On the Relation of Continuous and Discrete Time Models for the State-Task Network Formulation

Christos T. Maravelias[1] and Ignacio E. Grossmann[2]*
[1] Department of Chemical and Biological Engineering, University of Wisconsin, Madison, WI 53706
[2] Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

Abstract
The relationship between discrete- and continuous-time State Task Network (STN) representations is studied. We show that the continuous-time model of Maravelias and Grossmann (2003) is a generalization of the discrete-time model of Shah et al. (1993). More specifically, discrete-time models can be shown to be a special case of continuous-time models if two restrictions are imposed on the latter: a) fixed time grid, and b) processing times fixed multiples of the duration of the time period of the time grid.

Keywords: Scheduling, State Task Network, Time Representations

1. Introduction

The purpose of this research note is to formally study the relationship between discrete-time and continuous-time State Task Network (STN) MIP formulations. The STN (Kondili et al., 1993) and its equivalent Resource Task Network (RTN) (Pantelides, 1994) representation were proposed as general formulations for the scheduling of complex plants. In the original discrete-time formulation (Kondili et al., 1994; Shah et al., 1993), 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. The disadvantage of discrete-time models 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 (Schilling and Pantelides, 1996; Ierapetritou and Floudas, 1998; Mockus and Reklaitis, 1999; Castro et al., 2001; Lee et al., 2001; Giannelos and Georgiadis, 2002; Maravelias and Grossamnn, 2003). 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, since time points are not fixed, big-M 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, and an expensive iterative procedure is needed.

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

Since 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

[*] Author to whom correspondence should be addressed: E-mail: grossmann@cmu.edu, Tel.: 412-268-3642, Fax: 412-268-7139

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. The models of Maravelias and Grossmann (2003) and Shah et al. (1993) are used as bases of the analysis and are presented in sections 2 and 3, respectively. The derivation of the model of Shah et al. (1993) from the model of Maravelias and Grossmann (2003) is presented in section 4.

2. Continuous-time STN Model

A compact form of the continuous-time model of Maravelias and Grossmann (2003) 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.

We describe model (M1) in the next section. 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. 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 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 benoted by B^I_{isn} (B^O_{isn}). The details and derivation of the proposed model can be found in Maravelias and Grossmann (2003).

2.1. 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, while constraints (3) and (4) enforce the condition that not more than one task can start or finish on a specific unit at any time:

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

$$\sum_n Ws_{in} = \sum_n Wf_{in} \quad \forall i \tag{2}$$

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

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

2.2. Batch-size Constraints and Material Balances

Constraints (5) and (6) impose upper and lower bounds on the batch sizes Bs_{in} and Bs_{in}, while 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 constraitns (8) and (9), respectively, where

$O(s)/I(s)$ is the set of tasks producing/consuming state s, and ρ_{is} is the mixing ratio of state s. Constraint (10) is the mass balance and the capacity constraint for state s at time n, where C_s is the storage capacity:

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

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

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

$$B_{isn}^{I} = \rho_{is} Bs_{in} \quad \forall i, \forall n, \forall s \in SI(i) \tag{8}$$

$$B_{isn}^{O} = \rho_{is} Bf_{in} \quad \forall i, \forall n, \forall s \in SO(i) \tag{9}$$

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

2.3. 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 (13):

$$R_{irn}^{I} = \gamma_{ir} Ws_{in} + \delta_{ir} Bs_{in} \quad \forall i, \forall r, \forall n \tag{11}$$

$$R_{irn}^{O} = \gamma_{ir} Wf_{in} + \delta_{ir} Bf_{in} \quad \forall i, \forall r, \forall n \tag{12}$$

$$R_{rn} = R_{rn-1} - \sum_{i} R_{irn-1}^{O} + \sum_{i} R_{irn}^{I} \le R_i^{MAX} \quad \forall r, \forall n \tag{13}$$

2.4. 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} via constraint (17). The elimination of start times, Ts_{in}, is accomplished through constraint (18), where H is the scheduling horizon.

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

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

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

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

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

2.5. 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 (20)] if it produces a state for which zero-wait policy applies, where *ZWI* is the subset of tasks that produce a zero-wait state:

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

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

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

2.6. Time ordering

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

$$T_{n=1} = 0 \tag{22}$$

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

$$T_{n+1} \ge T_n \quad \forall n \tag{24}$$

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

$$Ws_{in},\ Wf_{in} \in \{0,1\},\ Bs_{in},\ Bp_{in},\ Bf_{in},\ S_{sn},\ T_n,\ Tf_{in},\ D_{in},\ B^I{}_{isn},\ B^O{}_{isn}, R^I{}_{irn},\ R^O{}_{irn},\ R_{rn} \ge 0 \tag{25}$$

3. Discrete-time STN Model

The discrete-time model (M2) of Shah et al. (1993) 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. (1993)). Constraint (28) is the material balance equation and capacity constraint for state *s* at time *n*, and constraint (29) is the utility constraint. Time points T_n are fixed parameters in discrete-time models, but we have included constraint (30) for comparison with continuous-time models.

$$\sum_{i \in I(j)} \sum_{n' \ge n-\tau+1}^{n' \le n} Ws_{in'} \le 1 \quad \forall j, \forall n \tag{26}$$

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

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

$$R_{rn} = \sum_i \sum_{n' \ge n-\tau}^{n' \le n} \left(\gamma_{ir} Ws_{in'} + \delta_{ir} Bs_{in'} \right) \le R_r^{MAX} \quad \forall r, \forall n \tag{29}$$

$$T_n = n\left(H \big/ \left(|N| - 1 \right) \right) \quad \forall n \tag{30}$$

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

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

4. Discrete-time as a Special Case of 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:
A) the time grid is fixed, and
B) the processing times of all tasks are constant multiples of the interval of the fixed time grid.

4.1. Assumptions

The assumption of fixed time grid with equal time intervals means that variables T_n of model (M1) are fixed, and given by equation (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 is zero. Thus, D_{in} is always given by:

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

Constraint (32) can also be derived from (14) for $\alpha_i = \tau_i$ and $\beta_i = 0$.
Since 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, binary Wf_{in} is always given by:

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

Moreover, since task i cannot finish before τ_i, we 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:

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

and $Bf_{in}=0$ for $n<\tau_i$.
Constraint (34) can also be derived from (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, due to constraints (1) and (33)). This is expressed via constraint (35):

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

A description of the reductions that can be applied to model (M1) if we apply constraints (32) – (35) follows.

4.2. Derivation

4.2.1. Time Ordering

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

4.2.2. Assignment Constraints

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

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

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

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

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

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

4.2.3. Batch-size Constraints and Material Balances

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

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

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

If we substitute (34) and (35) into (7) we obtain a constraint that is trivially satisfied:

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

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

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

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

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

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

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

4.2.4. Resource Constraints

If we substitute (11) and (12), where R^O_{irn} is a function of Ws_{in} and Bs_{in} due to (33) and (34), into (13) we obtain constraint (36):

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

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

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

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:

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

$$R_{rn-1} = R_{rn-2} + \sum_i \left(\gamma_{ir} (Ws_{in-1} - Ws_{in-\tau_i-2}) + \delta_{ir} (Bs_{in-1} - Bs_{in-\tau_i-2}) \right) \le R_i^{MAX} \quad \forall r, \forall n$$

$$R_{rn-2} = R_{rn-3} + \sum_i \left(\gamma_{ir} (Ws_{in-2} - Ws_{in-\tau_i-3}) + \delta_{ir} (Bs_{in-2} - Bs_{in-\tau_i-3}) \right) \le R_i^{MAX} \quad \forall r, \forall n$$

$$R_{rn-3} = R_{rn-4} + \sum_i \left(\gamma_{ir} (Ws_{in-3} - Ws_{in-\tau_i-4}) + \delta_{ir} (Bs_{in-3} - Bs_{in-\tau_i-4}) \right) \le R_i^{MAX} \quad \forall r, \forall n$$

…

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

Variables R_{r1}, R_{r2}, ... R_{rn-2}, R_{rn-1} cancel out, giving:

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

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

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

which is equivalent to constraint (29):

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

Constraints (11) – (13) of (M1), therefore, reduce to constraint (29) of (M1).

4.2.5. Calculation of Duration and Finish Time

Constant processing times imply that constraint (14) is replaced by (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:

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

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

$$\tau_i Ws_{in} \le Tf_{in} - Tf_{in-1} \le H Ws_{in} \quad \forall i, \forall n \tag{39}$$

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.

4.2.6. 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 $Wf_{in-1}=1$*, i.e. they are used to enforce that the "correct" Wf_{in} binary is one. In discrete-time models, however, binary variables Wf_{in} are uniquely defined by (33), constraints (19) – (21) are trivially satisfied, and therefore can be removed.

4.3. Reduced Model

After the addition of constraints (31) – (34) and the reductions described above, the reduced model (M3) consists of constraints (25) – (30), (32) – (35) and (37)-(39):

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

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

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

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

$$T_n = n\left(H \big/ (|N|-1)\right) \quad \forall n \tag{30}$$

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

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

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

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

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

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

$$\tau_i Ws_{in} \leq Tf_{in} - Tf_{in-1} \leq HWs_{in} \quad \forall i, \forall n \tag{39}$$

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

If we remove constraint (30), that 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 model (M1), and constraints (32) – (35) and (37) – (39) into subset (II).

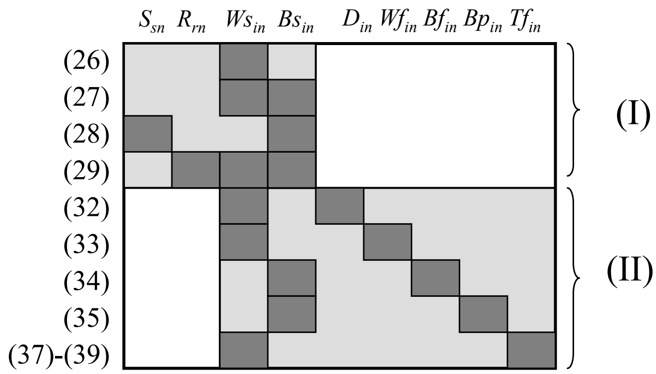

Figure 1: Incidence matrix of reduced model (M3).

There are two important observations:

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

Constraints (32) – (35) and (37) – (39), and variables Wf_{in}, D_{in}, Bf_{in}, Bp_{in} and Tf_{in}, therefore, do not affect the solution of subset (I), i.e. the solution of model (M2). A closer look at constraints (32) – (35) and (37) – (39) reveals that these are used only for the "calculation" of variables D_{in}, Wf_{in}, Bf_{in}, Bp_{in} and Tf_{in}, respectively. In other words, instead of solving model (M3) as is, we can solve subset (I) independently, and then use equations (32) – (35) and (37) – (39) to calculate variables D_{in}, Wf_{in}, Bf_{in}, Bp_{in} and Tf_{in}.

Hence, we have showed 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 (2003), it reduces into model (M3), which is equivalent to the discrete-time model (M2) of Shah et al. (1993). 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.

4. Conclusions

We have shown in this research note that the discrete-time model of Shah et al. (1993) can be derived as a particular case of the continuous time model of Marevelias and Grossmann (2003) when a uniform time grid is used with constant processing times. Aside from being of 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 (2004) for continuous time problems to discrete time representations.

Nomenclature

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*
$S0_s$	Initial amount of state *s*
C_s	Storage capacity for state *s*
R^{MAX}_r	Upper bound for utility *r*
B^{MIN}_i / B^{MAX}_i	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 time point *n*
Continuous Variables	
T_n	Time that corresponds to time point *n* (i.e. start of period *n*; finish of period *n-1*)
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*

References

Castro, P.; Barbosa-Povoa, A. P. F. D.; 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.

Giannelos, N.F.; Georgiadis, M.C. A Simple New Continuous-Time Formulation for Short-Term Scheduling of Multipurpose Batch Processes. *Ind. Eng. Chem. Res.* **2002**, 41, 2178-2184.

Ierapetritou, M. G.; Floudas, C. A. Effective Continuous-Time Formulation for Short-Term Scheduling. 1. Multipurpose Batch Processes. *Ind. Eng. Chem. Res.* **1998**, 37, 4341-4359.

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, 211-227.

Lee, K.; Il Park, H.; In Beum Lee. A Novel Nonuniform Discrete Time Formulation for Short-Term Scheduling of Batch and Continuous Processes. *Ind. Eng. Chem. Res.* **2001**, 40, 4902-4911.

Maravelias, C.T.; Grossmann, I.E. A New General Continuous-Time State Task Network Formulation for the Short-Term Scheduling of Multipurpose Batch Plants. *Ind. Eng. Chem. Res.*, **2003**, 42(13), 3056-3074.

Maravelias, C.T.; Grossmann, I.E. A Hybrid MILP/CP Decomposition Approach for the Short Term Scheduling of Multipurpose Batch Plants. *Comput. Chem. Eng.* **2004**, 28, 1921-1949.

Mockus, L.; Reklaitis, G.V. Continuous Time Representation Approach to Batch and Continuous Process Scheduling. 1. MINLP Formulation. *Ind. Eng. Chem. Res.* **1999**, 38, 197-203.

Pantelides, C. C. Unified Frameworks for the Optimal Process Planning and Scheduling. Proceedings on the Second Conference on Foundations of Computer Aided Operations. **1994**, 253-274.

Shah, N.; E.; Pantelides, C. C.; Sargent, R. A General Algorithm for Short-Term Scheduling of Batch Operations – II. Computational Issues. *Comput. Chem. Eng.* **1993**, 17, 229-244.

Schilling, G.; Pantelides, C. C. A Simple Continuous-Time Process Scheduling Formulation and a Novel Solution Algorithm. *Comput. Chem. Eng.* **1996**, 20, S1221-1226.