European Symposium on Computer Aided Process Engineering – 15
L. Puigjaner and A. Espuña (Editors)


# On the State-Task Network: Time Representations

Christos T. Maravelias*

Department of Chemical & Biological Engineering, University of Wisconsin-Madison
1415 Engineering Drive, Madison, WI 53706, USA

## Abstract

To model and solve complex Supply Chain problems we study the relationship between the discrete- and continuous-time State-Task Network (STN) representations. We show that the first is a special case of the second. We also propose a new mixed-time representation where the time grid is fixed but processing times are allowed to be variable and span an unknown number of time periods. The proposed scheme accounts linearly for holding and backlog costs, can be extended to handle continuous processes, accounts for due dates at no additional computational cost, and handles variable processing times; i.e. it combines the modelling advantages of both discrete- and continuous-time STN models. Finally, we present strong cutting planes that are used to enhance the solution of the proposed MIP model.

**Keywords:** Scheduling, State Task Network, Time Representations

## 1. Introduction

The purpose of the paper is to present a time representation for STN models that can be readily used for Supply Chain (SC) optimization. Issues usually neglected in standalone scheduling models become important when scheduling is considered within SC:

- Smooth demand patters at the customer-facing nodes are transformed into high demand peaks at the production facilities, due to the various inventory policies applied at the intermediate nodes (warehouses, distribution centres) to account for demand uncertainty.
- Demand cannot always be satisfied and backlog costs are introduced to account for unmet demand and/or late deliveries.
- Shipments are usually made at fixed time intervals which means that scheduling models have to handle multiple intermediate due dates at low computational cost.
- Features such as seasonality of demand, low volume products with large minimum batch sizes, scheduled maintenance and shut-downs require long planning horizons. Thus, the lot-sizing problem (holding vs. changeover costs) becomes very important.

To optimize the SC of process industries, therefore, current planning and scheduling methods should be amended to simultaneously account for lot-sizing, holding, backlog, changeover and shipment costs and multiple due dates, while solution methods should be enhanced to effectively solve problems over long time-horizons. The State-Task Network (STN) formulation (Kondili et al., 1993) and the equivalent Resource Task

* maravelias@wisc.edu

Network (RTN) formulation (Pantelides, 1994) are two powerful modelling tools that have been refined by several researchers (Schilling and Pantelides, 1996; Ierapetritou and Floudas, 1999; Mockus and Reklaitis, 1999; Castro et al., 2001; Maravelias and Grossamnn, 2003; Sundaramoorthy and Karimi, 2005) and extensively used for planning and scheduling problems. Nevertheless, continuous-time STN models do not account linearly for holding and backlog costs and are computationally expensive for the solution of problems with due dates, while discrete-time STN models are computationally expensive for operations with variable processing times. In this paper, we rigorously study the relationship between the two representations, and explore whether we can circumvent their limitations and combine their advantages.

## 2. Discrete-time Models: a Special Case of Continuous-time Models

A compact form of the continuous-time model (M1) of Maravelias and Grossmann (2003) in Table 1, and the discrete-time model (M2) of Shah et al. (1993) in Table 2 are used as the basis of our analysis. Indices *i*, *j*, *s*, *n* and *r* are used to denote a task, unit, state, time point (or period) and shared resource, respectively.

*Table 1. Continuous-time STN model (M1) of Maravelias and Grossmann (2003).*

| | | | |
|---|---|---|---|
| Time Grid | | Utility Constraints | |
| $T_0 = 0, \quad T_N = H$ | (1) | $Rs_{irn} = \gamma_{ir} Ws_{in} - \delta_{irs} Bs_{in} \quad \forall i, \forall r, \forall n$ | (9) |
| $T_{n+1} \geq T_n \quad \forall n$ | (2) | $Rf_{irn} = \gamma_{ir} Wf_{in} + \delta_{irs} Bf_{in} \quad \forall i, \forall r, \forall n$ | (10) |
| Assignment Constraints | | $R_{rn} = R_{rn-1} - \sum_i Rf_{irn} + \sum_i Rs_{irn} \leq R_r^{MAX} \quad \forall r, \forall n$ | (11) |
| $\sum_{i \in I(j)} (\sum_{n' \leq n} Ws_{in'} - \sum_{n' < n} Wf_{in'}) \leq 1 \quad \forall j, \forall n$ | (3) | Calculation of duration $D_{in}$ and finish time $Tf_{in}$ | |
| $\sum_n Ws_{in} = \sum_n Wf_{in} \quad \forall i$ | (4) | $D_{in} = \alpha_i Ws_{in} + \beta_i Bs_{in} \quad \forall i, \forall n$ | (12) |
| Batch-size and Mass Balance Constraints | | $Tf_{in} \leq T_n + D_{in} + H(1 - Ws_{in}) \quad \forall i, \forall n$ | (13) |
| $B_i^{MIN} Ws_{in} \leq Bs_{in} \leq B_i^{MAX} Ws_{in} \quad \forall i, \forall n$ | (5) | $Tf_{in} \geq T_n + D_{in} - H(1 - Ws_{in}) \quad \forall i, \forall n$ | (14) |
| $B_i^{MIN} Wf_{in} \leq Bf_{in} \leq B_i^{MAX} Wf_{in} \quad \forall i, \forall n$ | (6) | Activation of binary variable $Wf_{in}$ * | |
| $Bs_{in-1} + Bp_{in-1} = Bp_{in} + Bf_{in} \quad \forall i, \forall n$ | (7) | $Tf_{in-1} \leq T_n + H(1 - Wf_{in}) \quad \forall i, \forall n$ | (15) |
| $S_{sn} = S_{s,n-1} + \sum_{i \in O(s)} \rho_{is} Bf_{in} + \sum_{i \in I(s)} \rho_{is} Bs_{in} \leq C_s \quad \forall s, \forall n$ | (8) | $Tf_{in-1} \geq T_{n-1} - H(1 - Wf_{in}) \quad \forall i, \forall n$ | (16) |
| | | * Different constraints hold for zero-wait storage | |

*Table 2. Discrete-time STN model (M2) of Shah et al. (1993).*

| | |
|---|---|
| $T_n = n\left(H/N\right) = n\Delta t \quad \forall n$ | (17) |
| $\sum_{i \in I(j)} \sum_{n' \geq n - \tau + 1}^{n' \leq n} Ws_{in'} \leq 1 \quad \forall j, \forall n$ | (18) |
| $B_i^{MIN} Ws_{in} \leq Bs_{in} \leq B_i^{MAX} Ws_{in} \quad \forall i, \forall n$ | (19) |
| $S_{sn} = S_{s,n-1} + \sum_{i \in O(s)} \rho_{is} Bs_{in-\tau_i} - \sum_{i \in I(s)} \rho_{is} Bs_{in} \leq C_s \quad \forall s, \forall n$ | (20) |
| $R_{rn} = \sum_i \sum_{n' \geq n - \tau}^{n' \leq n} \left(\gamma_{ir} Ws_{in'} + \delta_{ir} Bs_{in'}\right) \leq R_r^{MAX} \quad \forall r, \forall n$ | (21) |

In model (M1), binary $Ws_{in}$ ($Wf_{in}$) is equal to one if task *i* starts at (finishes at or before) time point $n \in \{0, 1, ..N\}$. 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 processing time of task $i$ is denoted by $D_{in}$. The start time 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 same variables $Ws_{in}$, $Bs_{in}$, $S_{sn}$ and $R_{rn}$ are used in model (M2) for uniformity.

Model (M2) is obtained from model (M1) if A) the time grid is fixed, and B) the processing times of all tasks are constant multiples of the interval $\Delta t=H/N$ of the fixed time grid. The implications of restrictions A and B are shown in Table 3.

*Table 3. Restrictions applied to (M1).*

| Restriction | Implication | Constraint | |
|---|---|---|---|
| **A** | Fixed time points | $T_n = n\Delta t \quad \forall n$ | (17) |
| **B** | Constant processing times | $D_{in} = \tau_i Ws_{in} \quad \forall i, \forall n$ | (22) |
| | Binary $Wf_{in}$ function of $Ws_{in}$ | $Wf_{in} = Ws_{i,n-\tau_i} \quad \forall i, \forall n \geq \tau_i$ | (23) |
| | Variable $Bf_{in}$ function of $Bs_{in}$ | $Bf_{in} = Bs_{i,n-\tau_i} \quad \forall i, \forall n \geq \tau_i$ | (24) |
| | Variable $Bf_{in}$ function of $Bs_{in}$ | $Bp_{in} = \sum_{n' \geq n-\tau_i+1}^{n' \leq n-1} Bs_{in'} \quad \forall i, \forall n$ | (25) |

An outline of the reductions that can be applied to model (M1) follows:

1. If the time grid is fixed, equations (1)-(2) are replaced by eq. (17).
2. If we replace $Wf_{in}$ from eq. (23) into eq. (3), we obtain eq. (18) of (M2). If no tasks that finish after $H$ are allowed to start, eq. (4) is trivially satisfied due to eq. (23).
3. Eq. (5) is the same as eq. (19). If we replace $Wf_{in}$ from eq. (23) into eq. (6), we obtain eq. (5) for $n \geq \tau_i$; i.e. eq. (6) is removed. If we plug eq. (24) and (25) into eq. (7) we obtain a constraint that is trivially satisfied, and hence dropped. By replacing $Bf_{in}$ from eq. (24) into eq. (8) we obtain eq. (20) of model (M2). Constraints (5)-(8) of (M1), therefore, are equivalent to constraints (19)-(20) of (M2).
4. If we replace $Rs_{irn}$ from eq. (9) and $Rf_{irn}$ from eq. (10) (which is a function of $Ws_{in}$ and $Bs_{in}$ due to equations (23) and (24)) into eq. (11) we obtain eq. (21) of (M2).
5. Constant processing times imply that eq. (12) is replaced by eq. (22). Constraints (13) and (14) are used to enforce the condition that *if* $Ws_{in}=1$ *then* $Tf_{in}=Ts_{in}+D_{in}$, *otherwise unconstrained*, and cannot be simplified further.
6. Constraints (15) and (16) are used to enforce the following condition: *if a task that started before point n finishes between* $T_{n-1}$ *and* $T_n$, *then* $Wf_{in}=1$. Variable $Wf_{in}$, however, is uniquely defined by eq. (23), and constraints (15) and (16) are trivially satisfied, and therefore removed.

The reduced model (M1*) consists of equations (17)-(21) of model (M2) and equations (13)-(14) and (22)-(25). Note that variables $Wf_{in}$, $D_{in}$, $Bf_{in}$, $Bp_{in}$ and $Tf_{in}$ do not appear in equations (17)-(21), and that for any feasible solution of model (M2) (i.e. equations (17)-(21)), we can find values of variables $Wf_{in}$, $D_{in}$, $Bf_{in}$, $Bp_{in}$ and $Tf_{in}$ that satisfy constraints (13)-(14) and (22)-(25). Moreover, a solution of (M1*) corresponds to a unique solution of (M2). In other words, the feasible regions of model (M2) and the reduced model (M1*) in the space of variables $Ws_{in}$, $Bs_{in}$, $S_{sn}$ and $R_{rn}$ (i.e. when variables $Wf_{in}$, $D_{in}$, $Bf_{in}$, $Bp_{in}$ and $Tf_{in}$ are projected out) are exactly the same. The discrete-time model (M2), therefore, is equivalent to the reduced model (M1*), which means that model (M2) is a special case of the continuous-time model (M1) when restrictions A and B are applied.

## 3. A New Time Representation

In the previous paragraph, we showed that the assumption of the fixed-time grid is not necessarily coupled with the assumption of constant processing times. This allows us to develop a new time representation, where the time grid is fixed but processing times can be variable. Tasks are forced to start exactly at a time point but are allowed to span an unknown number of time intervals and finish anywhere within a time interval. The proposed MIP model (M3) consists of equations (3)-(17) and has a number of modelling advantages:

- Compared to (M1), it accounts linearly for holding and backlog costs.
- Compared to (M1), it accounts for due dates at no additional computational cost.
- Compared to discrete-time models, it does not require the introduction of additional binary variables for the modelling of tasks with variable processing times.
- It can be readily extended to account for continuous processes, with very good computational results (not presented in this paper due to space limitations).

The only modelling drawback of model (M3) is that it cannot handle zero-wait storage policy between two batch processes. Zero-wait policy, however, is usually applied between two continuous processes, and this case is effectively handled.

## 4. Computational Improvements

Model (M3) appears to "carry the computational curses" of both discrete- and continuous-time representations (i.e. large number of time intervals and poor LP-relaxation gaps). Nevertheless, the fact that the time points are fixed allows us to apply several computational improvements that greatly enhance the solution of model (M3). Note that similar improvements cannot always be applied to general continuous-time models. The most important enhancements are:

1. Tightening of big-M parameters: The inequality $Tf_{in} \geq T_n$ is always valid, which means that eq. (16) can be removed. We can also use $H=0$ in eq. (14). Moreover, we can use $H=D_i^{MAX}-\Delta t$ in eq. (13) and in eq. (15), where $D_i^{MAX}$ $(=\alpha_i+\beta_i B_i^{MAX})$ is the maximum processing time of task *i*.
2. The minimum $N_i^{MIN}$ and maximum $N_i^{MAX}$ number of time intervals a task *i* can span is determined by $N_i^{MIN}=\lceil D_i^{MIN}/\Delta t \rceil$ and $N_i^{MAX}=\lceil D_i^{MAX}/\Delta t \rceil$, where $D_i^{MIN}$ $(=\alpha_i+\beta_i B_i^{MIN})$ is the minimum duration of task *i*. This allows us to generate the valid inequalities of Table 4 that tighten the formulation. Additional valid inequalities can be developed and used in a branch-and-cut algorithm.

*Table 4. Valid inequalities added to model (M3).*

| | | | |
|---|---|---|---|
| $Ws_{in} \leq \sum_{\theta \geq N_i^{MINi}}^{\theta \leq N_i^{MAX}} Wf_{in+\theta} \quad \forall i, \forall n$ | (26) | $Wf_{in} \leq \sum_{\theta \geq N_i^{MINi}}^{\theta \leq N_i^{MAX}} Ws_{in-\theta} \quad \forall i, \forall n$ | (27) |
| $\sum_{i \in I(j)} \sum_{n' > n - N_i^{MIN}}^{n' \leq n} Ws_{in'} \leq 1 \quad \forall j, \forall n$ | (28) | $\sum_{n' \leq n} Wf_{in'} \leq \sum_{n'' \leq n - N_i^{MIN}} Ws_{in''} \quad \forall i, \forall n$ | (29) |

3. Finally, the balance constraints of Sundaramoorthy and Karimi (2005) can be used for the calculation of the remaining processing time on an equipment unit, instead of big-M constraints (13)-(16).

## 5. Example and Computational Results

Model (M3) is used for the scheduling of the STN in Figure 1 (modified from Kondili et al., 1993; data given in Table 5). The scheduling horizon is 10 hours and there are three orders for 20 kg of P1 each (with intermediate due dates at $t$ = *4*, *6* and *8* hr). The objective is to meet the orders on time and maximize the revenues from additional sales.

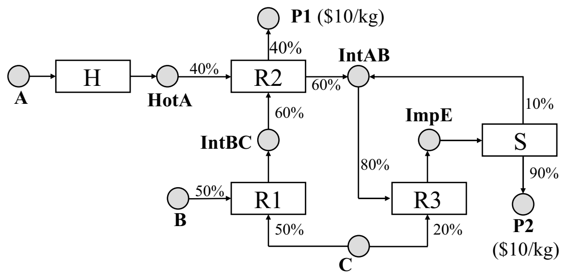


*Figure 1. State-Task Network.*

*Table 5. Processing data ($B^{MIN}/B^{MAX}$ in kg, α in hr, β in hr/kg).*

| | Task → | | H | | R1 | | R2 | | R3 | | S | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Unit | $B^{MIN}$ | $B^{MAX}$ | $\alpha$ | $\beta$ | $\alpha$ | $\beta$ | $\alpha$ | $\beta$ | $\alpha$ | $\beta$ | $\alpha$ | $\beta$ |
| HT | 50 | 100 | 1 | 0 | - | - | - | - | - | - | - | - |
| RI | 25 | 50 | - | - | 2 | 0 | .5 | .04 | .25 | .02 | - | - |
| RII | 40 | 80 | - | - | 2 | 0 | .5 | .025 | .25 | .0125 | - | - |
| DC | 100 | 200 | - | - | - | - | - | - | - | - | .5 | .01 |

To model intermediate due dates using a continuous-time model we have to introduce additional binary variables. In model (M3), however, we only introduce a (demand) parameter in eq. (8). The example is solved using both the continuous-time model (M1) with $N$=*8*, *9* and *10*, and model (M3) with $\Delta t$=*0.5*, *0.33* and *0.25* hr (solution statistics given in Table 6). Model (M1) cannot be solved to optimality within 600 CPU sec, and the best solution has an objective value of $1,797.3 (found with $N$=*8*). However, a solution with an objective value of $1,805.4 is obtained using model (M3) with $\Delta t$= *0.25hr* ($N$ = *40*) in 542.9 CPU sec. Model (M2), in other words, yields a ***better solution in less time*** than an "accurate" continuous-time model. Moreover, good solutions are also obtained with $\Delta t$= *0.5* ($1,720) and $\Delta t$= *0.33* ($1,763.3) in less than two minutes. The Gantt chart of the optimal solution of (M3) with $\Delta t$ = *0.25* hr is shown in Figure 2.

*Table 6. Computational results of models (M1) and (M3) (Δt in hours).*

| | (M1) | | | (M3) | | |
|---|---|---|---|---|---|---|
| | $N$=8 | $N$=9 | $N$=10 | $\Delta t$=0.5 | $\Delta t$=0.33 | $\Delta t$=0.25 |
| LP-rel. | 2,498.9 | 2,539.0 | 2,561.4 | 2,081.4 | 2,054.7 | 2,095.9 |
| Objective | 1,797.3[1] | 1,783.6[1] | 1,788.4[1] | 1,720.0 | 1,763.3 | 1,805.3 |
| CPU-sec[2] | 600 | 600 | 600 | 43.1 | 76.3 | 542.9 |
| Nodes | 114,651 | 71,106 | 65,620 | 2,368 | 1,534 | 5,106 |
| Optimality gap (%) | 2.03 | 8.71 | 12.3 | 0.5 | 0.5 | 0.5 |

[1] Best solution found after 600 CPU sec.

[2] Using GAMS 21.3/CPLEX 9.0 on a Pentium M at 1.7 GHz; solved to 0.5% optimality gap.

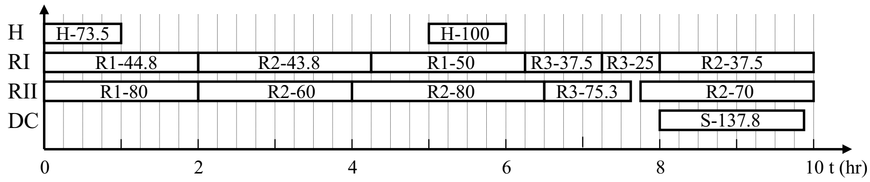


*Figure 2. Gantt chart of optimal solution of (M3) with Δt=0.25 hr.*

The scheduling of batch process remains a very hard problem and better methods are needed, especially for the solution of problems over long scheduling horizons. Nevertheless, model (M3) can be used for the solution of batch scheduling problems with intermediate release and due times, for which continuous-time models cannot be solved to optimality within reasonable time. Furthermore, it can be used for the solution of problems where holding and backlog costs are important. Finally, the proposed representation can be extended to account for continuous processes, with very good computational results. More details can be found in Maravelias (2005).

## 6. Conclusions

In this paper we formally show that discrete-time STN models are a special case of continuous-time models. A new mixed time representation (fixed grid, variable processing times) is proposed. Computational enhancements for the solution of the proposed representation are also presented.

### 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.

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.

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. A Mixed Time Representation for State Task Network Models. Submitted for Publication (**2005**).

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.

Sundaramoorthy, A.; Karimi, I.A. A Simpler Better Slot-based Continuous-time Formulation for Short-term Scheduling in Multipurpose Batch Plants. *Chem. Eng. Sci.*, In Press, **2005**.

### Acknowledgements

The author would like to thank Professor Ignacio Grossmann for stimulating discussions on the time representation of STN models.