Applications of the RTN Scheduling Model in the Chemical Industry

Hector D. Perez[a], Satyajith Amaran[b], Shachit S. Iyer[b], John M. Wassick[b], and Ignacio E. Grossmann[a]*

[a]*Carnegie Mellon University, Pittsburgh 15213, USA*
[b]*The Dow Chemical Company, Midland 48674, USA*
**grossmann@cmu.edu*

Abstract

The Resource-Task Network (RTN) model is a major contribution of Process Systems Engineering to the general area of scheduling optimization. The RTN representation models processes as bipartite graphs comprising two types of vertices: resources and tasks. A resource is general and includes all entities that are involved in the process steps (e.g., materials, processing and storage equipment, and utilities). A task is an abstract term for an operation that transforms a set of resources into another set. We provide a general review of the RTN model for both discrete and continuous-time representations. We then describe extensions to the standard RTN model that have been driven by the needs of the chemical industry. Successful industrial applications of these extensions in offline and online process scheduling and payload optimization, as well as a recent application to business processes, highlighting the impact that the RTN model continues to have in practice.

Keywords: Resource-Task Network, scheduling, spatial optimization, discrete event simulation, mathematical programming

1. Introduction

Over the past three decades, the Process Systems Engineering (PSE) community has pioneered the area of optimization-based process scheduling (Georgiadis et al., 2019; Maravelias, 2021). The essence of optimization-based scheduling is mathematical programming, and more specifically, mixed-integer programming (MIP), where a system is modelled by a set of continuous and discrete variables (e.g., batch sizes, process start times, equipment assignment) that are constrained by a series of algebraic inequalities and equalities (e.g., material balances, resource assignment constraints). An objective function is added to the resulting model, and the model is passed to a MIP solver that performs the optimization by seeking the point in the discrete-continuous feasible space with the optimum objective function value. Objective functions vary depending on what is being targeted with the model. Common examples of objective functions include, profit maximization, cost minimization, production makespan minimization, and tardiness minimization. Over the years, several models have been proposed to optimize the timing and sequencing of events at a chemical plant, as well as the allocation of limited resources, such as equipment, feedstock, utilities, and operators (Floudas and Lin, 2004; Méndez et al., 2006). At first, MIP models for scheduling were not readily accepted by the scientific community due to the claim that MIP models were too difficult to solve to render them of any practical use. However, improved model formulations (Shah et al., 1993), new algorithmic developments (Bixby and Rothberg, 2007), and the evolution of computing power (Chen, 2016) have facilitated the adoption of these models by industry, generating value in operations and manufacturing (Harjunkoski et al., 2014).

One of the more versatile, simple, and compact models proposed for process scheduling in PSE is the Resource-Task Network (RTN). The RTN model was developed in the 90's and has had a significant impact

in the chemical industry since its inception. In the last decade, several extensions have been developed and successfully implemented in industrial processes by researchers from The Dow Chemical Company and Carnegie Mellon University. These extensions have been prompted in response to limitations that have been observed in the standard RTN formulation when implementing it in practice (Wassick and Ferrio, 2011). The strengths of the RTN paradigm have also led to its application in other areas of PSE and other industries. These include payload optimization (Wassick and Ferrio, 2011), control systems (Nie et al., 2014), process network simulation models (Akiya et al., 2011), supply chain transactional processes (Perez et al., 2021a), and supply chain logistics (Chen, 2019).

This chapter is organized as follows. Section 2 presents a review of the classical RTN model formulations. Section 3 presents the extensions to the RTN model driven by the needs of the chemical industry. Section 4 highlights the impact that the model extensions have had in generating value in industry. Finally, Section 5 states concluding remarks.

2. Review of RTN Model

In 1993, the group at Imperial College led by Professor Roger Sargent published a novel modeling framework for process scheduling called the State-Task Network (STN) (Kondili et al., 1993; Shah et al., 1993). This modeling paradigm regards all processes as bipartite graphs with two types of vertices: *states* and *tasks*. A state represents a material involved in the process being scheduled (e.g., raw material, intermediate, or product). A task is an operation in which incoming states are transformed into outgoing states. The STN paradigm in its original form does not account for equipment and other resources explicitly in its network representation, but instead accounts for them mathematically with additional variables and constraints.

A year later, Costas Pantelides, a member of Roger Sargent's group, presented a conceptually more general and powerful modeling paradigm: the Resource-Task Network (Pantelides, 1994). This paradigm continues representing processes as bipartite graphs, but uses *resources* instead of *states*. The concept of a resource is general and includes all entities that are involved in the process steps, such as materials (e.g., raw materials, intermediates, and products), processing and storage equipment (e.g., tanks and reactors), and utilities (e.g., operators and steam). A sample diagram of an RTN is shown in **Figure 1**, which depicts the connections between resources and task nodes, and describes process typical specifications required for process scheduling. The resulting RTN model is represented mathematically using mixed-integer programming (MIP for both discrete and continuous time operations). Both commercial and open-source MIP solvers are available for optimizing the models. These solvers rely on techniques such as branch-and-bound (Dakin, 1965), cutting planes (Balas et al., 1993), and branch-and-cut (Johnson et al., 2000) when the models are linear. When nonlinearities are introduced in the models (e.g., when modeling inventory costs in continuous-time formulations), these may be solved using MINLP solvers that rely on branch-and-bound, Generalized Benders Decomposition (Geoffrion, 1972), Outer-Approximation methods (Duran and Grossmann, 1986; Su et al., 2018), and the Extended Cutting Plane method (Westerlund and Pettersson, 1995). Another option is to linearize the MINLP models via exact linearization (Glover, 1975) or piecewise linear approximations (Gupta and Grossmann, 2012; Sridhar et al., 2013).

The RTN paradigm has the advantage of producing more compact models, especially when systems have identical processing units (e.g., redundant or parallel equipment). When alternate processing units exist

for the same task, a copy of the task node must be made for each alternate resource type. In its simplest form, the RTN model only requires three sets of constraints (resource balances, resource limits, and operational constraints) and three variables types (resource inventories, task extents, and task triggers). Both discrete-time and continuous-time representations exist for the RTN model.

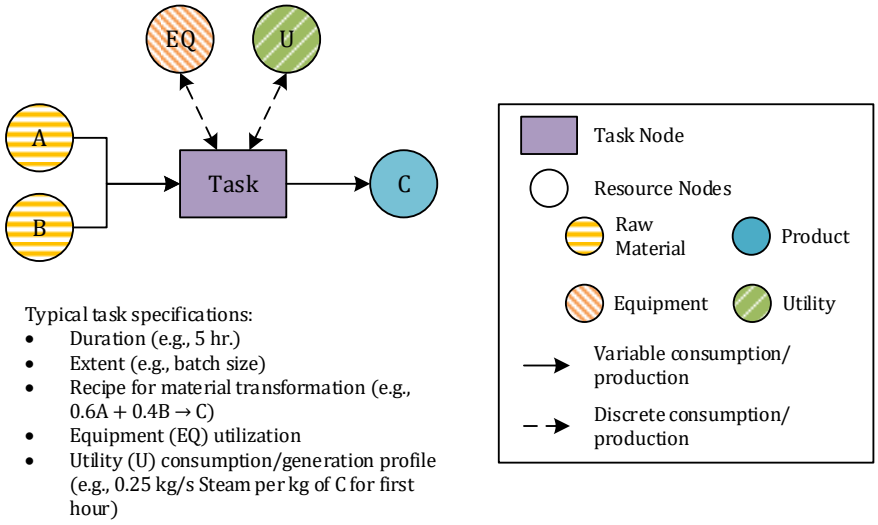

Figure 1. Sample network structure of an RTN

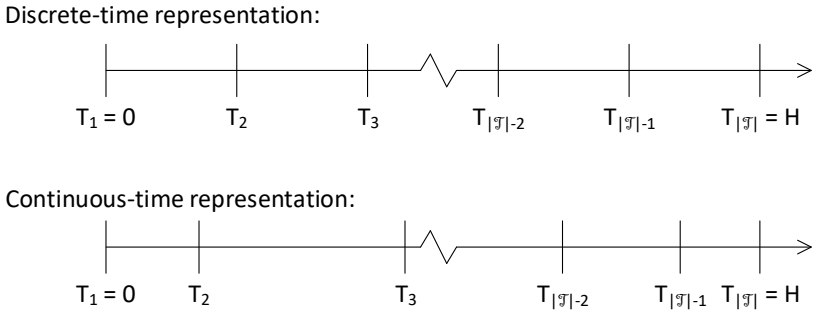

Figure 2. Discrete-time (top) and continuous-time (bottom) representations for RTN

2.1 Discrete-time Representation

The original formulation of the RTN model presented by Pantelides (1994) is for a discrete-time representation. The classical discrete-time formulation splits the temporal axis onto a uniform grid with time intervals of equal size as shown in **Figure 2**. The three sets of constraints in this formulation are described below.

2.1.1 Resource Balance

The resource level balance is given by (1), which indicates that the resource level for a resource r at time point t, $R_{r,t}$, is obtained by updating the resource level in the previous time point (or the initial resource value if t is the first time point) with the consumption or production of that resource and any entrance or exit of the resource into or from the system. The consumption and production of the resource is governed by the variables $N_{i,t}$ (integer) and $\xi_{i,t}$ (continuous), which indicate the number of occurrences of task i

and the extent of the task at time point t, respectively. The parameters $\mu_{i,r,t}$ and $\nu_{i,r,t}$ indicate the consumption/production ratios relative to the number of task occurrences and the task extent, respectively. The sign on the parameter indicates if it is a consumption (negative) or production (positive) term. The parameter $\mu_{i,r,t}$ is for resources that are consumed in discrete quantities (e.g., equipment and operators), and $\nu_{i,r,t}$ is for resources that are consumed in variable quantities (e.g., materials and utilities).

$$R_{r,t} = R_{r,t-1} + \sum_{i \in \mathcal{I}_r} \sum_{\theta=0}^{\tau_i} \left(\mu_{i,r,\theta} \cdot N_{i,t-\theta} + \nu_{i,r,\theta} \cdot \xi_{i,t-\theta} \right) + \Pi_{r,t} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \qquad (1)$$

The balance for resource r accounts for resource consumption and production in all tasks that involve the resource (set $\mathcal{I}_r$) and all time points in the range $[t - \tau_i, t]$, where τ_i is the duration of task i, since consumption or production can occur at any time point within the task duration. For continuous processes, τ_i is the minimum duration of the continuous task, such that a continuous task can be thought of as multiple small tasks in series of τ_i duration (typically $\tau_i = 1$). The entrance and exit of resource r to the system at time t is governed by parameter $\Pi_{r,t}$, which is positive when a resource enters (e.g., supply of raw materials), and negative when it leaves the system (e.g., delivery of a final product).

2.1.2 Resource Limits

All resource inventories have lower and upper bounds as expressed in (2).

$$R_r^{min} \leq R_{r,t} \leq R_r^{max} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \qquad (2)$$

2.1.3 Operational Constraints

The extent of a task (e.g., batch size) is forced to zero when the task is not executed (i.e., $N_{i,t} = 0$), and is bounded between the limits V_i^{min} and V_i^{max}, when the task is performed (i.e., $N_{i,t} \in \mathbb{Z}^+$).

$$V_i^{min} \cdot N_{i,t} \leq \xi_{i,t} \leq V_i^{max} \cdot N_{i,t} \qquad \forall i \in \mathcal{I}, t \in \mathcal{T} \qquad (3)$$

2.1.4 Variable Domains

Variable domains are $R_{r,t} \in \mathbb{R}^+ \; \forall r \in \mathcal{R}, t \in \mathcal{T}$ and $N_{i,t} \in \mathbb{Z}^+, \xi_{i,t} \in \mathbb{R}^+ \; \forall i \in \mathcal{I}, t \in \mathcal{T}$. It should be noted that resources that are consumed/produced in discrete quantities, such as processing equipment, may be declared as integer variables if desired. This may, in some cases, be computationally advantageous for the MIP solvers. MIP solvers may be able to further reduce model complexity during presolve by variable fixing and constraint propagation. Solvers may also more readily exploit integrality constraints when applying cutting plane methods (Marchand et al., 2002; Ostrowski et al., 2012).

2.1.5 Illustrative Example

Consider a system with one reactor (RX) that transforms raw materials A and B into intermediate C, which is immediately distilled in two parallel distillation columns (DC_1 and DC_2) to produce products D and E, as shown in **Figure 3**. The system begins with zero material inventories and has the following events: 1) a shipment of materials A (60 units) and B (40 units) enters at $t = 1$, 2) the reactor runs from $t = 2$ to $t = 4$ (100-unit batch with a 60A/40B feed ratio), 3) the two columns separate 50-unit batches simultaneously from $t = 4$ to $t = 5$, and 4) all products get shipped out at $t = 6$. The RTN representation for this system is shown in **Figure 4**. Resources are consumed when a task starts and produced a task completes. **Figure**

5 shows the resulting schedule and the resource inventory levels at each time point. Note that although intermediate C is produced at $t = 4$, it is immediately consumed, causing its inventory to always be zero.

2.2 Continuous-time Representation

In order to overcome the limitations of having to specify the length of the time intervals in the original discrete-time RTN formulation, which gave rise to large-scale Mixed-Integer Linear Programming (MILP) models, several alternative continuous-time formulations have been proposed by various researchers. The more modern and robust version of this continuous-time formulation was developed by researchers in Portugal at the INETI and the Instituto Superior Técnico during the early 2000s (Castro et al., 2001, 2004). This formulation considers time as continuous, where the system state is monitored at specified time points, the location of which is to be determined. This representation can be thought of as having a flexible non-uniform discrete time grid as shown in **Figure 2**, where the modeler sets the number of positions on the temporal axis to monitor, and the optimizer decides the best locations for these time points. The assumption here is that tasks can only be triggered at these time points, but can end at any time according to the duration of the task. A major challenge of this formulation is in determining the required number of time points. Since the optimal number of time points is not known a priori, the common practice is to successively increase the number of time points in the set $\mathcal{T}$ until no significant improvement is seen in the final objective function value. However, as pointed out by Castro et al. (2004), this can become intractable and may require fixing variables to reduce model complexity and avoid long computational times. See also Lee and Maravelias (2020) for an additional discussion on various strategies in this area.

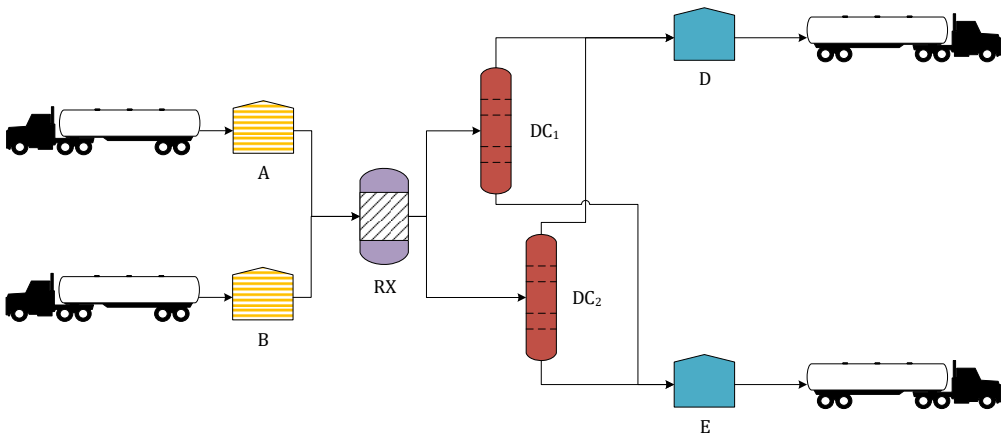

Figure 3. Flow diagram for illustrative example.

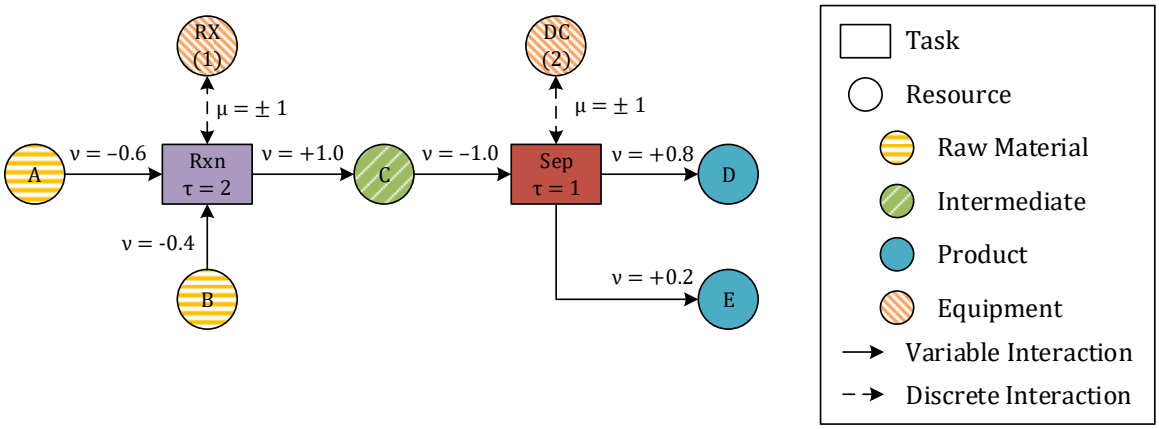

Figure 4. RTN representation for illustrative example with discrete (μ) and variable (ν) consumption and production ratios (indices are dropped for simplicity)

	Time Point						
Resource	**0**	**1**	**2**	**3**	**4**	**5**	**6**
A	0	60	0	0	0	0	0
B	0	40	0	0	0	0	0
C	0	0	0	0	0	0	0
D	0	0	0	0	0	80	0
E	0	0	0	0	0	20	0
RX	1	1	0	0	1	1	1
DC	2	2	2	2	0	2	2

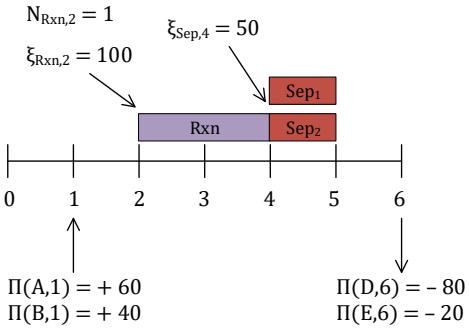

Figure 5. Schedule of events and variable values in illustrative example

2.2.1 Timing and Sequencing

The difference between any two time points, $t < t'$, on the flexible non-uniform time grid is determined by the durations of all tasks starting at time point t and ending in the interval $(t' - 1, t']$, as given by (4). The task durations can be modelled as having two parts: one composed of a fixed task duration (α_i), and the other of a variable task duration (β_i) that is proportional to the extent of the task. This constraint ensures that the time indices on the triggering and extent variables ($N_{i,t,t'}$ and $\xi_{i,t,t'}$, respectively) are mapped to the temporal axis accordingly. It also ensures that the difference between any two time points is greater than zero. Note that t and t' are not necessarily adjacent points on the time grid. The first and last time points are fixed to the beginning and end of the scheduling horizon: $T_0 = 0$ and $T_{|\mathcal{T}|} = H$. The feasible space defined by (4) can be further reduced by applying it to $t' \leq \Delta t + t$, where Δt is a parameter that represents the maximum number of time periods allowed for a task in the process.

$$T_{t'} - T_t \geq \sum_{i \in \mathcal{I}_r} \left(\alpha_i \cdot N_{i,t,t'} + \beta_i \cdot \xi_{i,t,t'} \right) \qquad \forall r \in \mathcal{R}^{EQ}, t \in \mathcal{T}, t' \in \mathcal{T}, t < t' \qquad (4)$$

2.2.2 Resource Balance

The resource inventory balance is analogous to that used in the discrete-time formulation. The production and consumption terms are split into the two summations in the second term on the right-hand side of (5) due to the double time point indexing on the variables. The first summation is for resource production and the second for resource consumption. An additional term is added at the end for storage tasks

associated with resource r ($\mathcal{I}_r^{ST}$) to account for the consumption and release of storage resources, which is particularly relevant in continuous processes. Additional constraints for these storage tasks are presented in Castro et al. (2004). There is also no parameter for external interactions in the base formulation since the exact position of the time points is not known a priori and would require additional logic-based constraints, which increases model complexity (Perez et al., 2021b).

$$R_{r,t} = R_{r,t-1} + \sum_{i\in\mathcal{I}_r}\left[\sum_{t'<t}\left(\mu_{i,r}^{P}\cdot N_{i,t',t} + \nu_{i,r}^{P}\cdot \xi_{i,t',t}\right) + \sum_{t<t'}\left(\mu_{i,r}^{C}\cdot N_{i,t,t'} + \nu_{i,r}^{C}\cdot \xi_{i,t,t'}\right)\right] + \sum_{i\in\mathcal{I}_r^{ST}}\left(\mu_{i,r}^{P}\cdot N_{i,t-1,t} + \mu_{i,r}^{C}\cdot N_{i,t,t+1}\right) \quad \forall r\in\mathcal{R}, t\in\mathcal{T} \qquad (5)$$

2.2.3 Variable Domains

The variable domains are the same as in the discrete-time formulation, except $N_{i,t,t'}$ is now binary.

2.3 Discrete-time vs. Continuous-time

2.3.1 Representation of Time

Continuous-time mathematical models are more accurate and flexible since time is a variable. On the other hand, discretization often introduces rounding error, especially for coarse time grids.

2.3.2 Model Size

As the time discretization becomes more refined, discrete-time models become significantly larger. The size of the time step needs to be specified by the user to minimize round-off errors that result in low-quality schedules. This trade-off between tractability and accuracy is relatively straightforward to determine by inspecting the system time scales. On the other hand, continuous-time models typically require fewer variables. The challenge here lies in determining the number of time points to use, which requires a successive manual adaptive refinement, as mentioned previously.

2.3.3 Linear Programming Relaxations

Discrete-time formulations have been found to present much tighter linear programming relaxations, making them usually faster to solve than continuous-time models, despite their larger model sizes (Harjunkoski et al., 2014). Discretization has also become less of an issue with the great advances by MILP solvers (e.g., CPLEX, Gurobi, Xpress).

2.3.4 Objective Functions

Continuous-time RTN is more amenable to makespan minimization than its discrete-time counterpart, which requires specifying the magnitude of the scheduling horizon a priori. On the other hand, discrete-time RTN handles cost minimization better than the continuous-time model, especially when inventory costs are considered, as these introduce nonlinearities in the latter because time is a variable (Méndez et al., 2006). In addition to preserving model linearity when accounting for inventory, discrete-time models monitor inventory and resources effectively because the monitoring points are fixed on the time grid.

2.3.5 Discrete-continuous-time Integration

More recently, efforts have been made to integrate the advantages of both discrete and continuous-time scheduling algorithms in work done at the University of Wisconsin – Madison led by Professor Christos T. Maravelias, who has since moved to Princeton University (Lee and Maravelias, 2020). Their work, which

is based on the STN model, proposes solving a coarse discrete-time MIP model to fix key assignment decisions and obtain an approximate schedule. A mapping algorithm then maps the approximate solution onto unit or material specific temporal grids. A continuous-time linear programming (LP) model then refines the initial discrete-time solution and eliminates approximation errors. A similar procedure could be applied in RTN to help overcome the challenges of the different model representations and exploit the best of both worlds.

3. Industry-led Developments

3.1 Extended RTN Model

3.1.1 Quality-Based Changeovers

During scheduling optimization, especially batch scheduling, equipment changeovers are considered when transitioning from one family of materials to another due to safety (e.g., chemical stability), throughput (e.g., fouling), operational constraints (e.g., time to reach steady state in semi-continuous processes), or quality reasons (e.g., product purity). Equipment cleaning tasks are included in the RTN framework via an additional changeover task. When the RTN model accounts for changeover tasks, duplicate equipment can no longer be aggregated in the same type of resource. Instead, each equipment involved in changeovers needs to be treated as its own resource to track which one is clean or needs cleaning (Méndez et al., 2006).

Another type of changeover that is used in industry when a changeover is required for quality reasons is that of quality-based changeovers. The idea with this type of changeover is to have the option of forgoing an equipment cleaning step when transitioning from one material family type to another by performing several sequential batches of the second family in tandem to dilute any residual contamination from the first family. This extension was first addressed explicitly by Brunaud et al. (2020). The diagram in **Figure 6** details the general RTN structure for a system with a single unit (e.g., reactor) that can perform two different reactions ($Rx1$ and $Rx2$). The first reaction converts $A \rightarrow E$ and the second reaction $B \rightarrow F$. When transitioning from $Rx1$ to $Rx2$, either a cleaning step or a quality-based changeover with $N+1$ sequential repetitions of $Rx2$ can be performed. When transitioning from $Rx2$ to $Rx1$, either a cleaning step or a quality-based changeover with $M+1$ sequential repetitions of $Rx1$ can be performed.

The RTN structure has four material resources, one for each material. Instead of one resource for the reactor, which would be the case for a system neglecting changeovers, there are $N+M+2$ resources for the single reactor, representing the single reactor in its clean state ($R1cl$ and $R2cl$) or dirty state ($R1dx$ and $R2dy$ where $x \in \{1, \dots, N\}$ and $y \in \{1, \dots, M\}$ indicate the number of repetitions of the second task). The initial values of these resource levels should be such that only one of the reactor states has an initial value of one and the rest are zero-valued. This, along with the proper consumption and production ratios, ensures that the unit can only be in one state at a time. With a quality-based changeover, instead of two reaction tasks, there are $N+M+2$ reaction tasks. As the changeover occurs, impurities in the product tank carried over from the previous reaction are gradually diluted. Under this structure, the network complexity increases linearly with the number of task repetitions for each quality-based changeover and with the number of task transition pairs.

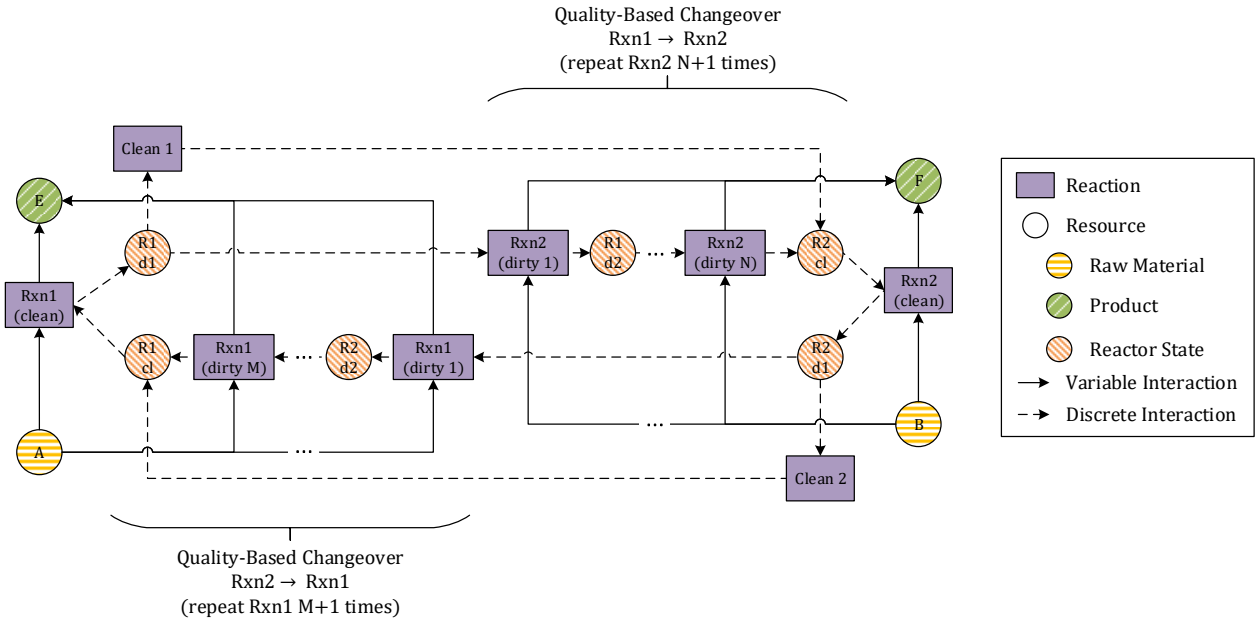

Figure 6. RTN representation illustrating both quality-chased changeovers and cleaning steps to transition between Reaction 1 and Reaction 2 on a single reactor

3.1.2 External Resource Transfers with Time Windows

Orders placed by customers or placed on suppliers can be mapped to the $\Pi_{r,t}$ term in the resource balance ((1)) via (6) and (7), respectively (Wassick and Ferrio, 2011). The $\Pi_{r,t}$ is separated into two terms to distinguish outgoing transfers ($\Pi_{r,t}^{out}$) from incoming transfers ($\Pi_{r,t}^{in}$). The set $\mathcal{R}_o$ contains the materials r associated with order o. The summation terms are for external material flows occurring between the early acceptance date (E_o) and final due date (D_o) of each order. It should be noted that the parameters E_o and D_o are converted from time values to time points using the discretization parameter (e.g., $D_o = 5$ for a 10 AM due date on the first day when the discretization time step is 2 hours and time point set $\mathcal{T}$ is zero-indexed). In a general sense, orders may also have a minimum and a maximum quantity required for each material ($Q_{o,r}^{min}$ and $Q_{o,r}^{max}$, respectively). Slack variables can also be introduced to ensure that the model is feasible even if $Q_{o,r}^{min}$ is not met. These slacks are penalized in the objective function.

$$Q_{o,r}^{min} - Q_{o,r}^{min,slack} \leq \sum_{t=E_o}^{D_o} -\Pi_{r,t}^{out} \leq Q_{o,r}^{max} \qquad o \in O, r \in \mathcal{R}_o \qquad (6)$$

$$Q_{o,r}^{min} - Q_{o,r}^{min,slack} \leq \sum_{t=E_o}^{D_o} \Pi_{r,t}^{in} \leq Q_{o,r}^{max} \qquad o \in O, r \in \mathcal{R}_o \qquad (7)$$

3.1.3 Point Orders

To ensure that external deliveries or shipments are made in a single material transfer, a common requirement in make-to-order systems, a point order task can be defined (Wassick and Ferrio, 2011). The point order task duration is typically much smaller than the delivery window for the order and is often zero (instantaneous order fulfillment). Point orders are included as an additional term in the resource balance that has the same form as the standard consumption/production term. The difference here is that the summation is applied only in the allowed delivery time point window for each point order i ($E_i \leq$

$t \leq D_i - \tau_i$) as shown in (8). The set $\mathcal{I}_r^{PO}$ is the subset of point order tasks that involve resource r. The bounds on the point order extents ($\xi_{i,t}$) are the order limits ($Q_o^{min} = \sum_{r \in \mathcal{R}_o} Q_{o,r}^{min}$ and $Q_o^{max} = \sum_{r \in \mathcal{R}_o} Q_{o,r}^{max}$) as shown in (9). The constraint in (10) ensures that each point order is triggered at most once in the allowed time window. The variable $N_{i,t}$ for the point order task i should be fixed to zero for all other time points. If multiple shipments are feasible, the right-hand side can be replaced with an integer value equal to the maximum number of shipments allowed.

$$R_{r,t} = R_{r,t-1} + \sum_{i \in \mathcal{I}_r} \sum_{\theta=0}^{\tau_i} \left(\mu_{i,r,\theta} \cdot N_{i,t-\theta} + \nu_{i,r,\theta} \cdot \xi_{i,t-\theta}\right) + \sum_{i \in \mathcal{I}_r^{PO}} \sum_{\theta=0}^{\tau_i} \left(\mu_{i,r,\theta} \cdot N_{i,t-\theta} + \nu_{i,r,\theta} \cdot \xi_{i,t-\theta}\right)|_{E_i \leq t \leq D_i - \tau_i} + \Pi_{r,t} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \qquad (8)$$

$$\left(\sum_{r \in \mathcal{R}_o} Q_{o,r}^{min}\right) \cdot N_{i,t} \leq \xi_{i,t} \leq \left(\sum_{r \in \mathcal{R}_o} Q_{o,r}^{max}\right) \cdot N_{i,t} \qquad \forall i \in \mathcal{I}^{PO}, t \in \mathcal{T} \qquad (9)$$

$$\sum_{t=E_i-\tau_i}^{D_i-\tau_i} N_{i,t} \leq 1 \qquad \forall i \in \mathcal{I}^{PO} \qquad (10)$$

3.1.4 Manipulation of Resource Limits

Material balances can be written for the resource level limits as well to facilitate operations such as material transfers without having to define auxiliary tasks, making the model more scalable (Wassick and Ferrio, 2011). This is particularly useful for situations in which total material transfers from one unit to another are required (e.g., when transferring the entire contents of a tank to a truck for delivery). In the standard RTN model, this requires defining auxiliary storage tasks and storage resources to guarantee that the entire contents of a tank are transferred to the truck in a single operation. The downside of doing this is that additional integer variables for the auxiliary tasks are introduced. Instead, continuous variables can be introduced by parametrizing the resource limits (R_r^{min} and R_r^{max}) with respect to time as shown in (11). These new variables ($R_{r,t}^{min}, R_{r,t}^{max} \in \mathbb{R}^+$) are bounded by the static resource limit parameters as indicated in (12). Additional constraints, given in (13) and (14), are then defined for these variables, analogous to the resource inventory balance in (1). Discrete and variable resource limit production/consumption rates are introduced for the respective resource inventory limits (i.e., $\mu_{i,r,t}^{max}$ and $\nu_{i,r,t}^{max}$ govern the discrete and continuous consumption of the limit $R_{r,t}^{max}$, respectively).

$$R_{r,t}^{min} \leq R_{r,t} \leq R_{r,t}^{max} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \qquad (11)$$

$$R_r^{min} \leq R_{r,t}^{min}, R_{r,t}^{max} \leq R_r^{max} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \qquad (12)$$

$$R_{r,t}^{max} = R_{r,t-1}^{max} + \sum_{i \in \mathcal{I}_r} \sum_{\theta=0}^{\tau_i} \left(\mu_{i,r,\theta}^{max} \cdot N_{i,t-\theta} + \nu_{i,r,\theta}^{max} \cdot \xi_{i,t-\theta}\right) \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \qquad (13)$$

$$R_{r,t}^{min} = R_{r,t-1}^{min} + \sum_{i \in \mathcal{I}_r} \sum_{\theta=0}^{\tau_i} \left(\mu_{i,r,\theta}^{min} \cdot N_{i,t-\theta} + \nu_{i,r,\theta}^{min} \cdot \xi_{i,t-\theta}\right) \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \qquad (14)$$

The new RTN structure for this extension is illustrated in **Figure 7**, where material in tank A is transferred to the delivery truck B. When the transfer task i is triggered, the upper limit of resource A is consumed by 100 units (e.g., $\mu_{i,A,t-\tau_i}^{max} = -100$), reducing it to zero, and thus forcing the entire resource to be transferred to resource B. Once the transfer task completes, the upper limit of resource A is restored (e.g., $\mu_{i,A,t}^{max} = +100$).

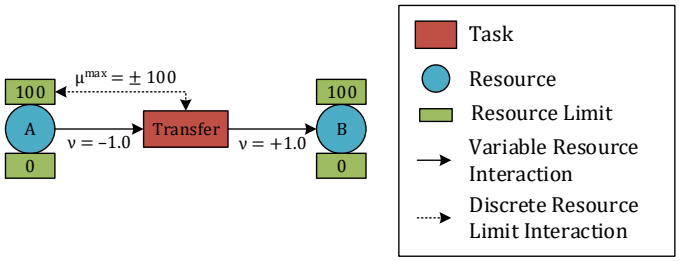

Figure 7. Example of extended RTN representation with resource limit interactions

3.1.5 Resource Slacks

Slack variables can be used to introduce soft bounds on resource inventories to produce schedules that would otherwise be infeasible. These include not having enough material to produce a full batch at the end of the scheduling horizon, or high demand scenarios where it may be allowed model to violate material inventory safety stocks (Wassick and Ferrio, 2011). The slacks are penalized in the objective.

$$R_{r,t}^{min} - R_{r,t}^{min,slack} \leq R_{r,t} \leq R_{r,t}^{max} - R_{r,t}^{max,slack} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \quad (15)$$

3.1.6 Multiple Task Extents

Multiple task extents can be used when multiple operations are performed under a single governing task (Wassick and Ferrio, 2011). Doing so avoids defining binary variables for the triggering of each operation involved. Instead, each operation is triggered by the governing task, $N_{i,t}$, which reduces the model complexity. To implement multiple task extents, the resource inventory balance in (1) is modified so that the task extent and variable interaction parameters are specific to the extent number, m (i.e., $\xi_{i,m,t}$ and $\nu_{i,m,r,t}$, respectively), as shown in (16). The index m is also added to the task extent bounds in (3).

$$R_{r,t} = R_{r,t-1} + \sum_{i \in \mathcal{I}_r} \sum_{\theta=0}^{\tau_i} \left(\mu_{i,r,\theta} \cdot N_{i,t-\theta} + \sum_{m \in \mathcal{M}_i} \nu_{i,m,r,\theta} \cdot \xi_{i,m,t-\theta} \right) + \Pi_{r,t} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \quad (16)$$

Multiple extents can be used to represent different material sizes processed in an operation, such as when a material is being split or materials are being combined in ways that cannot be described by single extents and consumption/production ratios. These multiple extents are further constrained with inequalities of the form shown in (17), where the set $\mathcal{K}_i$ is the set of constraints assigned to the multiple extents in task i. These constraints can be used to enforce material balances among the multiple procedures involved in the governing task (see the bottle filling task example given in Wassick and Ferrio (2011)).

$$\gamma_{i,k,t}^{min} \leq \sum_{m \in \mathcal{M}_i} \delta_{i,k,m,t} \cdot \xi_{i,m,t} \leq \gamma_{i,k,t}^{max} \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i, t \in \mathcal{T} \quad (17)$$

3.1.7 Industrial Example: Multiple Extents in a Continuous Processing Plant

Wassick and Ferrio (2011) show an industrial application at Dow of the extended RTN model for a continuous processing chemical plant. In the example, a one-year production schedule for the plant is performed using the concept of multiple extents to produce three different isomers. Isomers 1 and 2 can be produced in two different grades, and Isomer 3 is produced in a single grade as shown in **Figure 8**. Grade 1 isomers are produced in a single campaign that must last at least 14 days. Grade 2 isomers are also produced in a single campaign that lasts no less than 8 days. Campaign transitions take 3 days. Other parameters for the chemical plant are given in **Table 1**.

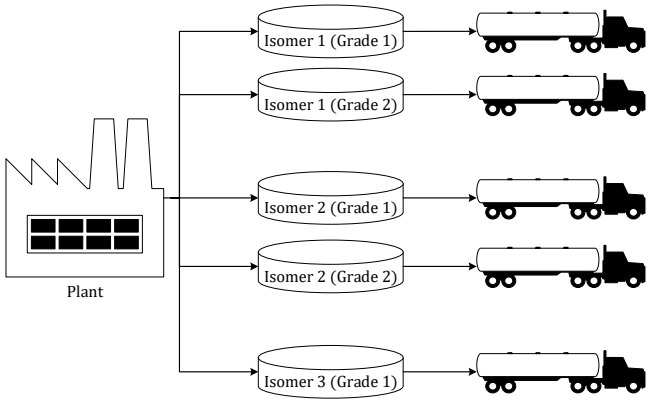

Figure 8. Continuous processing plant with three products of different grades[1]

The network diagram for the extended RTN model for the problem is illustrated in **Figure 9**. The diagram shown is for Grade 1 isomers. The other half of the network is the same as the one shown, where Grade 2 is used in place of Grade 1 and only two isomers are produced. A unique feature of this example is that time is modeled as a consumable resource. As a result, a non-uniform time grid is used, such that the number of time points must be specified a priori. The duration of each task is a single time slot (i.e., $\tau_i = 1$), and the duration of each time slot is dictated by the amount of time resource consumed in each task. The time resource is initialized at 365 days and is fixed to 0 at the last time point. Determining the minimum number of time points in this discrete-time model is done off-line. A few additional time points are then used with dummy tasks that consume no time to ensure that the number of time slots is sufficient and ensure that any unnecessary slots are consumed. There is a dummy task for each grade. The dummy task for Grade 1 is not shown in **Figure 9**, but is a task that consumes the *Production Plant* resource (PP node) and the *Plant ready for Grade 1* resource ($PRG1$ node) at the beginning of the time slot and releases them at the end of the slot. Note that this task has zero temporal duration because the time resource is not involved.

Table 1. Continuous processing plant parameters[2]

Parameter[a]	**Grade 1**			**Grade 2**	
	Isomer 1	*Isomer 2*	*Isomer 3*	*Isomer 1*	*Isomer 2*
Safety Stock	800	500	100	500	200
Inventory Capacity	5,000	6,049	1,000	5,000	2,000

[1] Adapted with permission Wassick, J.M., Ferrio, J., 2011. Extending the resource task network for industrial applications. Computers and Chemical Engineering 35, 2124–2140. Copyright (2011) Elsevier, Ltd.
[2] Ibid.

Daily Min. Production	75	50	6	140	20
Daily Max. Production	126	101	16	149	36
Daily Demand	95.89	76.71	12.05	21.41	4.33
Product Margin	$269.3	$361.2	$615.5	$674.0	$958.4
Inventory Cost	$27.0	$36.0	$62.0	$67.0	$96.0

[a]All parameters are on a Mlbs basis

The *Produce Grade 1* task has three types of extents: 1) campaign duration, 2) isomer production, and 3) isomer shipment. In the first extent type, the purple time resource is consumed (at the end of the time slot). The second extent type has three unique extents (one for each Grade 1 isomer), in which the green resources for stored isomers are filled with material produced (at the end of the time slot). The last extent type has five unique extents (one for each product), in which the green resources for stored isomers are drained (consumed) and the blue shipped isomer resources are filled (produced). Extents of the last type are indexed at the end of the time slot, allowing the model to monitor stored inventory and shipped products at the end of the campaign. Production extents are constrained to be within the production limits in **Table 1** multiplied by the time duration extent (i.e., $\dot{Q}^{min}_{G1,m} \cdot \xi_{G1,Time,t} \leq \xi_{G1,m,t} \leq \dot{Q}^{max}_{G1,m} \cdot \xi_{G1,Time,t} \;\; \forall m \in \mathcal{M}_{G1,2}, t \in \mathcal{T}$ where $\mathcal{M}_{G1,2}$ are the isomer production extents [type 2] in Grade 1).

Five resources are associated with the chemical plant (orange nodes in **Figure 9**), representing the plant itself (PP resource) and the four states it can be in: initial state before startup (PS resource), plant ready for production of a certain grade ($PRG1$ and $PRG2$ resources), and plant ready for a grade transition (PT resource). Each plant state can be either zero or one and the plant resource (PP) is forced to zero to ensure that the plant is always in use after startup. Isomer storage resources (green nodes in **Figure 9**) are bounded by the safety stocks and the inventory capacities given in **Table 1**. Isomer shipment resources (blue nodes in **Figure 9**) are bounded between zero and the annual demand for that isomer from **Table 1**.

In order to avoid unrealistic end-of-horizon effects on the produced schedule, the RTN model is constrained so that the ending inventory of each isomer matches its initial inventory: $R_{r,h} - R^{slack}_r \leq R_{r,0} \leq R_{r,h} + R^{slack}_r \;\; \forall r \in \mathcal{R}^P$ where $\mathcal{R}^P$ is the set of green nodes in **Figure 9** (stored product resources), and R^{slack}_r is a slack variable that is penalized in the objective. The objective function that is maximized is shown in (18), which includes the annual revenue for shipped resources minus the inventory holding costs and penalties for any non-zero slacks. Since time is a variable (resource), nonlinearities in the objective are avoided by approximating the inventory cost using the average inventory for each product over the horizon: $I^{ave}_r = \sum_{t\in\mathcal{T}} R_{r,t}/H \;\; \forall r \in \mathcal{R}^P$. The network model consists of 7,567 equations, 2,955 continuous variables, and 180 binary variables, and was solved using CPLEX 10.1.0 on a 1.83 GHz CPU. The system was optimized in 51 seconds with an optimality gap of 0.06%. The optimized production schedule is shown in **Figure 10**, which shows a larger production scheduled in Grade 1 campaigns. Inventory profiles for each product are obtained as shown in **Figure 11** for Grade 1 Isomer 1. The average daily product shipping rates obtained are quite stable, exhibiting minor variations throughout the year (relative standard deviations ~0.1%).

$$\max \phi = \sum_{t\in\mathcal{T}} \sum_{r\in\mathcal{R}^S} p_r \cdot R_{r,t} - \sum_{r\in\mathcal{R}^P} c^{inv}_r \cdot I^{ave}_r - \sum_{r\in\mathcal{R}^P} \sum_{t\in\mathcal{T}} c^{max}_r \cdot R^{max,slack}_{r,t} - \sum_{r\in\mathcal{R}^P} \sum_{t\in\mathcal{T}} c^{min}_r \cdot R^{min,slack}_{r,t} - \sum_{r\in\mathcal{R}^P} c^{slack}_r \cdot R^{slack}_r \tag{18}$$

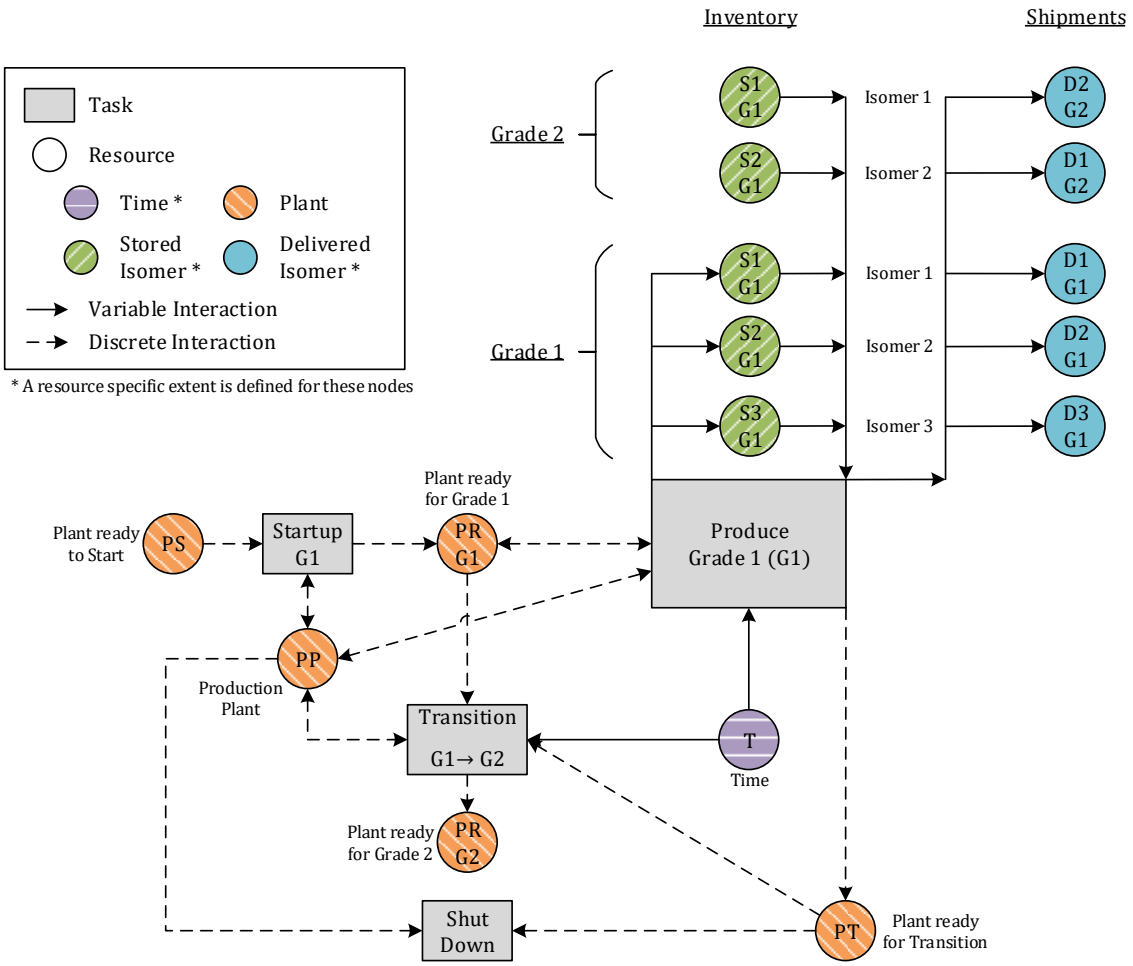

Figure 9. RTN diagram for Grade 1 production and product shipment in the continuous processing plant

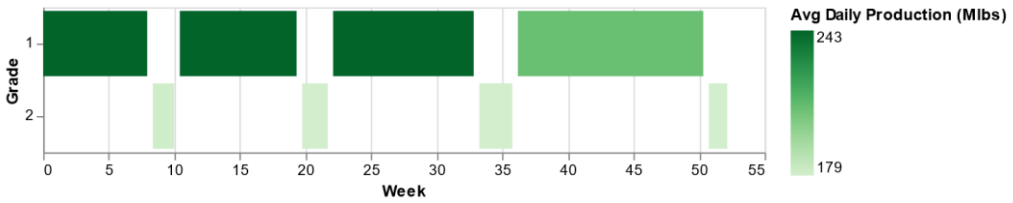

Figure 10. Production schedule indicating the average daily production for each campaign

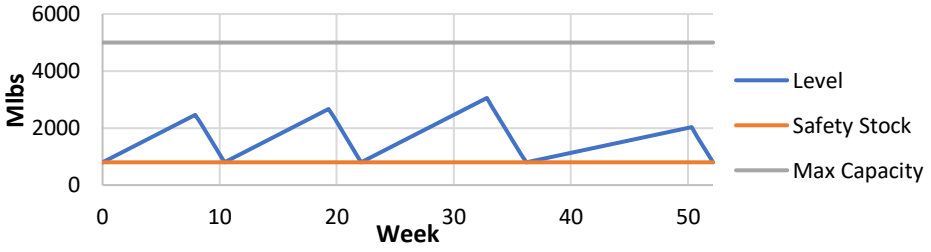

Figure 11. Optimized inventory profile for Isomer 1 (Grade 1)[3]

3.2 State Space Reformulation of Extended RTN Model

Nie et al. (2014) propose a reformulation of the extended discrete-time RTN model for reactive scheduling applications with a moving horizon. The extended RTN model is reformulated to the state space paradigm commonly used in process control. The reformulation requires lifting variables to introduce new state variables so that the system state can be fully specified at any point in time using the current state variables. The input variables in the model are the task assignments ($N_{i,t}$) and task extents, $\xi_{i,m,t}$. The state variables are the resource levels ($R_{r,t}$), resource limits ($R_{r,t}^{max}$ and $R_{r,t}^{min}$), and task history (lifted variables, $N_{i,t,\theta}$ and $\xi_{i,m,t,\theta}$). The lifted state variables ($N_{i,t,\theta}$ and $\xi_{i,m,t,\theta}$) record the past decisions in the system (i.e., the task assignment or extent for task i at time t that has been running actively for θ periods). Note that the input variables ($N_{i,t}$ and $\xi_{i,m,t}$) can be considered lifted variables for which $\theta = 0$. From a process control perspective, the external resource transfer term ($\Pi_{r,t}$) in the resource inventory balance is viewed as a system disturbance. In addition, the following dynamic state evolution equations are introduced,

$$N_{i,t,\theta} = N_{i,t-1,\theta-1} + \left(N_{i,t-1,\theta} - N_{i,t-1,\theta-1}\right) \cdot \sum_{\theta'=0}^{\theta} d_{i,t-1,\theta'} \qquad \forall i \in \mathcal{I}, t \in \mathcal{T}, 1 \le \theta \le \tau_i \quad (19)$$

$$\xi_{i,m,t,\theta} = \xi_{i,m,t-1,\theta-1} + \left(\xi_{i,m,t-1,\theta} - \xi_{i,m,t-1,\theta-1}\right) \cdot \sum_{\theta'=0}^{\theta} d_{i,t-1,\theta'} \qquad \forall i \in \mathcal{I}, m \in \mathcal{M}_i, t \in \mathcal{T}, 1 \le \theta \le \tau_i \quad (20)$$

Fixed binary disturbance parameters, $d_{i,t,\theta}$, are used to model unexpected delays in task durations, such that $d_{i,t,\theta} = 1$ indicates a one period delay at time point t on task i, which has been actively running for θ periods. Consecutive disturbance parameters can be used ($d_{i,t,\theta}, \dots, d_{i,t+\tau_d-1,\theta}$) to model longer task delays of duration τ_d. **Figure 12** illustrates an example of the dynamic state evolution of a task with a 2-period duration that is triggered at $t = 0$ (green state node). Without disruptions, the state evolution is $N_{i,0,0} \rightarrow N_{i,1,1} \rightarrow N_{i,2,2}$. If a 1-period delay occurs at $t = 1$ ($d_{i,1,1} = 1$), the evolution is $N_{i,0,0} \rightarrow N_{i,1,1} \rightarrow N_{i,2,1} \rightarrow N_{i,3,2}$. If a 2-period delay occurs at $t = 1$, which is equivalent to an additional 1-period delay at $t = 2$ ($d_{i,2,1} = 1$), the evolution is $N_{i,0,0} \rightarrow N_{i,1,1} \rightarrow N_{i,2,1} \rightarrow N_{i,3,1} \rightarrow N_{i,4,2}$.

Using a state space formulation has important advantages. The dynamic state of the system can be fully described with the state variables at any given time. Furthermore, the future state of the system can be forecast using the input and state variables. This makes this reformulation amenable to advanced process control applications, such as model predictive control.

[3] Ibid.

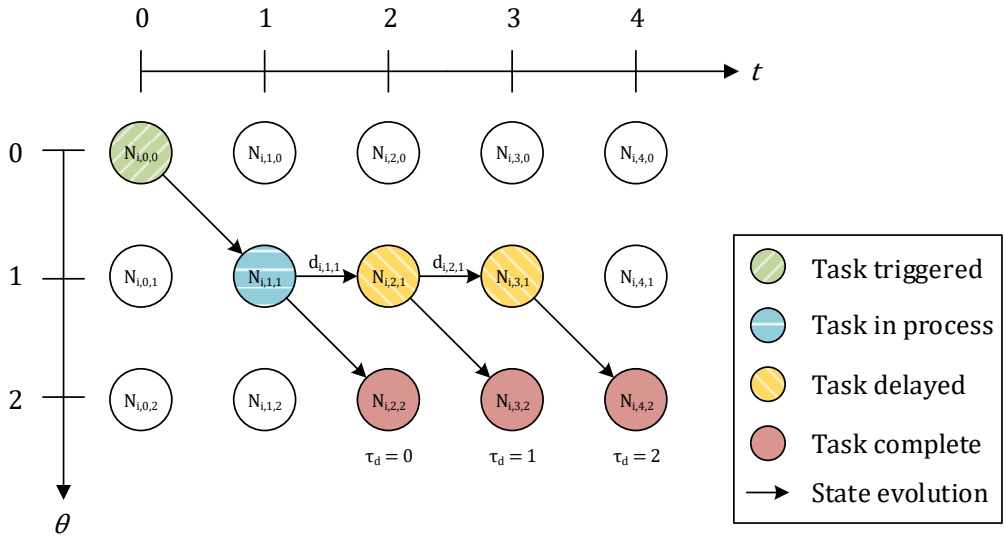

Figure 12. Sample dynamic state evolution of a 2-period task with different delays[4]

3.2.1 Industrial Example: Online Scheduling of a Mixed Batch/Continuous Processing Plant

Nie et al. (2014) present an industrial application of online RTN scheduling for the processing plant with both batch and continuous unit operations depicted in **Figure 13**. The system consists of two parallel batch units that are followed by a serial train with two buffer tanks, a continuous processing unit, another buffer tank, and a final continuous unit, in that order. The plant produces five types of products (A – E). Products A, C, and D belong to the same product family, and B and E are their own families. Raw materials are assumed to be unlimited, except for an intermediate F that can be produced in the batch units and is consumed to produce A, B, and E. The system operation is governed by the following characteristics,

- Batch durations and sizes are fixed,
- Batch units have a zero-wait transfer policy,
- Buffer tanks have variable inlet/outlet flow rates,
- Continuous units have variable processing rates and zero residence times,
- Continuous units store no inventory,
- Product mixing is not allowed,
- Changeover tasks must be considered for different product families in the continuous units,
- The second continuous unit has a consumable resource (e.g., filter) that requires shutting down for its regeneration/replacement, and
- Materials A – F have unlimited storage capacity.

The features described above, along with the plant structure and characteristics are modelled using the extended state space RTN model. The objective function maximizes is given in (21), which includes the revenue for order fulfillment within the specified time windows and penalties for product transition tasks, buffer tank inventories, and violating the safety stock levels at the end of the scheduling horizon. Sales prices for materials are indexed by t ($p_{o,r,t}$), such that higher revenue is received for early fulfillment. An

[4] Adapted with permission from Nie, Y., Biegler, L.T., Wassick, J.M., Villa, C.M., 2014. Extended discrete-time resource task network formulation for the reactive scheduling of a mixed batch/continuous process. Industrial and Engineering Chemistry Research 53, 17112–17123. Copyright (2014) American Chemical Society.

important feature in the model is the small penalty on the buffer tank inventory levels. This is done to break the symmetry in the flow transfer profiles when the continuous processing units are operated under full capacity. In such instances, the model forces flow from the buffer tanks to occur as fast as possible. This avoids oscillatory behavior in the flow profiles, which is operationally favorable. See Nie et al. (2014) for model parameter values.

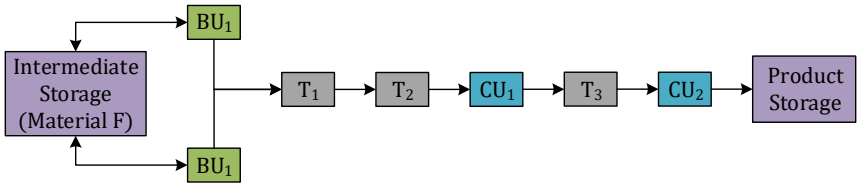

Figure 13. Block flow diagram for plant with batch units (BU), buffer tanks (T), continuous units (CU), and storage tanks[5]

$$\max \phi = \sum_{o\in O}\sum_{r\in \mathcal{R}_o}\sum_{t=E_o}^{D_o} p_{o,r,t}\cdot(-\Pi_{r,t}) - \sum_{i\in \mathcal{I}^{TR}}\sum_{t\in\mathcal{T}} c_i^{TR}\cdot N_{i,t} - \sum_{r\in\mathcal{R}^{ST}}\sum_{t\in\mathcal{T}} c_r^{inv}\cdot R_{r,t} - \sum_{r\in\mathcal{R}^{P}}\sum_{t\in\mathcal{T}} c_r^{min}\cdot R_{r,h}^{min,slack} \quad (21)$$

The model is used for the deterministic scheduling of the plant with a look-a-head horizon of 72 hours (3 days). The product orders for each day are indicated in **Table 2**, where the quantity of each material ordered on a given day is 70 units. After the first 24 hours, a rescheduling is performed looking ahead for days 2 – 4. Three scenarios are considered: 1) no disruptions, 2) a scheduled 11-hour delay on the first batch unit (BU_1) starting on day 2, and 3) a scheduled 13-hour maintenance on the last buffer tank (T_3) at hour 60. Delays are readily introduced in the second scenario using the disturbance parameters in the dynamic state evolution equations. The maintenance event is produced by setting the upper resource limit of the buffer tank to zero for the 13 hours in the maintenance event. The model in each of the cases consists of 140,382 constraints and 143,148 variables (4,380 discrete) and was solved within a 10-minute limit using Gurobi 5.5.0. After 10 minutes, the optimality gap was less than 5% in each run. The resulting schedule for the first 72 hours is shown in **Figure 14** with inventory profiles shown in **Figure 15**. Similar schedules for the next three scenarios are given in Nie et al. (2014). The advantage of the state space RTN model is in the implementation of the three scenarios for days 2-4, which are readily initialized with the relevant system history from the previous 24 hours of operation stored in the state variables. The rescheduling at $t = 24$ is smooth and does not introduce drastic operational changes. The state space model is also amenable to responding to system disturbances.

Table 2. Product orders matrix for materials A – E (all orders are for 70 units)[6]

Day	A	B	C	D	E
1	X				
2	X		X		X
3	X	X	X	X	
4	X			X	

[5] Ibid.
[6] Ibid.

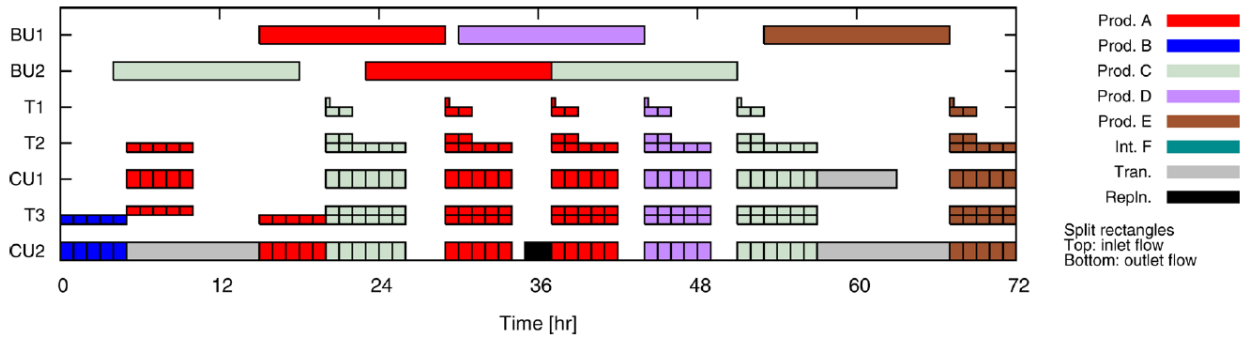

Figure 14. Optimal schedule for the first 72 hours in the mixed batch/continuous plant[7]

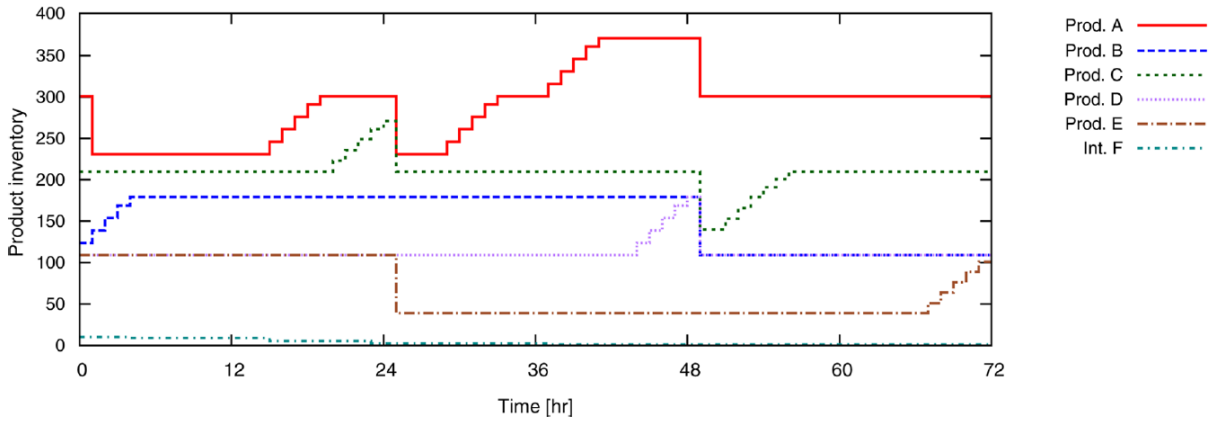

Figure 15. Inventory profiles for the first 72 hours in the mixed batch/continuous plant[8]

3.3 RTN for Spatial Packing Problems

Wassick and Ferrio (2011) also present a new application for RTN outside of process scheduling: 1D and 2D packing problems. The RTN model for 2D strip packing optimization is time agnostic and uses a single resource: horizontal (x) floor space at location y ($R_{x,y}$). Thus, the number of resources is equal to the number of discrete points on the x-axis. Instead of tracking each resource over time, the resources are tracked along the vertical (y) axis. Each horizontal resource has an initial condition of unity and an upper bound of unity:

$$R_{x,-1} = 1 \qquad \forall x \in \mathcal{X} \quad (22)$$

$$0 \leq R_{x,y} \leq 1 \qquad \forall x \in \mathcal{X}, y \in \mathcal{Y} \quad (23)$$

A placement task i is defined as placing a rectangular package of type i with width W_i and length L_i on the floor plan. When placing a package, the (x, y)-coordinates of the bottom left corner of the rectangular box must be specified (see red X's in **Figure 17**). Thus, instead of indexing the task triggering variable with the task i and "time" y as in the standard RTN model, the triggering binary variable is also indexed by the resource x ($N_{i,x,y}$). When a rectangular box is placed at a position (x, y), it restricts the possibility of

[7] Reprinted with permission from Nie, Y., Biegler, L.T., Wassick, J.M., Villa, C.M., 2014. Extended discrete-time resource task network formulation for the reactive scheduling of a mixed batch/continuous process. Industrial and Engineering Chemistry Research 53, 17112–17123. Copyright (2014) American Chemical Society.
[8] Ibid.

placing another box in the positions between x and $x + L_i - 1$ for that same y-coordinate. Thus, triggering the task at position y, consumes the horizontal resources $R_{x',y}$ for $x \leq x' \leq x + L_i - 1$ as indicated in **Figure 16**. This in turn implies that a resource x is consumed for any boxes of type i that are placed in a position $x' \in \mathcal{X}_i$ such that $x - L_i + 1 \leq x' \leq x$. The floor space inventory balance becomes:

$$R_{x,y} = R_{x,y-1} + \sum_{i \in \mathcal{I}} \sum_{x' \in \mathcal{X}_i} \sum_{\theta=0}^{W_i} \mu_{i,x,\theta} \cdot N_{i,x',y-\theta} \qquad \forall x \in \mathcal{X}, y \in \mathcal{Y} \quad (24)$$

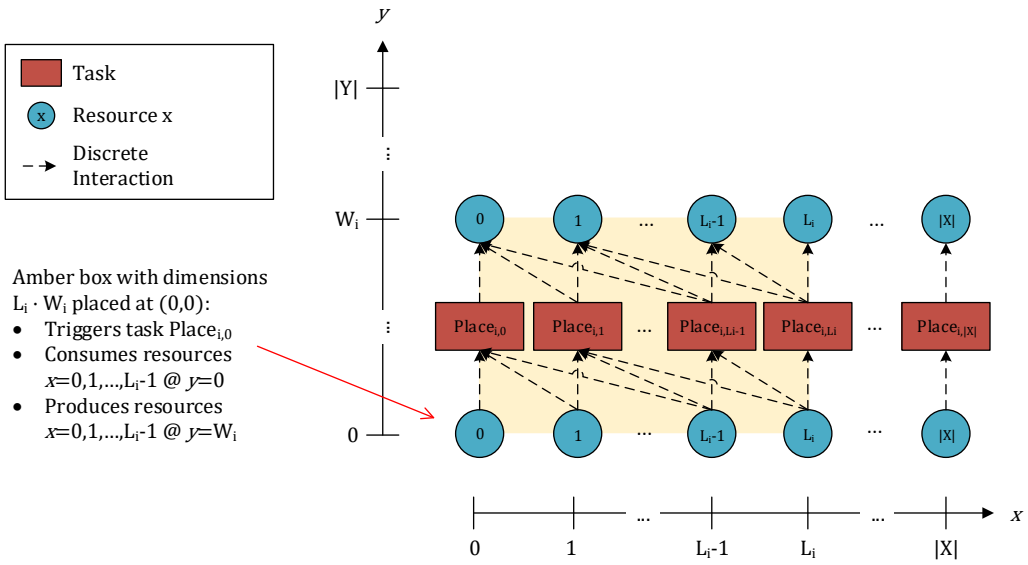

Figure 16. RTN representation for a 2D strip packing problem

Note that a resource x is consumed when the bottom left corner of a box is placed at position (x, y), and is then restored at the top left corner of the box (vertical position $y + W_i$), meaning a new box can be placed on position $(x, y + W_i)$. The width of the box can be viewed as the task "duration". The discrete consumption/production ratio μ takes the following values: -1 when $\theta = 0$, $+1$ when $\theta = W_i$, and 0 otherwise. As a result, (24) can be expressed as:

$$R_{x,y} = R_{x,y-1} + \sum_{i \in \mathcal{I}} \sum_{x'=x-L_i+1}^{x} \left(N_{i,x',y-W_i} - N_{i,x',y} \right) \qquad \forall x \in \mathcal{X}, y \in \mathcal{Y} \quad (25)$$

For the simpler 1D strip packing problem, the x dimension used in the 2D case can be dropped and the system is modeled as having a single resource along the vertical length (y) of the strip.

3.3.1 Industrial Example: Payload Loading Optimization

Wassick and Ferrio (2011) present an application of the spatial RTN model to load a semi-trailer for payload transport. The objective is to maximize the number of packages that are loaded onto the floor space of the semi-trailer. Package stacking is not allowed, and highway weight restrictions enforce a maximum total payload weight and a maximum load on the rear (tandem) axle. The weight limitations are introduced using the force balance illustrated in **Figure 17**. The force balance for the packages (m_i), the king pin (F_K), and the tandem (rear) axle (F_T) is represented in (26). The total weight constraint is given

in (27), the moment balance in (28), and the rear axle weight limit in (29). Four different package types allowed on the 10-unit long semi-trailer. See Wassick and Ferrio (2011) for model parameter values.

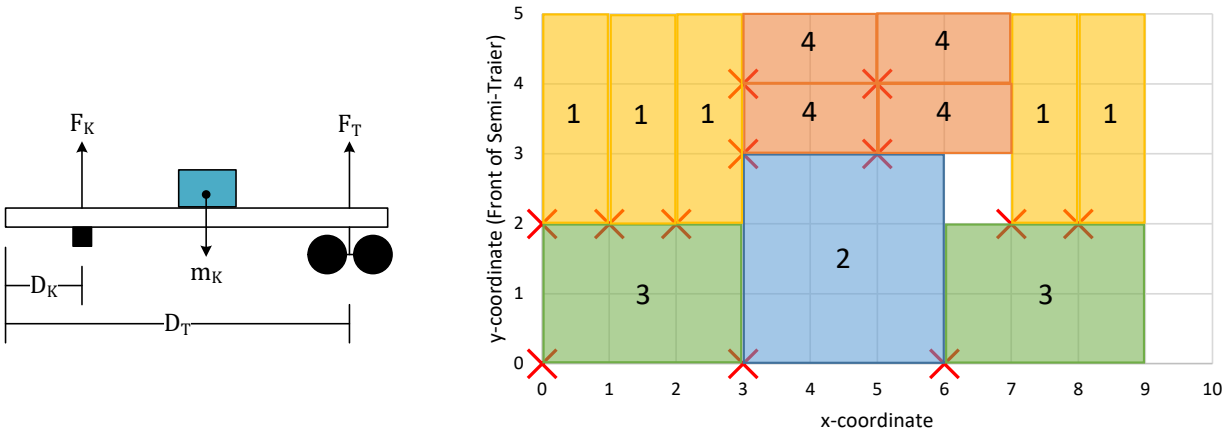

Figure 17. Semi-trailer force diagram (left) and optimized layout (right) for payload packing problem[9]

$$F_T + F_K = \sum_{i\in\mathcal{I}} \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} N_{i,x,y} \cdot m_i \tag{26}$$

$$F_T + F_K \leq W^{max} \tag{27}$$

$$\sum_{i\in\mathcal{I}} \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} N_{i,x,y} \cdot m_i \cdot \left(x + \frac{L_i}{2}\right) = D_K \cdot F_K + D_T \cdot F_T \tag{28}$$

$$F_T \leq F_T^{max} \tag{29}$$

The model objective is to maximize the payload weight ($F_T + F_K$). The resulting model (131 binary variables, 211 continuous variables, and 77 equations) was solved using CPLEX 10.1.1 at the parent node (110 iterations) without branching. The optimized loading layout is shown in **Figure 17**, where 12 boxes that weigh a total of 44 units are loaded, placing the maximum allowable load on the rear axle (18 units). The spatial RTN model has been successfully applied at Dow to maximize payload and reduce freight costs. Additional constraints have also been included to stay within axle load limits, account for rotating packages, and other special packaging arrangements.

3.4 RTN for Transactional Business Process Optimization

Another interesting application of the RTN model currently under development is for the online optimization of business transactional processes in a supply chain (Perez et al., 2021a). Some common business processes found in supply chains are the order-to-cash (OTC) and procure-to-pay (PTP) processes. These processes describe the end-to-end transactions that must occur on a request (e.g., actions that must be performed between when a customer places an order and payment is received for the fulfilled order). A digital twin framework has been developed that models supply chain business processes as transactional queueing networks. This concept is illustrated in **Figure 18**. As requests (e.g., customer orders) arrive and move downstream, they go through a series of steps that are performed on

[9] Adapted with permission Wassick, J.M., Ferrio, J., 2011. Extending the resource task network for industrial applications. Computers and Chemical Engineering 35, 2124–2140. Copyright (2011) Elsevier, Ltd.

the request (e.g., credit check, import/export customs forms, inventory confirmation, shipment loading, and invoice generation). The steps need not be serial, as there may be parallel transactions, as well as deviations from the "normal" flow of a request through the system (e.g., to correct errors in the order or reschedule a shipment). At each transaction, there are a limited number of resources assigned to that transaction (e.g., human or automated agents). Each transaction has a duration that is uncertain and is modelled as a continuous probability distribution, which can be specific to the request type, requestor, or agent.

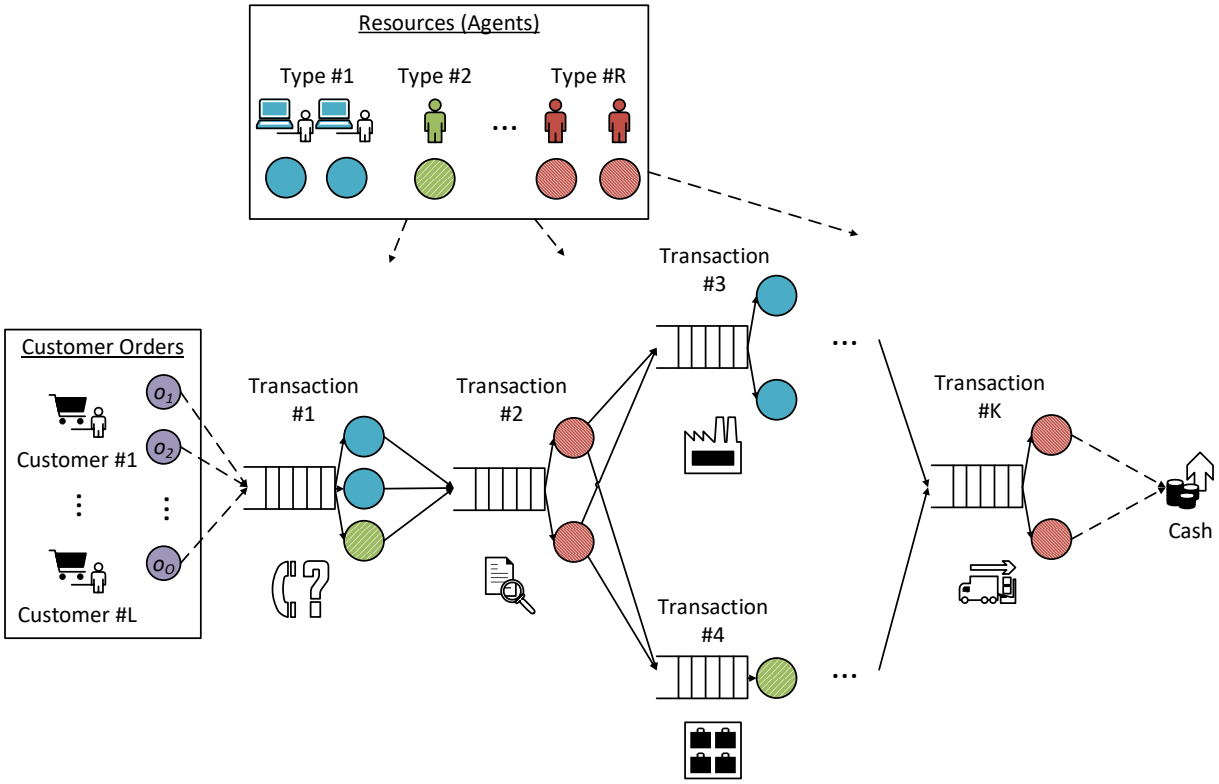

Figure 18. Sample network of transaction queues in the order-to-cash business process

A digital replica of the real processes is created from input data via a discrete event simulation that has an optimization loop as shown in **Figure 19**. The optimization loop runs as an event in the discrete event simulation that takes as its input the current state of the business process (e.g., active requests being executed and queued requests at the different transactions in the network), along with the transactional network structure, model parameters, request details, and resource mapping. An RTN model of the business process is built with these inputs to run a scheduling optimization that maximizes the number of orders that are fulfilled on time. The resulting schedule is then translated into queue priorities and any task preemptions, which get passed to the simulation. The simulation then continues running with the new prioritized queues and preemptions until the optimization loop is triggered again. The RTN model is updated with the new system state each time the optimization event is triggered, which can occur on a periodic basis or conditional to an event occurring (e.g., system disturbance). Note that the time required to solve the optimization is critical as this dictates the duration of the optimization event in the simulation. Long optimization run times may result in queue priorities that are no longer relevant if the simulation has moved significantly from the original state passed to the RTN model.

Stochasticity is introduced in the simulation via uncertainty in processing times, uncertainty in the occurrence of an exception that would require a request to take an alternate path in the network or to be recycled for reprocessing, uncertainty in request arrival dates, and uncertainty in a customer reneging. Initial case studies have shown that using an optimization loop that runs a transactional RTN has potential to bring substantial profit gains relative to an unoptimized first-in-first-out (FIFO) method for processing incoming orders.

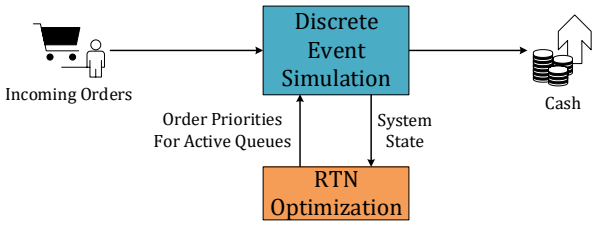

Figure 19. Integrated simulation-optimization framework

4. Industrial Impact

The RTN framework and its extensions have had a substantial impact at Dow As an example, the online reactive scheduling using the state space RTN has represented an annual increase in profit of over $5 million because of improved productivity and more stable operations. Since its implementation, batch wait times decreased by over 50%, operating capacity increased by 5%, and the number of manual plant interventions decreased by 75% (Hayot, 2017). The technological success of the application of RTN for integrated scheduling and dynamic modeling also won a Manufacturing Leadership Award by the Frost & Sullivan's Manufacturing Leadership Council in 2014.

5. Conclusions

An overview of the classical RTN modeling framework in both discrete and continuous time has been presented in the context of process scheduling optimization, where it has proven to be a mathematically compact model, especially for problems with identical processing units and multiple pathways. Extensions to the classical model structure have been described, including 1) the network structure for quality-based changeovers, 2) constraints for dealing with external material transfers with time windows, 3) constraints to ensure the fulfillment of orders in single shipments, 4) resource limit balances to mathematically simplify material transfers, 5) constraint modifications to allow for soft constraints (e.g., minor safety stock violations), and 6) model complexity reduction via multiple task extents.

As a result of the modeling flexibility of the RTN, creativity and expertise by the practitioner are often required when mapping existing problems onto the resource-task paradigm. For example, resources need to be properly defined, along with their interactions in the different stages of a process, which includes populating all the necessary modeling parameters and describing the system constraints. Although this may not always be as intuitive as in other modeling paradigms, once the mapping is performed, the RTN becomes a powerful tool for optimization. The computational advantages of the RTN can then be exploited to tackle many different problems, from traditional offline batch scheduling to reactive scheduling, spatial packing optimization, and business process optimization. At Dow, such novel applications of RTN have and are making a positive impact.

From an implementation standpoint, it is important to ensure that RTN model parameters can be seamlessly updated by the end user, who is often not the original model developer, to enable its continued use in practical applications. Attempts have been made in the past to develop software packages for batch scheduling using the RTN paradigm, but these were never commercialized (e.g., gBSS software (Shah, 1992)). Regardless of whether such a software package is ever developed, the RTN model is expected to continue to deliver value to many industries as well as in other applications within the chemical industry.

6. Acknowledgements

The authors gratefully acknowledge the support and funding from The Dow Chemical Company, and the Center for Advanced Process Decision-making at Carnegie Mellon University.

7. Nomenclature

Sets			
$i \in \mathcal{I}$	Tasks or package types	$r \in \mathcal{R}$	Resources
$i \in \mathcal{I}_r$	Tasks involving resource r	$r \in \mathcal{R}^{EQ}$	Equipment
$i \in \mathcal{I}^{PO}$	Point order tasks	$r \in \mathcal{R}^{P}$	Products
$i \in \mathcal{I}_r^{PO}$	Point order tasks involving material r	$r \in \mathcal{R}^{S}$	Shipped products
$i \in \mathcal{I}_r^{ST}$	Storage tasks involving material r	$r \in \mathcal{R}^{ST}$	Stored materials
$i \in \mathcal{I}^{TR}$	Changeover (transition) tasks	$r \in \mathcal{R}_o$	Materials requested in order o
$k \in \mathcal{K}_i$	Constraints on the multiple extent task i	$t \in \mathcal{T}$	Time points
$m \in \mathcal{M}_i$	Extents associated with task i	$x \in \mathcal{X}$	Horizontal coordinate points
$o \in O$	Orders	$x \in \mathcal{X}_i$	Horizontal resources consumed by task i
		$y \in \mathcal{Y}$	Vertical coordinate points

Parameters	
α_i	Constant processing time ratio for task i
β_i	Proportional processing time ratio for task i
$\gamma_{i,k,t}^{min}, \gamma_{i,k,t}^{max}$	Upper and lower bounds on constraint k associated with the multiple extents in task i at time point t
$\delta_{i,k,m,t}$	Coefficient for constraint k associated with extent m of task i at time point t
Δt	Maximum number of time periods allowed for a task
$\mu_{i,r,t}$	Discrete consumption/production ratio for resource r in task i at time point t
$\mu_{i,r}^{C}, \mu_{i,r}^{P}$	Discrete consumption and production ratios for resource r in task i, respectively
$\mu_{i,r}^{min}, \mu_{i,r}^{max}$	Discrete consumption/production ratios for the lower and upper limits of resource r in task i, respectively
$\nu_{i,r,t}$	Variable consumption/production ratio for resource r in task i at time point t
$\nu_{i,m,r,t}$	Variable consumption/production ratio for resource r in extent m of task i at time point t
$\nu_{i,r}^{C}, \nu_{i,r}^{P}$	Variable consumption and production ratios for resource r in task i, respectively
$\nu_{i,r}^{min}, \nu_{i,r}^{max}$	Variable consumption/production ratios for the lower and upper limits of resource r in task i at time point t, respectively
τ_i	Duration of task i
τ_d	Duration of delay d
c_r^{inv}	Inventory holding cost for material r
c_r^{min}, c_r^{max}	Penalty (cost) for violating lower and upper resource limits of resource r
c_r^{slack}	Penalty (cost) for not matching initial and end-of-horizon inventory levels of material r
c_i^{TR}	Cost for changeover (transition) task i
D_i	Final due date for point order task i
D_o	Final due date for order o
D_K	Distance of the king ping from the front of the semi-trailer (x-direction)

D_T	Distance of the tandem axle from the front of the semi-trailer (x-direction)
E_i	Early acceptance date for point order task i
E_o	Early acceptance date for order o
F_T^{max}	Maximum force that can be exerted on a tandem axle
h	Last time point in set $\mathcal{T}$
H	Scheduling horizon (i.e., T_h)
L_i	Length (x-direction) of a box of type i
m_i	Weight of a box of type i
p_r	Sales price of product r
$p_{o,r,t}$	Sales price of product r in order o that is delivered at time point t
$Q_{o,r}^{min}, Q_{o,r}^{max}$	Minimum and maximum quantities of material r requested in order o, respectively
Q_o^{min}, Q_o^{max}	Minimum and maximum quantities of materials in order o, respectively
$\dot{Q}_{i,m}^{min}, \dot{Q}_{i,m}^{max}$	Minimum and maximum production rates for extent m in task i.
V_i^{min}, V_i^{max}	Minimum and maximum limits on the extent for task i
$V_{i,m}^{min}, V_{i,m}^{max}$	Minimum and maximum limits on extent m in task i
W_i	Width (y-direction) of a box of type i
W^{max}	Maximum weight allowed on a semi-trailer
Continuous (non-negative) Variables	
ϕ	Objective function value
$\Pi_{r,t}$	External transfer of resource r at time point t
$\Pi_{r,t}^{in}, \Pi_{r,t}^{out}$	Incoming and outgoing external transfers of resource r at time point t, respectively
$\xi_{i,t}$	Extent of task i at time point t
$\xi_{i,m,t}$	Extent m for task i at time point t
$\xi_{i,m,t,\theta}$	Extent m for task i that has been actively running for θ time periods at time point t
$\xi_{i,t,t'}$	Extent of task i starting at time point t and ending at or before time point t'
F_K	Force exerted by the king pin on a semi-trailer
F_T	Force exerted by the tandem axle on a semi-trailer
I_r^{ave}	Time averaged inventory of material r
$R_{r,t}$	Resource inventory level for resource r at time point t
$R_{x,y}$	Horizontal floor space resource level at location x, y
R_r^{min}, R_r^{max}	Minimum and maximum limits for resource r, respectively
$R_{r,t}^{min}, R_{r,t}^{max}$	Minimum and maximum limits for resource r at time point t, respectively
$R_{r,t}^{min,slack}$	Violation of the lower limit of resource r at time point t
$R_{r,t}^{max,slack}$	Violation of the upper limit of resource r at time point t
R_r^{slack}	Violation of end-of-horizon inventory constraint of material r
T_t	Time at time point t
Discrete Variables	
$d_{i,t,\theta}$	Indicator of a one period delay at time point t on task i that has been actively running for θ time periods
$N_{i,t}$	Number of occurrences of task i starting at time point t
$N_{i,t,\theta}$	Number of occurrences of task i that have been actively running for θ time periods at time point t
$N_{i,t,t'}$	Occurrence of task i starting at time point t and ending at or before time point t'
$N_{i,x,y}$	Indicator that the bottom left corner of a box of type i has been placed with at location x, y

8. References

Akiya, N., Bury, S., Wassick, J.M., 2011. Generic Framework for Simulating Networks Using Rule-Based Queue and Resource-Task Network, in: Jain, S., Creasey, R.R., Himmelspach, J., White, K.P., Fu, M. (Eds.), Proceedings of the 2011 Winter Simulation Conference. IEEE, Phoenix, pp. 2194–2205.

Balas, E., Ceria, S., Cornuéjols, G., 1993. A lift-and-project cutting plane algorithm for mixed 0-1 programs. Mathematical Programming 58, 295–324. https://doi.org/10.1007/BF01581273

Bixby, R., Rothberg, E., 2007. Progress in computational mixed integer programming - A look back from the other side of the tipping point. Annals of Operations Research. https://doi.org/10.1007/s10479-006-0091-y

Brunaud, B., Perez, H.D., Amaran, S., Bury, S., Wassick, J., Grossmann, I.E., 2020. Batch scheduling with quality-based changeovers. Computers and Chemical Engineering 132. https://doi.org/10.1016/j.compchemeng.2019.106617

Castro, P., Barbosa-Póvoa, A.P.F.D., Matos, H., 2001. An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. Industrial and Engineering Chemistry Research 40, 2059–2068. https://doi.org/10.1021/ie000683r

Castro, P.M., Barbosa-Póvoa, A.P., Matos, H.A., Novais, A.Q., 2004. Simple Continuous-Time Formulation for Short-Term Scheduling of Batch and Continuous Processes. Industrial and Engineering Chemistry Research 43, 105–118. https://doi.org/10.1021/ie0302995

Chen, J.X., 2016. The Evolution of Computing: AlphaGo. Computing in Science and Engineering. https://doi.org/10.1109/MCSE.2016.74

Chen, V., 2019. Plenary: Planning Transportation Capacity at Amazon - 2019 INFORMS Annual Meeting | October 20-23, 2019 [WWW Document]. INFORMS Annual Meeting. URL http://meetings2.informs.org/wordpress/seattle2019/2019/10/23/plenary-planning-transportation-capacity-at-amazon/ (accessed 1.29.21).

Dakin, R.J., 1965. A tree-search algorithm for mixed integer programming problems. The Computer Journal 8, 250–255. https://doi.org/10.1093/comjnl/8.3.250

Duran, M.A., Grossmann, I.E., 1986. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Mathematical Programming 36, 307–339. https://doi.org/10.1007/BF02592064

Floudas, C.A., Lin, X., 2004. Continuous-time versus discrete-time approaches for scheduling of chemical processes: A review. Computers and Chemical Engineering 28, 2109–2129. https://doi.org/10.1016/j.compchemeng.2004.05.002

Geoffrion, A.M., 1972. Generalized Benders decomposition. Journal of Optimization Theory and Applications 10, 237–260. https://doi.org/10.1007/BF00934810

Georgiadis, G.P., Elekidis, A.P., Georgiadis, M.C., 2019. Optimization-based scheduling for the process industries: From theory to real-life industrial applications. Processes 7, 438. https://doi.org/10.3390/pr7070438

Glover, F., 1975. Improved linear integer programming formulations for nonlinear integer problems. Management Science 22, 455–460. https://doi.org/10.1287/mnsc.22.4.455

Gupta, V., Grossmann, I.E., 2012. An efficient multiperiod MINLP model for optimal planning of offshore oil and gas field infrastructure. Industrial and Engineering Chemistry Research 51, 6823–6840. https://doi.org/10.1021/ie202959w

Harjunkoski, I., Maravelias, C.T., Bongers, P., Castro, P.M., Engell, S., Grossmann, I.E., Hooker, J., Méndez, C., Sand, G., Wassick, J., 2014. Scope for industrial applications of production scheduling models and solution methods. Computers and Chemical Engineering 62, 161–193. https://doi.org/10.1016/j.compchemeng.2013.12.001

Hayot, P., 2017. Online Scheduling and Model-based Optimization: A Candid Practitioner View, in: Foundations of Computer Aided Process Operations / Chemical Process Control. Tucson.

Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., 2000. Progress in Linear Programming-Based Algorithms for Integer Programming: An Exposition. INFORMS Journal on Computing 12, 2–23. https://doi.org/10.1287/ijoc.12.1.2.11900

Kondili, E., Pantelides, C.C., Sargent, R.W.H., 1993. A general algorithm for short-term scheduling of batch operations-I. MILP formulation. Computers and Chemical Engineering 17, 211–227. https://doi.org/10.1016/0098-1354(93)80015-F

Lee, H., Maravelias, C.T., 2020. Combining the advantages of discrete- and continuous-time scheduling models: Part 3. General algorithm. Computers and Chemical Engineering 139, 106848. https://doi.org/10.1016/j.compchemeng.2020.106848

Maravelias, C., 2021. Chemical Production Scheduling: Mixed-Integer Programming Models and Methods, Cambridge Series in Chemical Engineering. Cambridge University Press, Cambridge. https://doi.org/DOI:

Marchand, H., Martin, A., Weismantel, R., Wolsey, L., 2002. Cutting planes in integer and mixed integer programming. Discrete Applied Mathematics 123, 397–446. https://doi.org/10.1016/S0166-218X(01)00348-1

Méndez, C.A., Cerdá, J., Grossmann, I.E., Harjunkoski, I., Fahl, M., 2006. State-of-the-art review of optimization methods for short-term scheduling of batch processes. Computers and Chemical Engineering. https://doi.org/10.1016/j.compchemeng.2006.02.008

Nie, Y., Biegler, L.T., Wassick, J.M., Villa, C.M., 2014. Extended discrete-time resource task network formulation for the reactive scheduling of a mixed batch/continuous process. Industrial and Engineering Chemistry Research 53, 17112–17123. https://doi.org/10.1021/ie500363p

Ostrowski, J., Anjos, M.F., Vannelli, A., 2012. Formulations for the Unit Commitment Problem. IEEE Transactions on Power Systems 27, 39–46. https://doi.org/10.1109/TPWRS.2013.2251373

Pantelides, C.C., 1994. Unified frameworks for optimal process planning and scheduling, in: Proceedings on the Second Conference on Foundations of Computer Aided Operations. pp. 253–274.

Perez, H.D., Amaran, S., Erisen, E., Wassick, J.M., Grossmann, I.E., 2021a. A Digital Twin Framework for Business Transactional Processes in Supply Chains. Submitted for publication.

Perez, H.D., Amaran, S., Erisen, E., Wassick, J.M., Grossmann, I.E., 2021b. Optimization of Extended Business Processes in Digital Supply Chains using Mathematical Programming. Submitted for publication.

Shah, N., 1992. Efficient scheduling, planning and design of multipurpose batch plants. Imperial College London (University of London), London.

Shah, N., Pantelides, C.C., Sargent, R.W.H., 1993. A general algorithm for short-term scheduling of batch operations-II. Computational issues. Computers and Chemical Engineering 17, 229–244. https://doi.org/10.1016/0098-1354(93)80016-G

Sridhar, S., Linderoth, J., Luedtke, J., 2013. Locally ideal formulations for piecewise linear functions with indicator variables. Operations Research Letters 41, 627–632. https://doi.org/10.1016/j.orl.2013.08.010

Su, L., Tang, L., Bernal, D.E., Grossmann, I.E., 2018. Improved quadratic cuts for convex mixed-integer nonlinear programs. Computers and Chemical Engineering 109, 77–95. https://doi.org/10.1016/j.compchemeng.2017.10.011

Wassick, J.M., Ferrio, J., 2011. Extending the resource task network for industrial applications. Computers and Chemical Engineering 35, 2124–2140. https://doi.org/10.1016/j.compchemeng.2011.01.010

Westerlund, T., Pettersson, F., 1995. An extended cutting plane method for solving convex MINLP problems. Computers and Chemical Engineering 19, 131–136. https://doi.org/10.1016/0098-1354(95)87027-X