18th European Symposium on Computer Aided Process Engineering – ESCAPE 18
Bertrand Braunschweig and Xavier Joulia (Editors)

A Novel Network-based Continuous-time Formulation for Process Scheduling

Diego M. Giménez,[a] Gabriela P. Henning,[a] Christos T. Maravelias[b]

[a]*INTEC (Universidad Nacional del Litoral - CONICET), Güemes 3450, S3000GLN Santa Fe, Argentina*
[b]*Department of Chemical and Biological Engineering, University of Wisconsin – Madison, USA*

Abstract

We present a novel network-based continuous-time formulation for process scheduling that addresses multiple limitations of existing approaches. Specifically, it handles non-simultaneous transfers of input/output materials to/from processing units; it employs a more flexible time representation; and it explicitly accounts for unit connectivity. This is accomplished via the modelling of two key issues: (i) the *state* of a processing/storage unit, and (ii) the material transfer between processing and storage units. The proposed formulation allows us to model many complexities associated with process scheduling and obtain solutions to problems that cannot be addressed by existing methods.

Keywords: process scheduling; network-based continuous-time formulation.

1. Introduction

Existing network-based scheduling formulations are based on the state-task network (STN) or the resource-task network (RTN) representation (Kondili et al., 1994; Pantelides, 1994). In these formulations it is implicitly assumed that: a) material transfer between units is always possible, i.e. all processing units are connected to all the vessels that are used for the storage of the corresponding input and output materials; b) all input/output materials of a task are transferred simultaneously to/from the processing unit when the task starts/ends; and c) stable output materials can be temporarily stored in a processing unit after a task is completed, but stable input materials cannot be temporarily stored before a task starts, i.e. in continuous time representations the beginning of a task must coincide with a time point and at such point all the materials should be available.

However, these assumptions do not always hold. For example, in recovery and purification processes the solvent can be drained earlier. Similarly, in certain chemical reactions reactants are fed before the beginning of the task, which actually occurs when the catalyst is added. Interestingly, despite the large number of methods recently proposed to tackle process scheduling, there are very few attempts to address these limitations. Barbosa-Póvoa and Macchietto (1994) discuss the issue of connectivity and material transfer in the context of discrete-time formulations. However, to our knowledge none of the existing methods deals with the shortcomings due to assumption (c).

The goal of this paper is the development of a novel approach that overcomes these shortcomings. The key ideas of the proposed method are presented in section 2. The main variables and constraints of the mixed-integer linear programming (MILP) formulation are presented in section 3. The advantages of the new method are illustrated through a small example problem in section 4.

2. Proposed Approach

2.1. Time Representation

We introduce a new global continuous-time representation: a set of global time points $k \in \boldsymbol{K}=\{1, 2, \ldots K\}$ span the scheduling horizon from 0 to H delimiting a set of K-1 time intervals of unknown length, where interval k starts at time point k and ends at k+1. The novelty of this new representation is that tasks do not have to start (or finish) exactly at a time point. In other words, a task assigned to start on a unit at time point k can actually start any time within interval k (late beginning) as Fig. 1a shows. Similarly, a task that is assigned to end at time point k can actually end at any time within interval $k-1$ (early end). Thus, the new representation can potentially lead to formulations with fewer time points.

2.2. Process States

Unlike previous network-based models, a processing unit can be used both for carrying out process tasks or storing input/output materials before/after the start/end of a task. Furthermore, input/output materials do not have to be simultaneously transferred to/from a unit. Hence, a unit $j \in \boldsymbol{J}$ can be in three different states during time interval k (Fig. 1b): a) *idle* state ($W_{j,k}$=1); b) *storage* state; and c) *execution* state ($E_{j,k}$=1). If used for storage, then it can either be used for input ($S^I_{j,k}$=1) or output ($S^O_{j,k}$=1) materials. The *execution* state is delimited by the formal task boundaries, which are given by the values of the *event* variables associated with task beginning and end. If a task $i \in \boldsymbol{I_j}$ is assigned to start in unit j within interval k (at or after time point k) then $X_{i,j,k}$=1, where $\boldsymbol{I_j}$ is the set of tasks that can be carried out in unit j. If a task is assigned to end within interval $k-1$ (at or before time point k) then $Y_{i,j,k}$=1. In addition, an auxiliary variable indicates when a task started before time point k is still being processed in unit j at such time ($Z_{j,k}$=1).

2.3. Time Balances

To accurately account for a late beginning or early end of a task in unit j after and before time point k, respectively, we introduce two new variables: a) $\overline{T}^{LB}_{j,k}$ that denotes the *lateness* within interval k in starting a task, and b) $\overline{T}^{EE}_{j,k}$ that denotes the *earliness* within interval $k-1$ in finishing a task. We also introduce variables to model the time a processing unit remains idle ($\overline{T}^{ID}_{j,k}$) or is used for storage $\overline{T}^{S}_{j,k}$, during interval k (Fig. 1c).

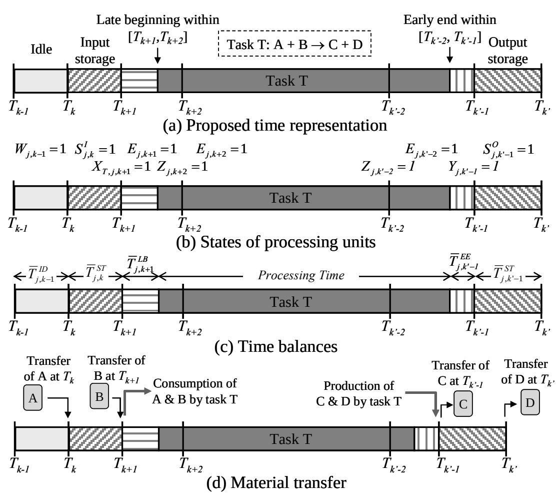


Figure 1. Key concepts of the proposed formulation

2.4. Material Transfer and Material Balances

Material transfer is formulated explicitly via *flow* variables. In the context of this contribution, the concept of *flow* represents an instantaneous transfer of material from a storage/processing unit to another physically connected storage/processing unit. Only one material can be stored in a storage unit $v \in \boldsymbol{V}$ in any time interval, but multiple input/output materials can be simultaneously stored in a processing unit before/after a task starts/ends. Material balance constraints in storage units include only incoming and outgoing flows. The corresponding balances in processing units include the incoming and outgoing flows as well as production and consumption terms that correspond to the transformation of materials by process tasks (Fig. 1d).

3. Mathematical Formulation

In addition to the *event* (X_{ijk}, Y_{ijk}), *state* ($W_{jk}, E_{jk}, S_{jk}^{I}, S_{jk}^{O}$) and auxiliary ($Z_{jk}$) binary variables, time corresponding to point k (T_k), and timing variables ($\overline{T}_{jk}^{LB}, \overline{T}_{jk}^{EE}, \overline{T}_{jk}^{ID}, \overline{T}_{jk}^{ST}$), the following continuous variables are defined:

- Flows $F_{m,j,v,k}^{UV}, F_{m,j,j',k}^{UU}, F_{m,v,v',k}^{VV}, F_{m,v,j,k}^{VU}$ to represent instantaneous transfers of material m at time point k between storage vessels (V) and processing units (U), where the letter sequence in the superscript denotes the direction of the transfer.
- Batch-sizes $B_{i,j,k}^{S}, B_{i,j,k}^{P}, B_{i,j,k}^{F}$ to denote the total amount of task i that starts to be processed, that keeps processing and finishes, respectively, in unit j at time point k.
- Inventory $I_{m,v,k}^{V}$ of material m in storage vessel v during time interval k, and inventory $I_{m,j,k}^{UI} / I_{m,j,k}^{UO}$ of input/output material m in processing unit j during time interval k.

To facilitate the presentation, in the remaining we use capital letters for variables, small letters for parameters (with the exception of horizon H) and bold capital letters for sets.

3.1. State Constraints

Clearly, a processing unit has to be in exactly one state during each time interval:

$$E_{j,k} + W_{j,k} + S_{j,k}^{I} + S_{j,k}^{O} = 1, \quad \forall j, k < K \tag{1}$$

A unit is in the *execution* state during interval k if a task starts within such interval, i.e. at or after point k, or another task started in a previous interval is still being executed:

$$E_{j,k} = Z_{j,k} + \sum_{i \in \boldsymbol{I_j}} X_{i,j,k}, \quad \forall j, k < K \tag{2}$$

Finally, the $Z_{j,k}$ auxiliary variable (denoting that at time point k unit j continues executing a task previously started) can be defined as follows:

$$Z_{j,k} = Z_{j,k-1} + \sum_{i \in \boldsymbol{I_j}} X_{i,j,k-1} - \sum_{i \in \boldsymbol{I_j}} Y_{i,j,k} \quad \forall j, k > 1 \tag{3}$$

3.2. Timing Constraints

A late beginning (early end) with respect to time point k can only occur if a task is assigned to start (end) in unit j at such time point, as the following inequalities indicate:

$$\overline{T}_{j,k}^{LB} \le H \sum_{i \in \boldsymbol{I_j}, i \notin \boldsymbol{I}^{cZW}} X_{i,j,k}, \quad \forall j, k < K \, ; \qquad \overline{T}_{j,k}^{EE} \le H \sum_{i \in \boldsymbol{I_j}, i \notin \boldsymbol{I}^{pZW}} Y_{i,j,k}, \quad \forall j, k > 1 \tag{4}$$

where $\boldsymbol{I}^{cZW}/\boldsymbol{I}^{pZW}$ are the sets of tasks consuming/producing unstable materials for which late beginnings and early ends are forbidden.

Similarly, storage and idle times occur only if the unit is in the corresponding state:

$$\overline{T}_{j,k}^{ST} \le H(S_{j,k}^{I} + S_{j,k}^{O}), \quad \forall j,k < K \quad ; \quad \overline{T}_{j,k}^{ID} \le HW_{j,k}, \quad \forall j,k < K \tag{5}$$

In that case, the idle and storage times should be equal to the length of the time interval:

$$T_{k+1} - T_k - H(1 - S_{j,k}^{I} - S_{j,k}^{O} - W_{j,k}) \le \overline{T}_{j,k}^{ST} + \overline{T}_{j,k}^{ID} \le T_{k+1} - T_k \quad \forall j,k < K \tag{6}$$

3.3. Time Balance Constraints

Constraints (7)-(9) are used to define the continuous-time grid and enforce the appropriate timing constraints without resorting to big-M terms:

$$T_k \ge \sum_{k' \le k} \sum_{i \in \boldsymbol{I_j}} (a_{ij} Y_{i,j,k'} + b_{ij} B_{i,j,k'}^{F}) + \sum_{k' \le k} \overline{T}_{j,k'}^{EE} + \sum_{k' < k} (\overline{T}_{j,k'}^{LB} + \overline{T}_{j,k'}^{ST} + \overline{T}_{j,k'}^{ID}), \quad \forall j,k > 1 \tag{7}$$

$$\sum_{k' \ge k} \sum_{i \in \boldsymbol{I_j}} (a_{ij} X_{i,j,k'} + b_{ij} B_{i,j,k'}^{S}) + \sum_{k' > k} \overline{T}_{j,k'}^{EE} + \sum_{k' \ge k} (\overline{T}_{j,k'}^{LB} + \overline{T}_{j,k'}^{ST} + \overline{T}_{j,k'}^{ID}) \le H - T_k, \quad \forall j,k < K \tag{8}$$

$$\sum_{k} (\overline{T}_{j,k}^{LB} + \overline{T}_{j,k}^{EE} + \overline{T}_{j,k}^{ID} + \overline{T}_{j,k}^{ST}) + \sum_{i \in \boldsymbol{I_j}} \sum_{k} (a_{ij} Y_{i,j,k} + b_{ij} B_{i,j,k}^{F}) = H, \quad \forall j \tag{9}$$

where a_{ij} and b_{ij} are the fixed and proportional processing time constants.

3.4. Batching constraints

Batch-size variables are constrained as follows:

$$\beta_j^{MIN} X_{i,j,k} \le B_{i,j,k}^{S} \le \beta_j^{MAX} X_{i,j,k}, \quad \forall i \in I, j \in J_i, k < K \tag{10}$$

$$B_{i,j,k}^{F} \le \beta_j^{MAX} Y_{i,j,k}, \quad \forall i \in I; j \in J_i, k > 1 \tag{11}$$

$$B_{i,j,k}^{S} + B_{i,j,k}^{P} = B_{i,j,k+1}^{P} + B_{i,j,k+1}^{F}, \quad \forall i, j \in J_i, k < K \tag{12}$$

where $\beta_j^{MIN}/\beta_j^{MAX}$ is the minimum/maximum capacity of unit j.

3.5. Material Balances

3.5.1. Storage Vessels

The material balance constraint in storage vessels is expressed as follows:

$$I_{m,v,k}^{V} = I_{m,v,k-1}^{V} - \sum_{j \in \boldsymbol{J_v}} F_{m,v,j,k}^{VU} - \sum_{v' \in \boldsymbol{V_v}} F_{m,v,v',k}^{VV} + \sum_{j \in \boldsymbol{J_v}} F_{m,j,v,k}^{UV} + \sum_{v' \in \boldsymbol{V_v}} F_{m,v',v,k}^{VV}, \quad \forall\, m \notin (\boldsymbol{M}^{NIS} \cup \boldsymbol{M}^{ZW}), v \in \boldsymbol{V}_m, k \tag{13}$$

where $\boldsymbol{J_v}/\boldsymbol{V_v}$ are the sets of units/vessels connected to vessel v, $\boldsymbol{M}^{NIS}/\boldsymbol{M}^{ZW}$ are the sets of tasks for which non-intermediate storage/zero-wait storage policies are enforced, and $\boldsymbol{V_m}$ is the set of vessels that can be used to store up material m. The inventory is constrained not to exceed the maximum storage capacity $\varsigma_{m,v}^{MAX}$ by expression (14).

$$I_{m,v,k}^{V} \le \varsigma_{m,v}^{MAX}, \quad \forall\, m \notin (\boldsymbol{M}^{NIS} \cup \boldsymbol{M}^{ZW}), v \in \boldsymbol{V}_m, k \tag{14}$$

3.5.2. Processing Units

The corresponding material balances in processing units for input and output materials are expressed via equations (15) and (16), respectively:

$$I_{m,j,k}^{UI} = I_{m,j,k}^{UI} + \sum_{v \in \boldsymbol{V_i} \cap \boldsymbol{V_m}} F_{m,v,j,k}^{VU} + \sum_{j' \in \boldsymbol{J_j}} F_{m,j',j,k}^{UU} + \sum_{i \in \boldsymbol{I_j} \cap \boldsymbol{I_m^C}} \gamma_{im} B_{i,j,k}^{S}, \quad \forall m, j, k \tag{15}$$

$$I^{UO}_{m,j,k} = I^{UO}_{m,j,k} + \sum_{i \in \boldsymbol{I_j} \cap \boldsymbol{I^P_m}} \gamma_{im} B^F_{i,j,k} - \sum_{v \in \boldsymbol{V}_j \cap \boldsymbol{V_m}} F^{PS}_{m,j,v,k} - \sum_{j' \in \boldsymbol{J_j}} F^{PP}_{m,j',j,k}, \quad \forall m,j,k \qquad (16)$$

where $\boldsymbol{I_m}^{\boldsymbol{C}}/\boldsymbol{I_m}^{\boldsymbol{P}}$ are the sets of tasks consuming/producing material m, $\boldsymbol{J_j}/\boldsymbol{V_j}$ are the sets of units/vessels connected to unit j, and γ_{im} is the stoichiometric coefficient of material m in task i (negative if consumed).

Note that inventory level changes in processing units are due to material transfer as well as material consumption and production by processing tasks. Obviously, input/output materials can only be stored in a processing unit if the unit is in the corresponding state:

$$\sum_m I^{UI}_{m,j,k} \le \beta^{MAX}_j S^I_{j,k}, \quad \forall j,k < K \;\; ; \;\; \sum_m I^{UO}_{m,j,k} \le \beta^{MAX}_j S^O_{j,k}, \quad \forall j,k < K \qquad (17)$$

3.6. Utility Constraints

The total amount $R_{r,k}$ of utility r consumed at time interval k is calculated through equation (18), and constrained not to exceed the maximum availability ρ_r^{MAX} by (19):

$$R_{r,k} = R_{r,k-1} + \sum_{i \in \boldsymbol{I_r}} \sum_{j \in \boldsymbol{J_i}} [f_{ijr}(X_{i,j,k} - Y_{i,j,k}) + g_{ijr}(B^S_{i,j,k} - B^F_{i,j,k})], \quad \forall r,k < \mathrm{K} \qquad (18)$$

$$R_{r,k} \le \rho^{MAX}_r, \quad \forall r,k < K \qquad (19)$$

where $\boldsymbol{I_r}$ is the set of tasks requiring utility r, and f_{ijr} and g_{ijr} are the fixed and proportional, respectively, constants for the consumption of utility r by task i in unit j.

3.7. Objective function

The proposed model consists of expressions (1)–(19) and can be used to tackle various objective functions. In this short communication the profit maximization is studied:

$$z = \max \sum_{m \in \boldsymbol{M^{FP}}} \sum_{v \in \boldsymbol{V_m}} \pi_m I^V_{m,v,K} \qquad (20)$$

where π_m is the price of material m and $\boldsymbol{M^{FP}}$ is the set of products that can be sold.

4. Example

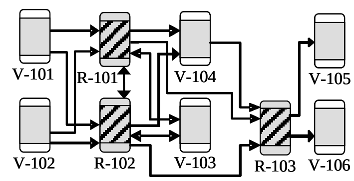
A scheduling problem corresponding to a simple multipurpose batch plant is studied in order to show the main advantages of the proposed formulation. The process structure, task information and material data are described in Fig. 2. The profit maximization for a time horizon of 8 hours (H=8) is pursued. The problem instance was solved with the aim of getting an optimal schedule in a case where existing models cannot even obtain a feasible solution. In this example it is easy to note that, since no intermediate initial inventory is held, the only way to obtain final products is by performing task T2 first (so INT1 and INT2 can be available), then executing T1 (so INT3 can be available), and finally either T3 or T4. Nevertheless, since a NIS policy is adopted for INT2, T4 should begin immediately after task T2 finishes. However, this is infeasible for current approaches because INT3 cannot be available at that time (INT3 is produced by T1, which cannot start until T2 finishes since it consumes INT1). The proposed formulation overcomes this limitation by allowing a temporal storage of INT2 in unit R-103 until INT3 becomes available. Thus, the material load/discharge is decoupled from the task beginning/end.

Elements	*Tasks*	*Stoichiometric Relations*
Processing Units: R-101, R-102, R-103	T1	0.8 RM1 + 0.2 INT1 → INT3
Tasks: T1, T2, T3, T4	T2	RM2 → 0.3 INT1 + 0.7 INT2
Vessels: V-101, V-102, V-103, V-104, V-105, V-106	T3	INT3 → P1
Materials: RM1, RM2, INT1, INT2, INT3, P1, P2	T4	0.6 INT2 + 0.4 INT3 → P2
Utilities: Hot Steam (HS), Cooling Water (CW)		

Material Data

m	π_m ($/kg)	*v*	$\varsigma^{MAX}_{m,v}$	Inventory (kg)
RM1	0	V-101	UIS	1000
RM2	0	V-102	UIS	1000
INT1	0	V-103	FIS (200)	0
INT2	0	-	NIS	0
INT3	0	V-104	FIS (500)	0
P1	30	V-105	UIS	0
P2	40	V-106	UIS	0

Plant Topology



Task Information

i	*j*	a_{ij} (h)	b_{ij} (h/kg)	β_j^{MIN}(kg)	β_j^{MAX} (kg)	*r*	f_{ijr} (kg/min)	g_{ijr} (kg/min.kg)	ρ_r^{MAX} (kg/min)
T1	R-101	0.5	0.025	40	80	HS	6	0.25	30
	R-102	0.5	0.04	25	50	HS	4	0.25	30
T2	R-101	0.75	0.0375	40	80	CW	4	0.3	30
	R-102	0.75	0.06	25	50	CW	3	0.3	30
T3	R-103	0.25	0.0125	40	80	HS	8	0.2	30
T4	R-103	0.5	0.025	40	80	CW	4	0.5	30

Figure 2. Example of a very simple multipurpose facility

Figure 3 presents the optimal schedule obtained by implementing the proposed MILP model in GAMS/CPLEX 10.0 on a Pentium IV (3.0 GHz) PC with 2 GB of RAM, adopting a zero integrality gap. It can be seen that six global time points (five time intervals) were required to obtain this optimal solution. The model instance involved 87 binary variables, 655 continuous ones, and 646 constraints. An optimal solution of $3592.2 was found in only 0.87 s by exploring 282 nodes.

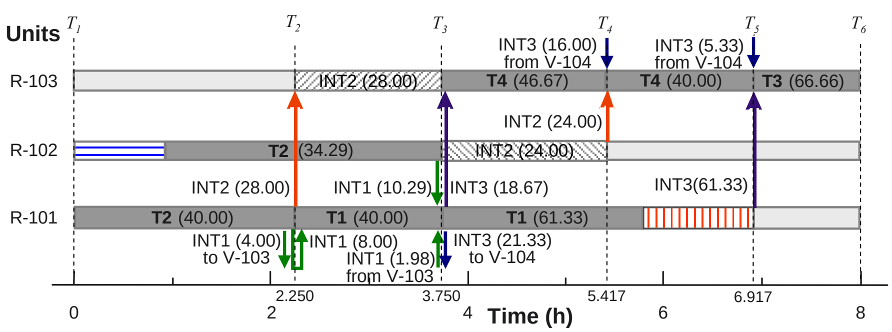


Figure 3. Optimal schedule for the motivating example

References

A. P. Barbosa-Póvoa and S. Macchietto, 1994, Detailed design of multipurpose batch plants, Comp. Chem. Eng., 18, 11/12, 1013-1042.

E. Kondili, C. C. Pantelides, and W. H. Sargent, 1993, A general algorithm for short-term scheduling of batch operations-I. MILP formulation, Comp. Chem. Eng., 17, 2, 211-227.

C. C. Pantelides, 1994, Unified frameworks for optimal process planning and scheduling. In: Foundations of computer-aided process operations, New York, 253-274.