le V_r \le R_r^0 V_r^{\max}\ \forall r\in R^{EQ} \tag{8}$$

$$R_r^{\min} \le R_{r,t} \le R_r^{\max}\ \forall r\in R, t\in T \tag{9}$$

$$\min \sum_{r\in R^{EQ}}(\tilde{\alpha}_r R_r^0 + \tilde{\beta}_r V_r) - \sum_{r\in R} p_r R_{r,|T|} \tag{10}$$

The objective function considered (Eq. 10) minimizes the capital cost of units, which consists of a fixed term ($\tilde{\alpha}_r$) plus a term ($\tilde{\beta}_r$) proportional to the size of the unit (V_r), minus profits due to product sales. Other performance criteria can also be incorporated.

Case Studies

Two examples taken from Lin and Floudas (2001), BMFIX and KPSLIN, were chosen to illustrate the capabilities of the proposed formulation.

In the first example (BMFIX), two valuable products (S5 and S6) are to be produced from two raw-materials (S1 and S2). Three main equipment units (U1a, U1b and U2) and one storage vessel (V4) can be used for that purpose. The RTN representation of the process is given in Figure 1, while the other required data is given in Table 1.

The second example is more complex and features 4 possible main equipment units (Heater, Reactor1, Reactor2, Still) and four possible storage vessels (V4-V7). The RTN representation of the process is given in Figure 4, while the data is shown in Table 2.

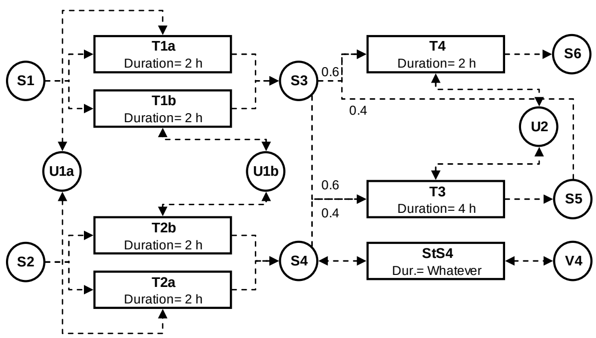

Figure 1. RTN representation for BMFIX

Table 1. Problem data for BMFIX

Resource	$V_r^{\min}/V_r^{\max}$	$\tilde{\alpha}_r/\tilde{\beta}_r$	p_r	$R_{r,\|T\|}$
U1a,U1b	50/150	20/0.5	-	-
U2	50/200	30/1	-	-
V4	10/100	1/0.1	-	-
S5	-	-	0.04	80
S6	-	-	0.015	80

Table 2. Problem data for KPSLIN

Resource	$V_r^{\min}/V_r^{\max}$	$\tilde{\alpha}_r/\tilde{\beta}_r$	p_r	$R_{r,\|T\|}$
Heater	20/50	100/0.2	-	-
Reactor1	50/70	150/0.5	-	-
Reactor2	70/70	120/0	-	-
Still	50/80	150/0.3	-	-
V4	10/30	30/0.1	-	-
V5	10/60	15/0.1	-	-
V6	10/70	10/0.1	-	-
V7	50/100	20/0.2	-	-
P1	-	-	0.02	40
P2	-	-	0.03	60

Results

The above mathematical formulation gives rise to Mixed Integer Linear Problems that were solved to optimality by GAMS/CPLEX 8.1 on a Pentium IV-2.53 GHz machine. The two example problems were solved for three different scenarios: A) all materials subject to ZW and exact demands for the final products (the one considered by Lin and Floudas, 2001); B) no ZW and fixed demands; C) ZW with minimum instead of fixed demands.

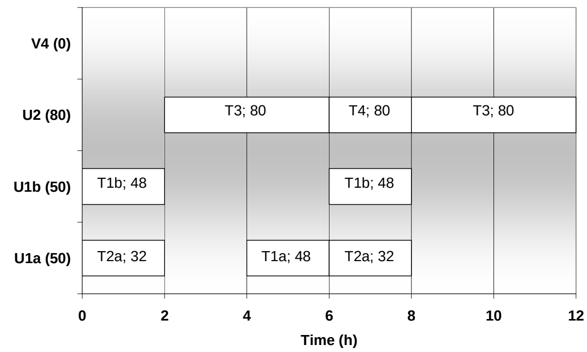

Figure 2. Optimal solution for BMFIX, C

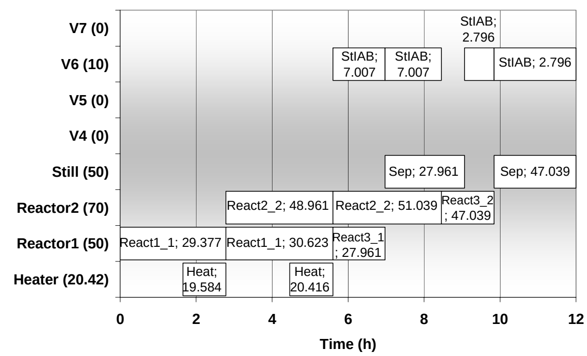

Figure 3. Best solution found for KPSLIN, C

The computational statistics are given in Table 3. All three scenarios of BMFIX are solved rather rapidly and the reported solutions are global optimal solutions. The optimal solution for scenario C is given in Figure 2, where it can be seen that the capacities of equipments U1a and U1b are their predefined minimum capacities (50). If these values were lower, both capacities would be equal to 48. The optimal schedule is quite similar to that of scenario A with the only difference being that more material is processed in the first three tasks of the schedule, thus leading to the production of 128 units of S5 instead of 80. The installed capacities are the same in all scenarios.

The second example, KPSLIN, is much harder to solve. Due to problem tractability, the global optimal solution can only be achieved for scenario B. The additional constraint of ZW policies has the effect of causing more event points to be used to get to the same solution. In fact, for scenario A, the solution for 10 event points is slightly worse (the capacity of the heater is equal to 20.416 instead of 20) than that of scenario B (7 event points). In scenario C (see Figure 3) a better solution is achieved due to the additional production of more 7.5 units of P2 and to the lower capacity of V6 (10 instead of 13.33).

In terms of computational performance, our formulation can only match that of the non-uniform time grid formulation of Lin and Floudas (2001) if the condition of ZW is dropped. Otherwise, the results for scenario A clearly show that more event points are required to get to the same solutions (for KPSLIN the difference between scenario B and L&F's scenario A is due to slightly different data, i.e. rounding errors, since the installed capacities are the same). Since a single increase in the number of event points has typically a 1 order of magnitude effect on computational performance this can quickly become a major inconvenience.

Conclusions

This paper presents a new RTN-based continuous-time formulation for the simultaneous design and scheduling of multipurpose plants. The efficiency of the proposed formulation was tested through the solution of two example problems taken from the literature for three different scenarios involving (or not) ZW policies and fixed or minimum product demands. The results obtained showed that our formulation performs similarly to that of Lin and Floudas (2001) only when the materials are not subject to ZW policies. When they are, our uniform time grid formulation may require a few more event points than that of Lin and Floudas (2001) to get to the same solutions, which is translated into a poorer performance.

Future work should include the development of a RTN-based non-uniform time grid formulation that would lead to a better computational performance and hence allow the solution of larger problems. In addition, and since the design of the plant is a long term decision, the more stable, periodic mode of operation should be preferred over the short-term mode.

References

Barbosa-Póvoa, A., Macchietto, S. (1994). Detailed Design of Multipurpose Batch Plants. *Comp. Chem. Eng., 18*, 1013.

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.

Lin, X., Floudas, C. (2001). Design, Synthesis and Scheduling of Multipurpose Batch Plants via an Effective Continuous-time Formulation. *Comp. Chem. Eng., 25*, 665.

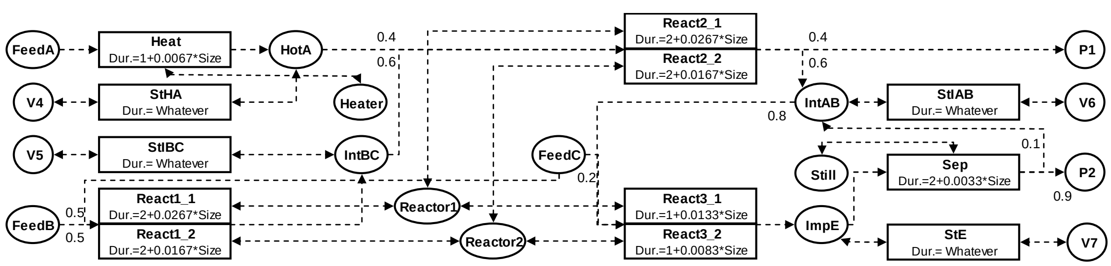

Figure 4. RTN representation for KPSLIN

*Table 3. Computational statistics (H=12 h, Δt =2, *HP-C160 workstation)*

Example	BMFIX				KPSLIN			
Authors	L&F	This paper			L&F	This paper		
Scenario	A	A	B	C	A	A	B	C
|T|	5	6	5	6	6	10	7	10
Integer variables	59	43	30	43	128	134	74	134
Continuous variables	175	163	126	163	341	468	294	468
Constraints	332	240	164	240	877	710	395	710
Obj. relaxed MILP	-	59.81	90.3	47.71	-	103.8	182.98	99.656
Obj. MILP	195.6	195.6	195.6	193.7	572.9	572.82	572.73	572.26
CPUs	0.4*	0.38	0.19	0.30	22.5*	2954	9.3	11185