On the Relation of Continuous- and Discrete-Time State–Task Network Formulations

Christos T. Maravelias
Dept. of Chemical and Biological Engineering, University of Wisconsin, Madison, WI 53706

Ignacio E. Grossmann
Dept. of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

DOI 10.1002/aic.10684
Published online October 5, 2005 in Wiley InterScience (www.interscience.wiley.com).

Keywords: scheduling, state–task network, time representations

Introduction

The purpose of this research note is to formally study the relationship between discrete-time and continuous-time state–task network[1] (STN) mixed-integer linear programming (MILP) formulations. The STN and its equivalent resource–task network[2] (RTN) representation were proposed as general formulations for the scheduling of complex batch plants. In the original discrete-time formulation,[1,3] the time horizon is divided into N intervals of equal duration, common for all units, and tasks must begin and finish exactly at a time point, which means that the duration of the intervals must be equal to the greatest common factor of the constant processing times. Although discrete-time models tend to exhibit strong linear programming (LP) relaxations, their disadvantage is that variable processing times can be handled only as discrete approximations, and that the number of intervals may be so large that the resulting model is too hard to solve.

To overcome these limitations several researchers have proposed continuous-time STN/RTN-based formulations.[4-11] In continuous-time models the time horizon is divided into time intervals of unequal and unknown duration. Continuous-time representations account for variable processing times and require significantly fewer time intervals, leading to smaller problems. However, because time points are not fixed, constraints that match a time point with the start (or finish) of a task are necessary, resulting in poor LP relaxations. Moreover, the number of intervals needed to accurately represent the optimal solution is unknown, thus requiring an iterative procedure.

Correspondence concerning this article should be addressed to C. T. Maravelias at maravelias@wisc.edu.

Continuous-time models are more accurate and can in principle yield better solutions. In practice, however, they often yield substantially suboptimal solutions, either because of their poor LP relaxation or because of the fact that the number of intervals is unknown. Thus, discrete-time models remain popular for industrial problems, keeping open the debate over the effectiveness of the two formulations.

Because discrete- and continuous-time models have been developed independently, an interesting question is whether one can in fact algebraically show that discrete models can be derived as a special case of continuous models. The reason for posing this question is not only of theoretical interest, but also because, by establishing such a relationship, one can think of applying unified solution techniques to both types of representations. In this paper we study the relationship between the continuous- and discrete-time representations. Specifically, we formally show that discrete time can be derived as a special case of continuous-time models.

Continuous-time STN Formulation

A compact form of the continuous-time model of Maravelias and Grossmann[10] is used for the analysis. Redundant and tightening constraints that were added to improve the performance of LP-based branch-and-bound method and tighten the LP relaxation are omitted. A common, continuous partition of the time horizon is used to account for all possible plant configurations and resource constraints other than those on units; $N + 1$ time points $\{0, 1, 2, \ldots, N\}$ are defined and the time horizon is partitioned into N periods $\{1, 2, \ldots N\}$, where period n starts at time point $n - 1$ and finishes at time point n. Assignment constraints are expressed through task binaries Ws_{in} and Wf_{in}. Binary Ws_{in} is 1 if task i starts at time point n, and binary Wf_{in} is 1 if task i finishes at or before time point n.

The start time (Ts_{in}) of task i is always equal to time point T_n and thus time-matching constraints are used only for the finish time (Tf_{in}) of task i. The batch size of task i that starts at, is being processed at, and finishes at or before time point n is denoted by Bs_{in}, Bp_{in}, and Bf_{in}, respectively. The amount of state s at time point n is denoted by S_{sn} and the amount of resource r consumed by various tasks at time point n is denoted by R_{rn}. The amount of state s consumed (produced) by task i at time point n is denoted by B^I_{isn} (B^O_{isn}). The details and derivation of the proposed model can be found in Maravelias and Grossmann.[10]

Assignment constraints

Constraint 1 is the main assignment constraint and enforces the condition that not more than one task can be processed in a unit at any time, where $I(j)$ is the subset of tasks that can be assigned to unit j. Constraint 2 enforces the condition that all tasks that start must finish, whereas constraints 3 and 4 enforce the condition that not more than one task can start or finish on a specific unit at any time:

$$\sum_{i\in I(j)} \sum_{n'\leq n} (Ws_{in'} - Wf_{in'}) \leq 1 \qquad \forall\, j,\ \forall\, n \tag{1}$$

$$\sum_{n} Ws_{in} = \sum_{n} Wf_{in} \qquad \forall\, i \tag{2}$$

$$\sum_{i\in I(j)} Ws_{in} \leq 1 \qquad \forall\, j,\ \forall\, n \tag{3}$$

$$\sum_{i\in I(j)} Wf_{in} \leq 1 \qquad \forall\, j,\ \forall\, n \tag{4}$$

Batch-size constraints and material balances

Constraints 5 and 6 impose upper and lower bounds on the batch sizes Bs_{in} and Bf_{in}, whereas constraint 7 enforces variables Bs_{in} and Bf_{in} to be equal for the same task. The amount of state s consumed (B^I_{isn}) and produced (B^O_{isn}) by task i at time n is calculated through constraints 8 and 9, respectively, where $O(s)/I(s)$ is the set of tasks producing/consuming state s, and ρ_{is} is the stoichiometric coefficient of state s in task i. Constraint 10 is the mass balance and the capacity constraint for state s at time n, where C_s is the storage capacity:

$$B_i^{MIN} Ws_{in} \leq Bs_{in} \leq B_i^{MAX} Ws_{in} \qquad \forall\, i,\ \forall\, n \tag{5}$$

$$B_i^{MIN} Wf_{in} \leq Bf_{in} \leq B_i^{MAX} Wf_{in} \qquad \forall\, i,\ \forall\, n \tag{6}$$

$$Bs_{in-1} + Bp_{in-1} = Bp_{in} + Bf_{in} \qquad \forall\, i,\ \forall\, n \tag{7}$$

$$B^I_{isn} = \rho_{is} Bs_{in} \qquad \forall\, i,\ \forall\, n,\ \forall\, s \in SI(i) \tag{8}$$

$$B^O_{isn} = \rho_{is} Bf_{in} \qquad \forall\, i,\ \forall\, n,\ \forall\, s \in SO(i) \tag{9}$$

$$S_{sn} = S_{s,n-1} + \sum_{i\in O(s)} B^O_{isn} + \sum_{i\in I(s)} B^I_{isn} \leq C_s \quad \forall\, s,\ \forall\, n \tag{10}$$

Resource constraints

The amount of renewable resource r required by task i that starts at n, R^I_{rn}, is calculated by constraint 11. The same amount, R^O_{rn}, is "released" when task i finishes and is calculated by constraint 12. The total amount of resource r required at time n is calculated and bounded not to exceed the maximum availability R_r^{MAX} by Eq. 13:

$$R^I_{irn} = \gamma_{ir} Ws_{in} + \delta_{ir} Bs_{in} \qquad \forall\, i,\ \forall\, r,\ \forall\, n \tag{11}$$

$$R^O_{irn} = \gamma_{ir} Wf_{in} + \delta_{ir} Bf_{in} \qquad \forall\, i,\ \forall\, r,\ \forall\, n \tag{12}$$

$$R_{rn} = R_{rn-1} - \sum_{i} R^O_{irn} + \sum_{i} R^I_{irn} \leq R_i^{MAX} \qquad \forall\, r,\ \forall\, n \tag{13}$$

Calculation of duration and finish time

The duration (D_{in}) and the finish time (Tf_{in}) of a task are calculated through constraints 14, and 15 and 16, respectively. If task i does not start at time point n, its finish time Tf_{in} is constrained to be equal to Tf_{in-1} by constraint 17, where H is the scheduling horizon. The elimination of start times Ts_{in} is accomplished through constraint 18:

$$D_{in} = \alpha_i Ws_{in} + \beta_i Bs_{in} \qquad \forall\, i,\ \forall\, n \tag{14}$$

$$Tf_{in} \leq Ts_{in} + D_{in} + H(1 - Ws_{in}) \qquad \forall\, i,\ \forall\, n \tag{15}$$

$$Tf_{in} \geq Ts_{in} + D_{in} - H(1 - Ws_{in}) \qquad \forall\, i,\ \forall\, n \tag{16}$$

$$D_{in} \leq Tf_{in} - Tf_{in-1} \leq HWs_{in} \qquad \forall\, i,\ \forall\, n \tag{17}$$

$$Ts_{in} = T_n \qquad \forall\, i,\ \forall\, n \tag{18}$$

Activation of Wf_{in} binary variables

The time matching between time points and finish times is achieved through constraints 19, 20, and 21. Note that in the general case a task may finish at or before a time point n (constraint 20), whereas a task must finish exactly at a time point (constraint 21) if it produces a state for which zero-wait policy applies, where ZWI is the subset of tasks that produce a zero-wait state:

$$Tf_{in-1} \leq T_n + H(1 - Wf_{in}) \qquad \forall\, i,\ \forall\, n \tag{19}$$

$$Tf_{in-1} \geq T_{n-1} - H(1 - Wf_{in}) \qquad \forall\, i \notin ZWI,\ \forall\, n \tag{20}$$

$$Tf_{in-1} \geq T_n - H(1 - Wf_{in}) \qquad \forall\, i \in ZWI,\ \forall\, n \tag{21}$$

Time ordering

Equations 22–24 define the start and the end of the time horizon and enforce an ordering among time points:

$$T_{n=0} = 0 \tag{22}$$

$$T_{n=N} = H \tag{23}$$

$$T_{n+1} \geq T_n \qquad \forall\, n \tag{24}$$

The continuous-time model (M1) consists of constraints 1–25, where Eq. 18 is used to eliminate Ts_{in}.

$$Ws_{in},\ Wf_{in} \in \{0, 1\},$$
$$Bs_{in},\ Bp_{in},\ Bf_{in},\ S_{sn},\ T_n,\ Tf_{in},\ D_{in},\ B^I_{isn},\ B^O_{isn},\ R^I_{irn},\ R^O_{irn},\ R_{rn} \geq 0 \tag{25}$$

Discrete-time STN Formulation

The discrete-time model (M2) of Shah et al.[3] is used for the analysis. To keep the notation uniform and make the derivation easier, we use task decoupling to drop index j from variables, we use the index n for time periods (instead of t), the index r for resources (instead of u), the assignment binary Ws_{in} (instead of W_{ijt}), and the continuous variables Bs_{in}, S_{sn}, and R_{rn} (instead of B_{ijt}, S_{st}, and U_{ut}, respectively). Constraint 26 is the unit allocation constraint that enforces that at most one task is processed at unit j at any time point, where τ_i is the fixed processing time of task i. Constraint 27 is a batch-size constraint (note that we use the original constraint of Kondili et al.[1]). Constraint 28 is the material balance equation for state s at time n, where we assume that there are no intermediate deliveries and/or shipments and that all states are consumed/produced at the beginning/end of a processing task. Constraint 29 is the utility constraint, where we assume constant utility consumption during a processing task. Time points T_n are fixed parameters in discrete-time models, but we have included constraint 30 for comparison with continuous-time models.

$$\sum_{i \in I(j)} \sum_{n' \geq n-\tau_i+1}^{n' \leq n} Ws_{in'} \leq 1 \qquad \forall\, j,\ \forall\, n \tag{26}$$

$$B_i^{MIN} Ws_{in} \leq Bs_{in} \leq B_i^{MAX} Ws_{in} \qquad \forall\, i,\ \forall\, n \tag{27}$$

$$S_{sn} = S_{s,n-1} + \sum_{i \in O(s)} \rho_{is} Bs_{in-\tau_i} + \sum_{i \in I(s)} \rho_{is} Bs_{in} \leq C_s \qquad \forall\, s,\ \forall\, n \tag{28}$$

$$R_{rn} = \sum_i \sum_{n' \geq n-\tau_i+1}^{n' \leq n} (\gamma_{ir} Ws_{in'} + \delta_{ir} Bs_{in'}) \leq R_r^{MAX} \qquad \forall\, r,\ \forall\, n \tag{29}$$

$$T_n = n\left(\frac{H}{N}\right) \qquad \forall\, n \tag{30}$$

The discrete-time model (M2) consists of constraints 26–31.

$$Ws_{in} \in \{0, 1\},\ Bs_{in},\ S_{sn},\ R_{rn} \geq 0 \tag{31}$$

Reduced Continuous-time Representation

In this section we show that model (M2) is a special case of (M1). More specifically, we show that model (M2) is obtained from (M1) if:

(1) the time grid is fixed, and

(2) the processing times of all tasks are constant multiples of the interval of the fixed time grid.

Assumptions

The assumption of fixed time grid with equal time intervals means that variables T_n of model (M1) are fixed, and given by Eq. 30. The assumption that processing times τ_i are constant multiples of the duration of the uniform time intervals implies the following:

If task i starts at time n, its duration D_{in} is equal to τ_i; otherwise it is zero. Thus, D_{in} is always given by

$$D_{in} = \tau_i Ws_{in} \qquad \forall\, i,\ \forall\, n \tag{32}$$

Constraint 32 can also be derived from Eq. 14 for $\alpha_i = \tau_i$ and $\beta_i = 0$.

Because the time grid and the processing times are fixed, if task i starts at time point n, it will necessarily finish at time point $n + \tau_i$. Thus, the binary variable Wf_{in} is always given by

$$Wf_{in} = Ws_{i,n-\tau_i} \qquad \forall\, i,\ \forall\, n \geq \tau_i \tag{33}$$

If no tasks that finish after H are allowed to start, then we have $W_{sin} = 0$, $\forall n > N - \tau_i$. Moreover, because task i cannot finish before τ_i, we also have $Wf_{in} = 0$ for $n < \tau_i$.

Similarly, the batch size of task i that finishes at time point n, will be given by

$$Bf_{in} = Bs_{i,n-\tau_i} \qquad \forall\, i,\ \forall\, n \geq \tau_i \tag{34}$$

with $Bf_{in} = 0$ for $n < \tau_i$ and $Bs_{in} = 0$ for $n > N - \tau_i$.

Constraint 34 can also be derived from Eqs. 5–7 and 33.

Task i is processed at time point n if and only if it has started before n but after $n - \tau_i$; note that no two batches of task i can start after $n - \tau_i$ and before n because of constraints 1 and 33. This is expressed by the following constraint:

$$Bp_{in} = \sum_{n' \geq n-\tau_i+1}^{n' \leq n-1} Bs_{in'} \tag{35}$$

If we apply constraints 32–35 to model (M1) we can make the following simplifications.

Time ordering constraints

Constraints 22–24 can be replaced by Eq. 30; variables T_n become parameters as in (M2).

Assignment constraints

Constraint 26 of (M2) is obtained if we replace Wf_{in} from Eq. 33 into Eq. 1:

$$\sum_{i\in I(j)}\sum_{n'\le n}(Ws_{in'}-Ws_{in'-\tau_i})\le 1 \qquad \forall j,\ \forall n \ \rightarrow$$

$$\sum_{i\in I(j)}\sum_{n'\le n}Ws_{in'}-\sum_{i\in I(j)}\sum_{n'\le n}Ws_{in'-\tau_i}\le 1 \qquad \forall j,\ \forall n \ \rightarrow$$

$$\sum_{i\in I(j)}\sum_{n'\le n}Ws_{in'}-\sum_{i\in I(j)}\sum_{n'\le n-\tau_i}Ws_{in'}\le 1 \qquad \forall j,\ \forall n \ \rightarrow$$

$$\sum_{i\in I(j)}\sum_{n'\ge n-\tau_i+1}^{n'\le n}Ws_{in'}\le 1 \qquad \forall j,\ \forall n$$

Moreover, if no tasks that finish after H are allowed to start (that is $W_{sin} = 0$, $\forall n > N - \tau_i$), constraint 2 is trivially satisfied. Constraints 3 and 4 are weaker than constraint 26 because their left-hand side (LHS) is always smaller than the LHS of Eq. 26 and thus are trivially satisfied. Thus, constraints 1–4 of (M1) reduce to constraint 26 of (M2).

Batch-size constraints and material balances

Constraint 5 is the same as Eq. 27. If we replace Wf_{in} from Eq. 33 and Bf_{in} from Eq. 34 into Eq. 6 we obtain constraint 27 for $n \ge \tau_i$:

$$B_i^{MIN}Ws_{in-\tau_i}\le Bs_{in-\tau_i}\le B_i^{MAX}Ws_{in-\tau_i} \qquad \forall i,\ \forall n \ \rightarrow$$

$$B_i^{MIN}Ws_{in}\le Bs_{in}\le B_i^{MAX}Ws_{in} \qquad \forall i,\ \forall n\le N-\tau_i$$

Therefore, constraint 6 can be dropped.

If we substitute Eqs. 34 and 35 into Eq. 7 we obtain a constraint that is trivially satisfied:

$$Bs_{in-1}+\sum_{n'\ge(n-1)-\tau_i+1}^{n'\le(n-1)-1}Bs_{in'}=\sum_{n'\ge n-\tau_i+1}^{n'\le n-1}Bs_{in'}+Bs_{in-\tau_i} \qquad \forall i,\ \forall n \ \rightarrow$$

$$Bs_{in-1}+\sum_{n'\ge n-\tau_i}^{n'\le n-2}Bs_{in'}=\sum_{n'\ge n-\tau_i+1}^{n'\le n-1}Bs_{in'}+Bs_{in-\tau_i} \qquad \forall i,\ \forall n \ \rightarrow$$

$$\sum_{n'\ge n-\tau_i}^{n'\le n-1}Bs_{in'}=\sum_{n'\ge n-\tau_i}^{n'\le n-1}Bs_{in'} \qquad \forall i,\ \forall n$$

If we replace Bf_{in} from Eq. 34 into Eq. 9 and substitute Eqs. 8 and 9 into Eq. 10, we obtain Eq. 28 of (M2):

$$S_{sn}=S_{s,n-1}+\sum_{i\in O(s)}\rho_{is}Bf_{in}+\sum_{i\in I(s)}\rho_{is}Bs_{in}\le C_s \qquad \forall s,\ \forall n \ \rightarrow$$

$$S_{sn}=S_{s,n-1}+\sum_{i\in O(s)}\rho_{is}Bs_{in-\tau_i}+\sum_{i\in I(s)}\rho_{is}Bs_{in}\le C_s \qquad \forall s,\ \forall n$$

Thus, constraints 5–10 reduce to constraints 27 and 28 of model (M2).

Resource constraints

If we substitute Eqs. 11 and 12 (where R^O_{irn} is a function of Ws_{in} and Bs_{in} according to Eqs. 33 and 34) into Eq. 13, we obtain the following constraint:

$$R_{rn}=R_{rn-1}-\sum_i(\gamma_{ir}Wf_{in}+\delta_{irs}Bf_{in})+\sum_i(\gamma_{ir}Ws_{in}+\delta_{ir}Bs_{in})$$
$$\le R_i^{MAX} \qquad \forall r,\ \forall n \ \rightarrow$$

$$R_{rn}=R_{rn-1}-\sum_i(\gamma_{ir}Ws_{in-\tau_i}+\delta_{irs}Bs_{in-\tau_i})+\sum_i(\gamma_{ir}Ws_{in}$$
$$+\delta_{ir}Bs_{in})\le R_i^{MAX} \qquad \forall r,\ \forall n \ \rightarrow$$

$$R_{rn}=R_{rn-1}+\sum_i[\gamma_{ir}(Ws_{in}-Ws_{in-\tau_i})+\delta_{ir}(Bs_{in}-Bs_{in-\tau_i})]$$
$$\le R_i^{MAX} \qquad \forall r,\ \forall n \quad (36)$$

To calculate R_{rn} as a function of variables Ws_{in} and Bs_{in} only (as in constraint 29), we express constraint 36 for n, $n-1$, $n-2$, . . . , 1, and add them up:

$$R_{rn}=R_{rn-1}+\sum_i[\gamma_{ir}(Ws_{in}-Ws_{in-\tau_i})$$
$$+\delta_{ir}(Bs_{in}-Bs_{in-\tau_i})]\le R_i^{MAX} \qquad \forall r$$

$$R_{rn-1}=R_{rn-2}+\sum_i[\gamma_{ir}(Ws_{in-1}-Ws_{in-\tau_i-1})$$
$$+\delta_{ir}(Bs_{in-1}-Bs_{in-\tau_i-1})]\le R_i^{MAX} \qquad \forall r$$

$$R_{rn-2}=R_{rn-3}+\sum_i[\gamma_{ir}(Ws_{in-2}-Ws_{in-\tau_i-2})$$
$$+\delta_{ir}(Bs_{in-2}-Bs_{in-\tau_i-2})]\le R_i^{MAX} \qquad \forall r$$

$$R_{rn-3}=R_{rn-4}+\sum_i[\gamma_{ir}(Ws_{in-3}-Ws_{in-\tau_i-3})$$
$$+\delta_{ir}(Bs_{in-3}-Bs_{in-\tau_i-3})]\le R_i^{MAX} \qquad \forall r$$

. . .

$$R_{r1}=\sum_i(\gamma_{ir}Ws_{i1}+\delta_{ir}Bs_{i1})\le R_i^{MAX} \qquad \forall r$$

Variables $R_{r1}, R_{r2}, \ldots, R_{rn-2}, R_{rn-1}$ appear both on the LHS and the righthand side (RHS) of equations, and thus cancel out, giving

$$R_{rn}=\sum_i\left[\gamma_{ir}\left(\sum_{n'\le n}Ws_{in'}-\sum_{n''\le n-\tau_i}Ws_{in''}\right)\right.$$
$$\left.+\delta_{ir}\left(\sum_{n'\le n}Bs_{in'}-\sum_{n''\le n-\tau_i}Bs_{in''}\right)\right]\le R_i^{MAX} \qquad \forall r,\ \forall n$$

Terms $\gamma_{ir}Ws_{in'}$ and $\delta_{ir}Bs_{in'}$ with $n' < n - \tau_i - 1$ also cancel out:

$$R_{rn} = \sum_i \left(\gamma_{ir} \sum_{n' \geq n-\tau_i+1}^{n' \leq n} Ws_{in'} + \delta_{ir} \sum_{n' \geq n-\tau_i+1}^{n' \leq n} Bs_{in'} \right) \leq R_i^{MAX} \quad \forall\, r,\ \forall\, n$$

which is equivalent to constraint 29:

$$R_{rn} = \sum_i \sum_{n' \geq n-\tau_i+1}^{n' \leq n} (\gamma_{ir} Ws_{in'} + \delta_{ir} Bs_{in'}) \leq R_i^{MAX} \qquad \forall\, r,\ \forall\, n$$

Therefore, constraints 11–13 of (M1) reduce to constraint 29 of (M1).

Calculation of duration and finish time

Constant processing times imply that constraint 14 is replaced by Eq. 32. Constraints 15–17 are used to impose the condition that *if* $Ws_{in} = 1$, *then* $Tf_{in} = Ts_{in} + D_{in}$; *otherwise* $Tf_{in} = Tf_{in-1}$. In discrete-time models $D_{in} = \tau_i Ws_{in}$ and $Ts_{in} = T_n$, and thus these constraints are written:

$$Tf_{in} \leq T_n + \tau_i Ws_{in} + H(1 - Ws_{in}) \qquad \forall\, i,\ \forall\, n \quad (37)$$

$$Tf_{in} \geq T_n + \tau_i Ws_{in} - H(1 - Ws_{in}) \qquad \forall\, i,\ \forall\, n \quad (38)$$

$$\tau_i Ws_{in} \leq Tf_{in} - Tf_{in-1} \leq H Ws_{in} \qquad \forall\, i,\ \forall\, n \quad (39)$$

Note that in discrete-time models the finish time of a task always coincides with a time point because the time grid is fixed and the processing time is a multiple of Δt. Constraint 18 is redundant both for discrete- and continuous-time models and can be dropped if Ts_{in} is replaced by T_n in all constraints.

Activation of Wf_{in} binary variables

Constraints 19 and 20 (or 21 for tasks that produce zero-wait states) are used to enforce the following condition: *if a task that started before point n finishes between* T_{n-1} (*or* T_n *if* $i \in ZWI$) *and* T_n, *then* $Wf_{in} = 1$; that is, they are used to enforce that the "correct" Wf_{in} binary is one. When the time grid is fixed and the processing times are exact multiples of Δt, however, binary variables Wf_{in} are uniquely defined by Eq. 33, constraints 19–21 are trivially satisfied, and therefore they can be removed.

Reduced model

After the addition of constraints 32–35 and the reductions described above, the reduced continuous-time model (M3) consists of constraints 25–30, 32–35, and 37–39:

$$\sum_{i \in I(j)} \sum_{n' \geq n-\tau_i+1}^{n' \leq n} Ws_{in'} \leq 1 \qquad \forall\, j,\ \forall\, n \quad (26)$$

$$B_i^{MIN} Ws_{in} \leq Bs_{in} \leq B_i^{MAX} Ws_{in} \qquad \forall\, i,\ \forall\, n \quad (27)$$

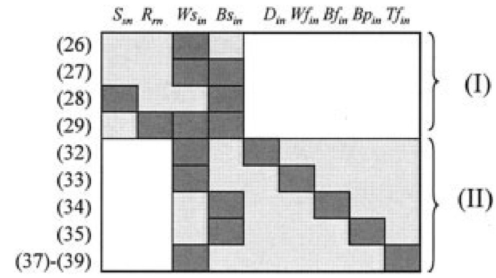

Figure 1. Incidence matrix of reduced model (M3).

$$S_{sn} = S_{s,n-1} + \sum_{i \in O(s)} \rho_{is} Bs_{in-\tau_i} + \sum_{i \in I(s)} \rho_{is} Bs_{in} \leq C_s \quad \forall\, s,\ \forall\, n \quad (28)$$

$$R_{rn} = \sum_i \sum_{n' \geq n-\tau_i+1}^{n' \leq n} (\gamma_{ir} Ws_{in'} + \delta_{ir} Bs_{in'}) \leq R_r^{MAX} \quad \forall\, r,\ \forall\, n \quad (29)$$

$$T_n = n\left(\frac{H}{N}\right) \qquad \forall\, n \quad (30)$$

$$D_{in} = \tau_i Ws_{in} \qquad \forall\, i,\ \forall\, n \quad (32)$$

$$Wf_{in} = Ws_{i,n-\tau} \qquad \forall\, i,\ \forall\, n \geq \tau \quad (33)$$

$$Bf_{in} = Bs_{i,n-\tau} \qquad \forall\, i,\ \forall\, n \geq \tau \quad (34)$$

$$Bp_{in} = \sum_{n' \geq n-\tau+1}^{n' \leq n-1} Bs_{in'} \quad (35)$$

$$Tf_{in} \leq T_n + \tau_i Ws_{in} + H(1 - Ws_{in}) \qquad \forall\, i,\ \forall\, n \quad (37)$$

$$Tf_{in} \geq T_n + \tau_i Ws_{in} - H(1 - Ws_{in}) \qquad \forall\, i,\ \forall\, n \quad (38)$$

$$\tau_i Ws_{in} \leq Tf_{in} - Tf_{in-1} \leq H Ws_{in} \qquad \forall\, i,\ \forall\, n \quad (39)$$

$$Ws_{in},\ Wf_{in} \in \{0, 1\},\ Bs_{in},\ Bp_{in},\ Bf_{in},\ S_{sn},\ T_n,\ Tf_{in},\ D_{in},\ R_{rn} \geq 0 \quad (25)$$

Discrete-time Formulation as a Special Case of Continuous-time Formulation

If we remove constraint 30, which includes only parameters T_n, the incidence matrix of the reduced model (M3) has the structure shown in Figure 1, where constraints 26–29 are grouped into subset (I), which is equivalent to discrete-time model (M2), and constraints 32–35 and 37–39 into subset (II).

There are three important observations regarding the incidence matrix of model (M3):

(1) Variables Wf_{in}, D_{in}, Bf_{in}, Bp_{in}, and Tf_{in} do not appear in any constraint of subset (I).

(2) Constraints 32–35 and 37–39 are used only for the

"calculation" of variables D_{in}, Wf_{in}, Bf_{in}, Bp_{in}, and Tf_{in}, respectively.

(3) For any feasible solution of subset (I), we can find variables Wf_{in}, D_{in}, Bf_{in}, Bp_{in}, and Tf_{in} that satisfy constraints 32–35 and 37–39.

Therefore, constraints 32–35 and 37–39, and variables Wf_{in}, D_{in}, Bf_{in}, Bp_{in}, and Tf_{in} do not affect the solution of subset (I); that is, the solution of model (M2). Thus, to solve model (M3) (that is, the reduced continuous-time formulation), we can solve subset (I) independently, and then use Eqs. 32–35 and 37–39 to calculate (redundant) variables D_{in}, Wf_{in}, Bf_{in}, Bp_{in}, and Tf_{in}. In other words, any solution of the reduced continuous-time model (M3) corresponds to a solution of the discrete-time model (M2), and any solution of the discrete-time model (M2) can be projected in the space of the variables of model (M3).

Therefore, we have shown that when the two restrictions of fixed time grid and fixed processing times are imposed on the continuous-time model (M1) of Maravelias and Grossmann,[10] it reduces into model (M3), which is equivalent to the discrete-time model (M2) of Shah et al.[3] The only difference between models (M2) and (M3) is that the additional variables D_{in}, Wf_{in}, Bf_{in}, Bp_{in}, and Tf_{in} are defined for the latter.

Remarks

For our analysis we considered a *basic* discrete-time STN formulation. In particular, we assumed that:

(1) There are no intermediate release/delivery times for the various states; that is, parameters R_{st} and D_{st} in Eq. 4 of Shah et al.[3] are equal to zero.

(2) The consumption of resources is constant during a processing task; that is, parameters $\alpha_{iu\theta}$ and $\beta_{iu\theta}$ in constraint 6 of Shah et al.[3] are equal to zero for $\theta \neq \tau_i$ and nonzero for $\theta = \tau_i$: $\alpha_{iu\tau_i} = \gamma_{ir}$ and $\beta_{iu\tau_i} = \delta_{ir}$, where indices u and r correspond to the same utility.

(3) The consumption/production of input/output states occurs only at the beginning/end of a processing task; that is, parameters $\rho_{is\theta}$ in constraint 15 of Kondili et al.[1] are equal to zero for $\theta \neq \tau_i$ and nonzero for $\theta = \tau_i$: $\rho_{is\tau_i} = \rho_{is}$.

The reason we made the preceding assumptions was that the purpose of the paper was to consider a discrete-time model that is equivalent, in terms of modeling, to most continuous-time models. In general, the modeling of *intermediate events* (such as the production of product states before the end of a processing task or the engagement of a utility after a processing task begins) is straightforward in discrete-time models but rather complex in continuous-time models, especially when the timing of such intermediate events is a function of the processing time of the corresponding task. To rigorously model these events using continuous-time models, additional binary variables are needed, leading to MILP formulations that are very hard to solve (for example, see Maravelias and Grossmann[12] for the general modeling of intermediate release and due dates). Thus, to our knowledge, there are no continuous-time models that can effectively address all the above *intermediate* events.

The remark of the previous paragraph raises questions on the accuracy and effectiveness of the two time representations. As explained in the introduction, processing times must often be approximated in discrete-time models, whereas they can be handled accurately by continuous-time models. However, to model intermediate release/due dates, semibatch processes and variable utility consumption using continuous-time models, additional binary variables and disjunctive constraints are needed. Moreover, to calculate holding, backlog, and utility consumption costs, bilinear terms are required, leading to MINLP formulations. Discrete-time models, on the other hand, handle intermediate events at practically no additional computational cost because no additional binary variables are needed, and holding, backlog, and utility consumption costs are handled linearly.

Conclusions

We have shown in this research note that the discrete-time model of Shah et al.[3] can be derived as a special case of the continuous-time model of Marevelias and Grossmann[10] when a uniform time grid is used with constant processing times. Aside from being a result of theoretical interest, one interesting implication of this relationship is that one might envisage applying a solution method developed for one representation to the other. An example would be to apply the hybrid MILP/constraint programming method by Maravelias and Grossmann[13] for continuous-time problems to discrete-time representations. This will be the subject of future research work.

Notation

Indices

i = tasks
j = equipment units
r = resource categories (utilities)
s = states
n = time points

Sets

$I(j)$ = set of tasks that can be scheduled on equipment j
$I(s)$ = set of tasks that use state s as input
$O(s)$ = set of tasks that produce state s
ZWI = set of tasks that produce at least one ZW state

Parameters

H = time horizon
α_i/τ_i = fixed duration of task i
β_i = variable duration of task i
γ_{ir} = fixed amount or utility r required for task i
δ_{ir} = variable amount of utility r required for task i
ρ_{is} = mass balance coefficient for the consumption/production of state s in task i
C_s = storage capacity for state s
R_r^{MAX} = upper bound for utility r
B_i^{MIN}/B_i^{MAX} = lower/upper bounds on the batch size of task i

Binary variables

Ws_{in} = 1 if task i starts at time point n
Wf_{in} = 1 if task i finishes at or before time point n

Continuous variables

T_n = time that corresponds to time point n
Ts_{in} = start time of task i that starts at time point n
Tf_{in} = finish time of task i that starts at time point n
D_{in} = duration of task i that starts at time point n
Bs_{in} = batch size of task i that starts at time point n
Bp_{in} = batch size of task i that is processed at time point n
Bf_{in} = batch size of task i that finishes at or before time point n

B^I_{isn}/B^O_{isn} = amount of input/output state s for task i at time point n
S_{sn} = amount of state s available at time point n
R^I_{irn} = amount of utility r consumed at time point n by task i
R^O_{irn} = amount of utility r released at or before time point n by task i
R_{rn} = amount of utility r utilized at time point n

Literature Cited

1. Kondili E, Pantelides CC, Sargent R. A general algorithm for short-term scheduling of batch operations—I. MILP formulation. *Comput Chem Eng.* 1993;17:211-227.
2. Pantelides CC. Unified frameworks for the optimal process planning and scheduling. Proc of the 2nd Conference on Foundations of Computer Aided Operations; 1994:253-274.
3. Shah NE, Pantelides CC, Sargent R. A general algorithm for short-term scheduling of batch operations—II. Computational issues. *Comput Chem Eng.* 1993;17:229-244.
4. Schilling G, Pantelides CC. A simple continuous-time process scheduling formulation and a novel solution algorithm. *Comput Chem Eng.* 1996;20:S1221-S1226.
5. Ierapetritou MG, Floudas CA. Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. *Ind Eng Chem Res.* 1998;37:4341-4359.
6. Mockus L, Reklaitis GV. Continuous time representation approach to batch and continuous process scheduling. 1. MINLP formulation. *Ind Eng Chem Res.* 1999;38:197-203.
7. Castro P, Barbosa-Povoa APFD, Matos H. An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. *Ind Eng Chem Res.* 2001;40:2059-2068.
8. Giannelos NF, Georgiadis MC. A simple new continuous-time formulation for short-term scheduling of multipurpose batch processes. *Ind Eng Chem Res.* 2002;41:2178-2184.
9. Lee KH, Park HI, Lee IB. A novel nonuniform discrete time formulation for short-term scheduling of batch and continuous processes. *Ind Eng Chem Res.* 2001;40:4902-4911.
10. Maravelias CT, Grossmann IE. A new general continuous-time state task network formulation for the short-term scheduling of multipurpose batch plants. *Ind Eng Chem Res.* 2003;42:3056-3074.
11. Sundaramoorthy A, Karimi IA. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. *Chem Eng Sci.* 2005;60:2679-2702.
12. Maravelias CT, Grossmann IE. A continuous-time state task network formulation for the short-term scheduling of multipurpose batch plants with due dates. In: Chen B, Westerberg AW, eds. *Proceedings of PSE 2003*. Amsterdam, The Netherlands: Elsevier; 2004.
13. Maravelias CT, Grossmann IE. A hybrid MILP/CP decomposition approach for the short term scheduling of multipurpose batch plants. *Comput Chem Eng.* 2004;28:1921-1949.

Manuscript received Nov. 13, 2004, and revision received Aug. 6, 2005.