Optimization of Business Process Operations in a Digital Supply Chain using Mathematical Programming

Hector D. Perez[a], Ignacio E. Grossmann[a]*, Satyajith Amaran[b], Esra Erisen[b], John M. Wassick[b]

[a]*Carnegie Mellon University, Pittsburgh 15213, USA*

[b]*The Dow Chemical Company, Midland 48674, USA*

**grossmann@cmu.edu*

Abstract

We propose a mathematical programming approach to optimize the business process transactions in a digital supply chain. Five scheduling models from the Process Systems Engineering (PSE) community are applied to schedule the processing of orders in the Order-To-Cash (OTC) business process, which is modeled as a multistage network with parallel units (agents). Three case studies are provided to compare the performance of the scheduling models on various sizes of the OTC process. The models are compared and scaled to select those that are more suitable to this application. The continuous-time general precedence model provides an accurate representation of the real system and performs well for small instances. The discrete-time State-Task Network (STN), however, proves best in terms of tractability, despite the well-known limitations resulting from approximating time with a discrete grid. The tightness of the linear programming (LP) relaxations in the discrete-time STN framework enables finding near optimal solutions rapidly during the optimization runs, even for larger instances.

Keywords: Business process optimization, digital supply chain, order-to-cash, scheduling, mathematical programming

1. Introduction

As industry more fully embraces the fourth industrial revolution, supply chains are undergoing a digital transformation (Supply Chain 4.0) [1] [2]. This transformation has increased the flow and transparency of information within the supply chain. At the corporate level, supply chains are viewed from the lens of the business processes, such as the procure-to-pay and order-to-cash processes, which store the key events and transactions associated with the different stages in the supply chain. In a digital supply chain, business processes are managed by Enterprise Resource Planning (ERP) systems and Blockchain platforms. In every business transaction, at least one of the key flows in a supply chain is involved (information, materials, or resources). Any inefficiency in the management of these business processes results in inefficient supply chain operations. Current efforts to detect and manage these inefficiencies have been made from the business process management (BPM) side.

Existing methodologies in business process optimization have been mainly led by the computer science and systems communities. The focus has been on the cyclic procedure of designing, implementing, analyzing, and improving business processes [3]. Key elements in business process management include process identification, process modeling, process discovery, process analysis (qualitative and quantitative), process redesign, process automation, and process intelligence [4] [5]. In the area of process modeling, Samaranayake proposes enhanced business process models (integrated process/data models) that provide greater visibility and flexibility upon which to optimize business processes in ERP systems [6].

Niedermann, *et al.* developed an optimization framework based on graph modeling and data mining to identify and propose structural changes in process designs that improve process performance, given a desired goal function [7]. Other studies apply evolutionary algorithms to automate the design and redesign of business processes [8] [9] [10]. Gröger, *et al.* presents a real-time recommender system for business process optimization based on data-mining (decision trees) [11]. In this work, process performance forecasts are used to recommending operational changes that should be made during process execution to avoid undesired outcomes. In the area of process intelligence and automation, industrial solutions are being offered, which combine process mining with robotic process automation (RPA) to identify process steps with the most automation potential [12]. Krumeich, *et al.* propose a framework that applies complex event processing and event-based predictions that can be used to prescribe control actions during the execution of business processes [13]. Thus, the BPM field presents diverse techniques to improve the performance of supply chains from what Shapiro calls the “bottom-up” approach [14]. This approach seeks to improve supply chains by improving the design and execution of business process using manual inspection or data-driven models.

In the processing industry, there exists a large community that uses rigorous mathematical programming techniques to optimize manufacturing processes in what Shapiro refers to as the “top-down” approach to supply chain management [14]. This top-down approach focusses on the modeling and optimization of the material and resource flows within the different levels of an enterprise supply chain. The process systems engineering (PSE) and operations research (OR) communities have extensively studied and applied mathematical programming to optimize the different operational levels within manufacturing systems, from the second-to-second plant operations level up to the year-to-year supply chain planning level. This multi-tier (planning, scheduling, and control) approach to optimization is referred to as Enterprise-Wide Optimization (EWO) by the PSE community. In 2012, Grossmann presented an overview of the developments in mathematical programming techniques for EWO with applications in the petrochemical and manufacturing industries [15]. At the scheduling level, Harjunkoski, *et al.* describe the optimization methodologies available and present successful applications of these in the dairy, petrochemical, and pulp and paper industries [16]. Shah describes the advances in optimization applied to the areas of supply chain network design, simulation, policy analysis, and planning [17].

Recent trends in EWO in the PSE community indicate a shift towards an integration of the different aspects of the supply chain. Mota, *et al.* use multi-objective mathematical programming to design and plan sustainable supply chains from a holistic viewpoint that includes economic, environmental, and societal considerations [18]. Varma, *et al.* present the need for a comprehensive approach towards EWO, one that targets the cross-functional coordination among the components of the enterprise and couples operational and design decisions with financial models [19]. Laínez and Puigjaner build on this concept of integrated SCM, discussing the need to account for the different supply chain flows (material, financial, and information) and combine data sources with mathematical models to build flexible decision support systems [20]. In line with this recent shift in EWO, we propose integrating business processes within EWO. As an initial step towards coupling the top-down and bottom-up SCM approaches, we approach business process optimization via mathematical programming. To the knowledge of the authors, no previous work exists describing the use of mathematical programming techniques to optimize supply chains at the business process transactional level.

In this paper, we apply mathematical programming scheduling models to optimize the execution of business processes. Scheduling models can capture the logic and structure of business processes upon which their operations can be optimized. Several scheduling modeling paradigms based on mixed integer models have been presented in the PSE literature [21] [22] [23]. These modeling paradigms are used to optimize the sequence of events in a production process while targeting a desired objective (maximum profit, minimum cost, minimum timespan, etc.). Using mathematical programming brings the benefit of using advanced solvers to exploit the structure of the system being modelled and achieve not just an improvement but a guarantee of optimality subject to the model of the system. Thus, this work complements the work presented in the BPM literature by using model-based optimization to find the best sequence and allocation of resources during the execution of business processes.

In this paper, business transactional processes are presented as multiperiod multistage network systems (**Section 2**). For illustrative purposes, the modeling approach is applied specifically to the order-to-cash (OTC or O2C) process. Different scheduling paradigms from the PSE community are applied to the OTC process (**Section 3**). These include, the time slot [24] and general precedence [25] continuous-time models, and both discrete-time and continuous-time network models (state-task [26] and resource-task [27] models). The performance of these models is compared in terms of their result quality and tractability via a series of case studies (**Sections 4-5**). Concluding remarks are summarized in **Section 6.**

2. Problem Description

The OTC business process describes the sequence of events that occur between when a customer places an order and invoice payment for delivered goods is received. A high-level overview of the events in this business process is given in **Figure 1**. The process can be modeled as a sequence of tasks (stages) that need to be executed on each order that is placed by a customer. One or more agents (human or robots) are assigned to each stage of the OTC process, creating a multistage network with parallel units (agents). As orders enter the system, they are assigned to the queue of one of the agents in the first stage. Once the task associated with the first stage is executed on an order, the order moves downstream to an agent's queue in the second stage. The process continues until all sequential tasks in the OTC process are performed on each order. The assignment of agents to stages dictates if the system behaves as a flow shop or as a job shop. When agents are assigned to only one stage in the process, the system can be modeled as a flow shop (**Figure 2**). On the other hand, the use of flexible agents, meaning agents being able to perform tasks belonging to more than one stage, results in a job shop configuration (**Figure 3**).

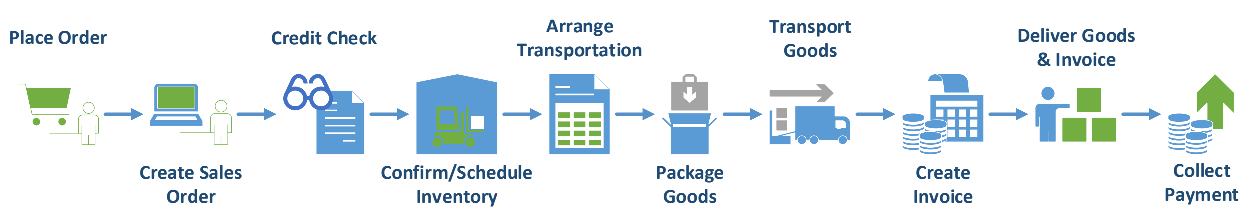

Figure 1. Overview of OTC process

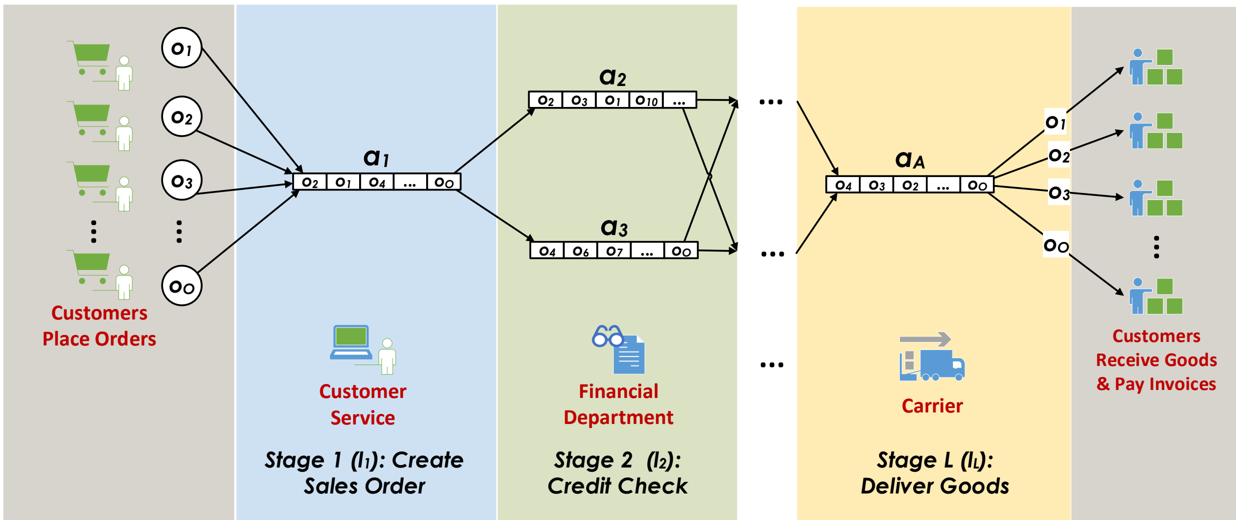

Figure 2. Flow shop sample configuration for OTC process with queues depicted for each agent

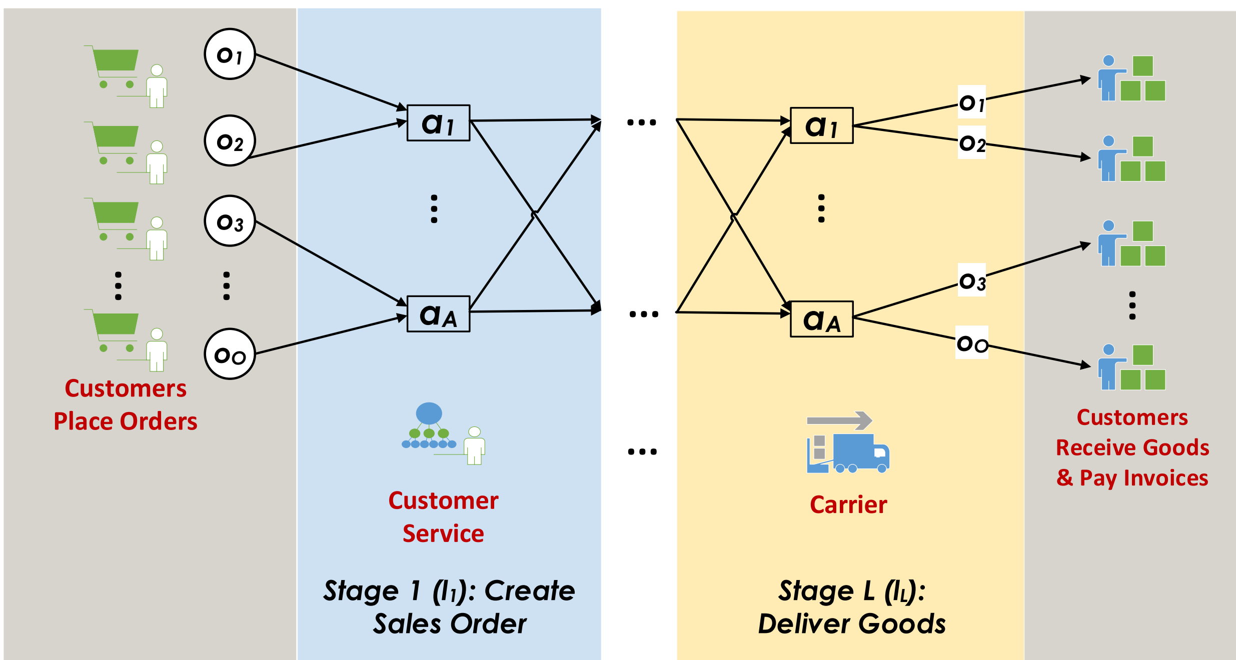

Figure 3. Job shop configuration for OTC process

For initial modeling purposes, the following parameters are assumed to be known a priori,

- Expected processing times for each agent at each stage.
- Release dates for each order (date and time when an order is placed by the customers).
- Due dates for each order.

Another major assumption in the initial modeling is that once a task is performed on an order, it is done so correctly so that reprocessing is not required. Additionally, external disturbances that could result in order reprocessing are ignored.

Due to the non-material nature of orders and transactions, the following features, which simplify the modeling from a process scheduling approach, can be introduced,

- No transition times are required between stages. In contrast to physical processes, in which materials need to be transferred from one stage to the next, flow of information between stages is instantaneous. This is due to the fact that information related to the business processes is stored in the ERP or Blockchain systems and is visible to all processing agents in the enterprise.
- No wait times are required in an agent's queue. In manufacturing processes, cleaning or changeover steps are usually required between executions of different batches of materials. In transactional processes, an agent can process the next order in its queue immediately after it finishes processing its successor.

3. Scheduling Models

Five models are described below for scheduling the multistage OTC network: 1) continuous-time queue slot model, 2) continuous-time general precedence model, 3) continuous-time resource-task network, 4) discrete-time resource-task network, and 5) discrete-time state-task network. It should be noted that in the formulations, Boolean variables and binary variables are used interchangeably for simplicity.

3.1. Nomenclature

	Description
Set	
$a \in A$	Agents: $\{1, \dots, \lvert A \rvert\}$.
$o \in O$	Orders: $\{1, \dots, \lvert O \rvert\}$.
$l \in L$	Stages: $\{1, \dots, \lvert L \rvert\}$.
$p \in P$	Queue positions (slots): $\{1, \dots, \lvert P \rvert\}$.
$t \in TP$	Timepoints: $\{1, \dots, \lvert TP \rvert\}$.
$a \in A_l$	Agents assigned to stage l.
$p \in P_a$	Queue positions (slots) in agent a's queue.
Parameters	
c_o	Backlog cost (penalty) of order$o$.
$E_{o,t}$	(Binary) order o enters the system at time point t.
ϵ	Small epsilon value to reformulate strict inequalities.
h	Scheduling horizon.
p_o	Revenue of order o.
$t_o^{r(d)}$	Release (due) date for order o.
$\overline{t_o^r}, \underline{t_o^d}$	Rounded release dates and due dates for order o in terms time points.
$\tau_{o,l,a}$	Processing time for order o in stage l by agent a.
$\tau'_{o,l,a}$	Rounded processing times in terms of time points.
Continuous (non-negative) Variables	
$AR_{a,t}$	Agent resource for agent a at timepoint t.
$OR_{o,l,t}$	Order resource for order o at stage l and timepoint t.
T_t	Time value at timepoint t.
$T_{o,l}^{s(f)}$	Start (finish) time of order o at stage l.
$T_{o,l,a,p}^{s(f)}$	Start (finish) time of order o at stage l in position p of agent a's queue.

Binary Variables	
B_o	1(0) = order o is (not) backlogged.
B_o'	Auxiliary variable for backlog logic in continuous-time models.
$D_{o,t}$	1(0) = order o is (not) delivered at timepoint t. Can be treated as continuous.
$E_{o,t}$	1(0) = order o enters (does not enter) the system in the interval $(t-1,t]$ in continuous-time RTN.
$E'_{o,t}$	Auxiliary variable for order entry logic in continuous-time RTN.
F_o	1(0) = order o is fulfilled. Can be treated as continuous in RTN.
$W_{o,l,a}$	Order o is assigned to agent a for executing task l.
$W_{o,l,a,p}$	Order o is assigned to position p in agent a's queue for executing task l.
$W_{o,l,a,t}$	Task l is performed on order o by agent a at timepoint t.
$W_{o,l,a,t,t'}$	Task l is performed on order o by agent a, starting at timepoint t and ending on or before timepoint t'.
$X_{o,l,o',l'}$	Task l is performed on order o before task l' is performed on order o'.

3.2. Continuous-time queue slot model

The queue slot model is a variation of the time slot model presented by Pinto and Grossmann [24]. Each agent has a queue with slot positions (**Figure 4**). Each slot has an associated start and end time. Agents process orders in the order of the slots in their queue. The number of slots in each agent's queue can be set to its upper bound of $|O| \cdot |L_a|$, where L_a is the stages assigned to agent a. Unlike, the Pinto and Grossmann model that uses two types of time variables, a single time variable that associates an order at a stage to a queue position is used. The queue slot model is described by the following constraints.

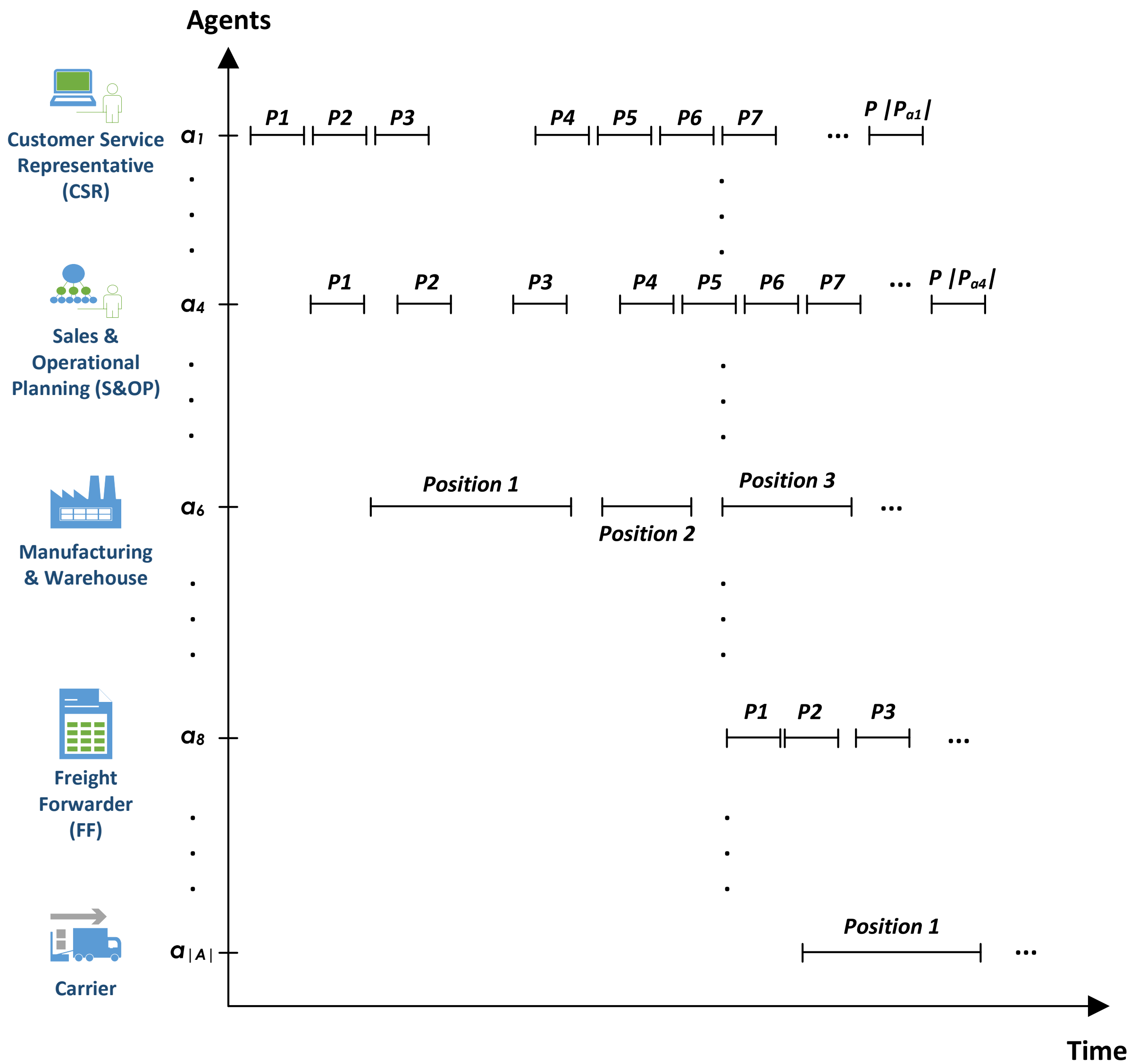

Figure 4. Queue slot representation for the OTC process

3.2.1. Constraints

Time Bounds: When order o is assigned to slot p in agent a's queue, its start ($T^s_{o,l,a,p}$) and end times ($T^f_{o,l,a,p}$) must be bounded by the order's release time (t^r_o) and the scheduling horizon (h). Each end time occurs $\tau_{o,l,a}$ time units after its corresponding start time. This is formulated with the disjunction in (1). The disjunction is reformulated into (2) using the convex hull method, which is the tightest algebraic formulation.

$$\begin{bmatrix} W_{o,l,a,p} \\ t_o^r \leq T_{o,l,a,p}^s \leq h \\ T_{o,l,a,p}^f = T_{o,l,a,p}^s + \tau_{o,l,a} \\ T_{o,l,a,p}^f \leq h \end{bmatrix} \vee \begin{bmatrix} \neg W_{o,l,a,p} \\ T_{o,l,a,p}^s = 0 \\ T_{o,l,a,p}^f = 0 \end{bmatrix} \qquad \forall o \in O, l \in L, a \in A_l, p \in P_a \tag{1}$$

$$\left.\begin{matrix} W_{o,l,a,p} \cdot t_o^r \leq T_{o,l,a,p}^s \\ T_{o,l,a,p}^f = T_{o,l,a,p}^s + \tau_{o,l,a} \cdot W_{o,l,a,p} \\ 0 \leq T_{o,l,a,p}^s \leq h \cdot W_{o,l,a,p} \\ 0 \leq T_{o,l,a,p}^f \leq h \cdot W_{o,l,a,p} \end{matrix}\right\} \qquad \forall o \in O, l \in L, a \in A_l, p \in P_a \tag{2}$$

Assignment Constraints: (3) enforces that each order o is processed at most once at each stage l. In (4), at most one order can occupy position p in agent a's queue.

$$\sum_{a \in A_l} \sum_{p \in P_a} W_{o,l,a,p} \leq 1 \qquad \forall o \in O, l \in L \tag{3}$$

$$\sum_{o \in O} \sum_{l \in L} W_{o,l,a,p} \leq 1 \qquad \forall a \in A, p \in P_a \tag{4}$$

Queue Precedence Constraints: Subsequent slot positions p and $p+1$ in agent a's queue must be filled sequentially, as enforced by the logical proposition in (5), which is reformulated into a linear constraint that is strengthened via (4) in (6). Proposition (7) ensures that the start time in slot $p+1$ must occur after the end time of slot p. (7) is reformulated using Big-M and strengthened using (4) in (8).

$$\bigvee_{o \in O} \bigvee_{l \in L} W_{o,l,a,p+1} \Rightarrow \bigvee_{o \in O} \bigvee_{l \in L} W_{o,l,a,p} \qquad \forall a \in A, p \in P_a, p < |P_a| \tag{5}$$

$$\sum_{o \in O} \sum_{l \in L} W_{o,l,a,p+1} \leq \sum_{o \in O} \sum_{l \in L} W_{o,l,a,p} \qquad \forall a \in A, p \in P_a, p < |P_a| \tag{6}$$

$$\bigvee_{o \in O} \bigvee_{l \in L} W_{o,l,a,p+1} \Rightarrow \left\{ \sum_{o \in O} \sum_{l \in L} T_{o,l,a,p}^f \leq \sum_{o \in O} \sum_{l \in L} T_{o,l,a,p+1}^s \right\} \qquad \forall a \in A, p \in P_a, p < |P_a| \tag{7}$$

$$\sum_{o \in O} \sum_{l \in L} T_{o,l,a,p}^f \leq \sum_{o \in O} \sum_{l \in L} T_{o,l,a,p+1}^s + h \cdot \left(1 - \sum_{o \in O} \sum_{l \in L} W_{o,l,a,p+1} \right) \qquad \forall a \in A, p \in P_a, p < |P_a| \tag{8}$$

Agent Precedence Constraints: Proposition (9) ensures that stages in the network are processed sequentially. (9) is reformulated and strengthened via (3) in (10). Proposition (11) enforces that the start time of an order in a stage $l+1$ occurs after the end time of that order in stage l. (11) is reformulated using Big-M and strengthened using (3) in (12).

$$\bigvee_{a\in A_{l+1}} \bigvee_{p\in P_a} W_{o,l+1,a,p} \Rightarrow \bigvee_{a\in A_l} \bigvee_{p\in P_a} W_{o,l,a,p} \qquad \forall o \in O, l \in L, l < |L| \tag{9}$$

$$\sum_{a\in A_{l+1}} \sum_{p\in P_a} W_{o,l+1,a,p} \leq \sum_{a\in A_l} \sum_{p\in P_a} W_{o,l,a,p} \qquad \forall o \in O, l \in L, l < |L| \tag{10}$$

$$\bigvee_{a\in A_{l+1}} \bigvee_{p\in P_a} W_{o,l+1,a,p} \Rightarrow \left\{ \sum_{a\in A_l} \sum_{p\in P_a} T^f_{o,l,a,p} \leq \sum_{a\in A_{l+1}} \sum_{p\in P_a} T^s_{o,l+1,a,p} \right\} \qquad \forall o \in O, l \in L, l < |L| \tag{11}$$

$$\sum_{a\in A_l} \sum_{p\in P_a} T^f_{o,l,a,p} \leq \sum_{a\in A_{l+1}} \sum_{p\in P_a} T^s_{o,l+1,a,p} + h \cdot \left(1 - \sum_{a\in A_{l+1}} \sum_{p\in P_a} W_{o,l+1,a,p} \right) \qquad \forall o \in O, l \in L, l < |L| \tag{12}$$

Order Fulfillment: Orders are fulfilled if and only if they are processed in the last stage, $|L|$.

$$F_o = \sum_{a\in A_{|L|}} \sum_{p\in P_a} W_{o,|L|,a,p} \qquad \forall o \in O \tag{13}$$

Order Backlog: Three scenarios are possible when determining if an order was backlogged,

1. If an order is not fulfilled, it is backlogged.
2. If an order is fulfilled late, it is backlogged.
3. If an order is fulfilled on time, it is not backlogged

(14) is reformulated using the Big-M method to obtain (15). The auxiliary variable B'_o equates to an order being fulfilled late ($\sum_{a\in A_{|L|}} \sum_{p\in P_a} T^f_{o,|L|,a,p} > t^d_o$). ϵ is a small value (e.g. 0.001) to reformulate strict inequalities. The tightest Big-M values for (15) are $M_0 = t^d_o + \epsilon$ and $M_1 = h - t^d_o$.

$$\left. \begin{array}{rcl} \neg F_o & \Rightarrow & B_o \\ \left\{ \sum_{a\in A_{|L|}} \sum_{p\in P_a} T^f_{o,|L|,a,p} > t^d_o \right\} & \Rightarrow & B_o \\ \left\{ \sum_{a\in A_{|L|}} \sum_{p\in P_a} T^f_{o,|L|,a,p} \leq t^d_o \right\} \wedge F_o & \Rightarrow & \neg B_o \end{array} \right\} \qquad \forall o \in O \tag{14}$$

$$\left. \begin{array}{c} -M_0 \cdot (1 - B'_o) + t^d_o + \epsilon \leq \sum_{a\in A_{|L|}} \sum_{p\in P_a} T^f_{o,|L|,a,p} \leq t^d_o + M_1 \cdot B'_o \\ 1 - F_o \leq B_o \\ B'_o \leq B_o \\ B_o \leq B'_o + (1 - F_o) \end{array} \right\} \qquad \forall o \in O \tag{15}$$

Variable Domains: The domains of the variables are given in (16)-(19).

$$0 \le F_o \le 1 \qquad \forall o \in O \qquad (16)$$

$$0 \le T^s_{o,l,a,p}, T^f_{o,l,a,p} \le h \qquad \forall o \in O, l \in L, a \in A_l, p \in P_a \qquad (17)$$

$$B_o, B'_o \in \{0,1\} \qquad \forall o \in O \qquad (18)$$

$$W_{o,l,a,p} \in \{0,1\} \qquad \forall o \in O, l \in L, a \in A_l, p \in P_a \qquad (19)$$

3.2.2. Objective Function

The objective is to maximize profit, which is the sales revenue for fulfilled orders minus the losses due to backlogs (assuming that backlogged orders are not lost sales). p_o is the price of order o and c_o is the backlog cost, which gives an indication of the order priority resulting from customer segmentation. This objective function is used for all models in this paper.

$$\max \sum_{o \in O} p_o \cdot F_o - \sum_{o \in O} c_o \cdot B_o \qquad (20)$$

The continuous-time queue slot model is given by (2)-(4), (6), (8), (10), (12)-(13), and (15)-(20).

3.3. General precedence model

In the general precedence model [25], all precedence relations in agents' queues are considered as agents are assigned to specific tasks on specific orders.

3.3.1. Constraints

Timing and sequencing: As in the queue slot model, the finish time for order o in stage l is equal its start time plus the duration of the task being executed as given in (21). Orders in downstream stages (l) cannot start until the execution of the previous stage ($l-1$) has finished as enforced in (22). The general sequencing of events in agent a is controlled by the disjunction in (23). If an agent is assigned to the order-stage tuples (o, l) and (o', l') such that o and o' are different, the precedence of these tuples in an agent's queue is governed by $X_{o,l,o',l'}$. The disjunction (23) is reformulated into (24) using Big-M, where $M_{2'} = h - t^r_{o'}$ and $M_2 = h - t^r_o$. (25) ensures that if an order is processed at a later stage $l+1$, it has already been processed in its previous stage l.

$$T^f_{o,l} = T^s_{o,l} + \sum_{a \in A_l} \tau_{o,l,a} \cdot W_{o,l,a} \qquad \forall o \in O, l \in L \qquad (21)$$

$$T^f_{o,l-1} \le T^s_{o,l} \qquad \forall o \in O, l \in L, l > 1 \qquad (22)$$

$$\begin{bmatrix} W_{o,l,a} \wedge W_{o',l',a} \\ \begin{bmatrix} X_{o,l,o',l'} \\ T^f_{o,l} \le T^s_{o',l'} \end{bmatrix} \vee \begin{bmatrix} \neg X_{o,l,o',l'} \\ T^f_{o',l'} \le T^s_{o,l} \end{bmatrix} \end{bmatrix} \vee \begin{bmatrix} \neg\left(W_{o,l,a} \wedge W_{o',l',a}\right) \\ t^r_{o'} \le T^s_{o',l'} \\ t^r_o \le T^s_{o,l} \end{bmatrix} \qquad (23)$$

$$\forall o, o' \in O, l, l' \in L, a \in A_l \cap A_{l'}, o \neq o'$$

$$\left.\begin{array}{r}-M_{2'}\cdot\left(1-X_{o,l,o',l'}\right)-M_{2'}\cdot\left(2-W_{o,l,a}-W_{o',l',a}\right)+T^{f}_{o,l}\le T^{s}_{o',l'}\\ -M_2\cdot X_{o,l,o',l'}-M_2\cdot\left(2-W_{o,l,a}-W_{o',l',a}\right)+T^{f}_{o',l'}\le T^{s}_{o,l}\end{array}\right\} \quad (24)$$
$$\forall o,o'\in O,l,l'\in L,a\in A_l\cap A_{l'},o\neq o'$$

$$\sum_{a\in A_{l+1}} W_{o,l+1,a}\le \sum_{a\in A_l} W_{o,l,a} \qquad \forall o\in O,l\in L,l<|L| \quad (25)$$

Assignment constraints: At most one agent can be assigned to order o at stage l. In typical general precedence formulations, constraint (26) is formulated as an equality. However, an inequality is used in this model since $W_{o,l,a}$ will be 0 for unfulfilled orders.

$$\sum_{a\in A_l} W_{o,l,a}\le 1 \qquad \forall o\in O,l\in L \quad (26)$$

Order Fulfillment: Orders are fulfilled if and only if they are processed in the last stage, $|L|$.

$$F_o=\sum_{a\in A_{|L|}} W_{o,|L|,a} \qquad \forall o\in O \quad (27)$$

Backlog Calculation: The same logic used in the queue slot model to determine if an order is backlogged is used in the general precedence model. The propositions in (28) are formulated into (29) as in the queue slot model, with a tighter Big-M value of $M_{0'}=t^d_o+\epsilon+t^r_o$.

$$\left.\begin{array}{rcl}\neg F_o & \Rightarrow & B_o\\ \left\{T^{f}_{o,|L|}>t^d_o\right\} & \Rightarrow & B_o\\ \left\{T^{f}_{o,|L|}\le t^d_o\right\}\wedge F_o & \Rightarrow & \neg B_o\end{array}\right\} \qquad \forall o\in O \quad (28)$$

$$\left.\begin{array}{c}-M_{0'}\cdot(1-B'_o)+t^d_o+\epsilon\le T^{f}_{o,|L|}\le t^d_o+M_1\cdot B'_o\\ 1-F_o\le B_o\\ B'_o\le B_o\\ B_o\le B'_o+(1-F_o)\end{array}\right\} \qquad \forall o\in O \quad (29)$$

Variable Domains: The domains of the variables are given in (30)-(34).

$$0\le F_o\le 1 \qquad \forall o\in O \quad (30)$$

$$t^r_o\le T^{s}_{o,l},T^{f}_{o,l}\le h \qquad \forall o\in O,l\in L \quad (31)$$

$$B_o,B'_o\in\{0,1\} \qquad \forall o\in O \quad (32)$$

$$W_{o,l,a}\in\{0,1\} \qquad \forall o\in O,l\in L,a\in A_l \quad (33)$$

$$X_{o,l,o',l'}\in\{0,1\} \qquad \forall o,o'\in O,l,l'\in L \quad (34)$$

The continuous-time queue slot model is given by (20) and (21)-(34).

3.4. Resource-task Network (RTN)

The RTN [27] formulation for this problem consists of parallel subnetworks for each order in the system. There are two types of resources in this framework, order resources (OR) and agent resources (AR). Order resources represent the presence of an order in between stages (i.e. waiting to be processed). Since each order is unique, these resources take on values of 0 or 1. An agent resource with a non-zero value represents the availability of an agent to process orders. Agents are shared resources that join each order subnetwork (**Figure 5**). Tasks are numbered by the tuple (o, l, a), where o is the order being processed, l is the stage being executed, and a is the agent that processes the task. Arrows connecting agent resources to tasks are double sided to indicate that an agent resource is consumed (occupied) at the beginning of the task and produced (released) at the end of the task. The structure in **Figure 5** can be thought of as a superstructure as in reality not all agents can or are allowed to perform all tasks. This superstructure allows for each task processing time to be dependent on the order (i.e. order type), stage, and agent. Under this network structure, each agent resource takes on a value of 0 or 1. However, if a subset of agents is assigned to the same set of stages and the processing times at each stage are the same for all agents in the subset, these tasks and agent resources can be aggregated, reducing the model complexity. Aggregated agent resource values will then take on non-negative integer values. In the extreme case that all agents can process all tasks and the processing times are agent independent, the network shown in **Figure 6** is obtained. Due to the formulation of the resource balances in the RTN framework, it is not necessary to declare the resources as discrete variables. However, doing so can lead to some computational advantages during presolve [28].

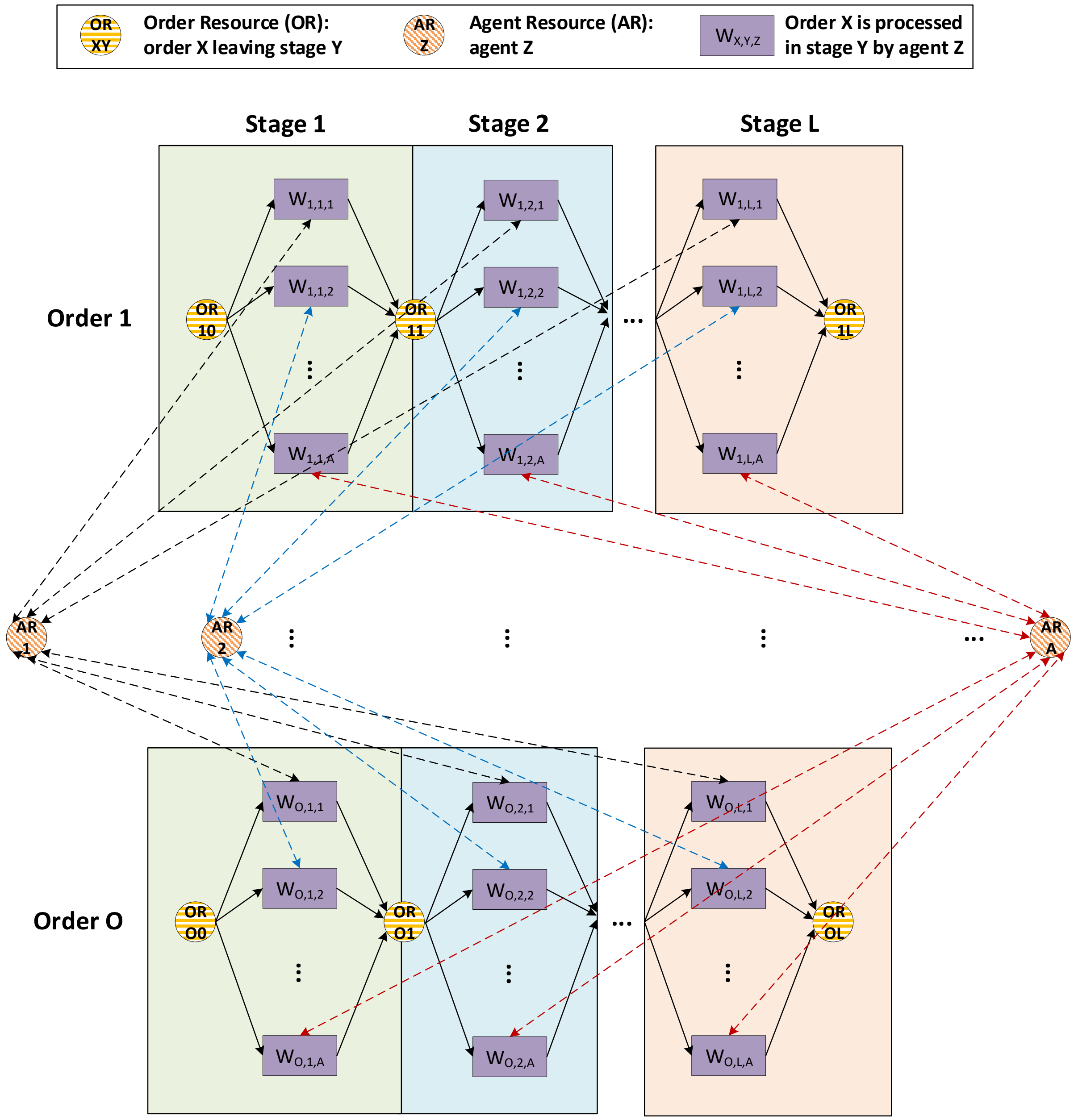

Figure 5. Generalized RTN structure for the OTC process

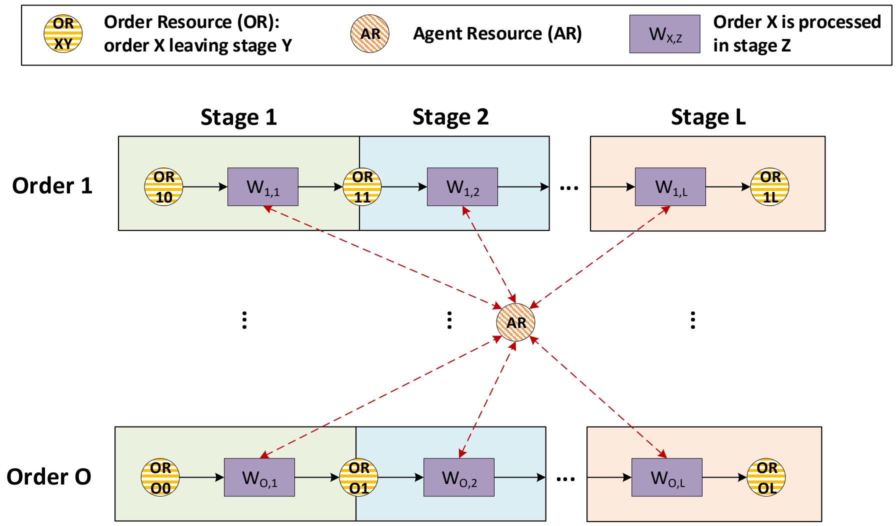

Figure 6. Aggregated RTN structure for the OTC process

The RTN model can be formulated in both continuous-time with general timepoints and discrete-time. Each formulation is given below,

3.4.1. Continuous-time Formulation

Timing and Sequencing: The time difference between to any two timepoints, $t < t'$, must be at least equal to the processing time of a task starting at t and ending at or before t'. This is expressed in the disjunction (35). $W_{o,l,a,t,t'}$ is a Boolean variable that indicates if executing stage l on order o by agent a begins at time t and ends in the interval $(t'-1, t']$. The processing time (task duration) of order o in stage l by agent a is given by $\tau_{o,l,a}$. Since an agent a can process at most one order o at a time (i.e. $\sum_{o\in O}\sum_{l\in L} W_{o,l,a,t,t'} \leq 1\ \forall a, t < t'$), the constraint can be reformulated using the convex hull and strengthened by aggregating the processing times for all orders executed by agent a between t and t' to obtain (36) [29].

$$\begin{bmatrix} W_{o,l,a,t,t'} \\ T_{t'} - T_t \geq \tau_{o,l,a} \end{bmatrix} \vee \begin{bmatrix} \neg W_{o,l,a,t,t'} \\ T_{t'} \geq 0 \\ T_t \geq 0 \end{bmatrix} \qquad \forall o \in O, l \in L, a \in A_l, t, t' \in TP, t < t' \tag{35}$$

$$T_{t'} - T_t \geq \sum_{o\in O}\sum_{l\in L} \tau_{o,l,a} \cdot W_{o,l,a,t,t'} \qquad \forall a \in A, t, t' \in TP, t < t' \tag{36}$$

Agent Resource Balance: The availability of an agent a to perform a task at timepoint t ($AR_{a,t}$) depends on its availability at the previous timepoint $t-1$ and on whether a task begins execution by that agent or the agent has been released from a previous task. The initial condition $AR_{a,0} = 1\ \ \forall a \in A$ is used since all agents are available at the start of the scheduling horizon.

$$AR_{a,t} = AR_{a,t-1} - \sum_{o\in O}\sum_{l\in L}\sum_{t'\in TP} W_{o,l,a,t,t'}|_{t<t'} + \sum_{o\in O}\sum_{l\in L}\sum_{t'\in TP} W_{o,l,a,t',t}|_{t'<t} \qquad \forall a \in A, t \in TP \tag{37}$$

Order Resource Balance: The presence of an order resource for order o after stage l at timepoint t is given by its availability at the previous timepoint $t-1$, the execution of the task for stage $l+1$ (resource is consumed), and the completion of the task for stage l (resource is produced), as formulated in (38). It should be noted that an order resource indexed with $l=0$ refers to the input node to the order's subnetwork (i.e. the order has not yet been processed by stage 1). The initial condition $OR_{o,l,0}=0\ \ \forall o\in O, l\in\{0\}\cup L$ is used since there are no orders in the system at the start of the scheduling horizon.

$$OR_{o,l,t} = OR_{o,l,t-1} - \sum_{\substack{a\in A_{l+1}\\ l<|L|}}\sum_{t'\in TP} W_{o,l+1,a,t,t'}|_{t<t'} + \sum_{\substack{a\in A_l\\ l>0}}\sum_{t'\in TP} W_{o,l,a,t',t}|_{t'<t} + E_{o,t}|_{l=0} - D_{o,t}|_{l=|L|} \qquad \forall o\in O, l\in\{0\}\cup L, t\in TP \tag{38}$$

$E_{o,t}$ is a parameter with a value of 1 if order o is received at time t. This term is only used for the input order nodes ($l=0$). However, since incoming orders are difficult to model in continuous-time network models [22], $E_{o,t}$ is treated as a binary variable in this model. Its value is controlled by a logical proposition (39) and a disjunction (40), which are reformulated via Big-M into (41) and (42), respectively, where $M_3=\epsilon$, and $M_4=h$, $M_5=t_o^r$, $M_6=h+\epsilon-t_o^r$. An auxiliary binary $E'_{o,t}$ is introduced in the reformulation to indicate when $T_t<t_o^r$ or $t_o^r\le T_{t-1}$.

$$\{t_o^r\le 0\}\Leftrightarrow E_{o,1} \qquad \forall o\in O \tag{39}$$

$$\begin{bmatrix} E_{o,t} \\ T_{t-1}<t_o^r\le T_t\end{bmatrix} \vee \begin{bmatrix} \neg E_{o,t} \\ \begin{bmatrix} E'_{o,t} \\ T_t<t_o^r\end{bmatrix} \vee \begin{bmatrix} \neg E'_{o,t} \\ t_o^r\le T_{t-1}\end{bmatrix}\end{bmatrix} \qquad \forall o\in O, t\in\{2,\dots,|TP|\} \tag{40}$$

$$-M_3\cdot(1-E_{o,1})+\epsilon\le t_o^r\le M_4\cdot E_{o,1} \qquad \forall o\in O \tag{41}$$

$$\left.\begin{aligned} -M_5\cdot(E_{o,t}+E'_{o,t})+t_o^r\le T_{t-1}\le t_o^r-\epsilon+M_6\cdot(1-E_{o,t}) \\ -M_5\cdot(1-E_{o,t})+t_o^r\le T_t\le t_o^r-\epsilon+M_6\cdot(1-E'_{o,t}+E_{o,t})\end{aligned}\right\} \qquad \forall o\in O, t\in\{2,\dots,|TP|\} \tag{42}$$

$D_{o,t}$ is a variable that indicates if order o is delivered to the customer at time t, meaning it has been processed at the last stage of the network ($l=|L|$). (43) forces deliveries to occur immediately after the last stage has been processed, limiting symmetry in the solutions.

$$D_{o,t} = \sum_{a\in A_{|L|}}\sum_{t'\in TP} W_{o,|L|,a,t',t}|_{t'<t} \qquad \forall o\in O, t\in TP \tag{43}$$

Order Fulfillment Calculation: Each order o is considered fulfilled when it is delivered in the scheduling horizon, even if it is late (44).

$$F_o = \sum_{t\in TP} D_{o,t} \qquad \forall o\in O \tag{44}$$

Backlog Calculation: The same three scenarios used in the queue slot model to determine when an order is backlogged are used in this formulation. These are reformulated into (46). The auxiliary variable $B'_{o,t}$ and big-M values $M_7 = t_o^d + \epsilon$ and $M_8 = h - t_o^d$ are introduced to indicate if $T_t > t_o^d$.

$$\left.\begin{array}{rcl} \bigwedge\limits_{t\in TP} \neg D_{o,t} & \Rightarrow & B_o \\ \{T_t > t_o^d\} \;\wedge\; D_{o,t} & \Rightarrow & B_o \\ \{T_t \le t_o^d\} \;\wedge\; D_{o,t} & \Rightarrow & \neg B_o \end{array}\right\} \quad \forall o \in O \qquad (45)$$

$$\left.\begin{array}{c} -M_7 \cdot \left(1 - B'_{o,t}\right) + t_o^d + \epsilon \le T_t \le t_o^d + M_8 \cdot B'_{o,t} \\ 1 \le \sum\limits_{t\in TP} D_{o,t} + B_o \\ B_o \le B'_{o,t} + (1 - D_{o,t}) \\ B'_{o,t} + D_{o,t} - 1 \le B_o \end{array}\right\} \quad \forall o \in O \qquad (46)$$

Variable Domains: The domains of the variables are given in (47)-(54).

$$0 \le AR_{a,t} \le 1 \qquad \forall a \in A, t \in TP \qquad (47)$$

$$0 \le D_{o,t} \le 1 \qquad \forall o \in O, t \in TP \qquad (48)$$

$$0 \le F_o \le 1 \qquad \forall o \in O \qquad (49)$$

$$0 \le OR_{o,l,t} \le 1 \qquad \forall o \in O, l \in L, t \in TP \qquad (50)$$

$$0 \le T_t \le h \qquad \forall t \in TP \qquad (51)$$

$$B_o \in \{0,1\} \qquad \forall o \in O \qquad (52)$$

$$B'_{o,t}, E'_{o,t}, E_{o,t} \in \{0,1\} \qquad \forall o \in O, t \in TP \qquad (53)$$

$$W_{o,l,a,t,t'} \in \{0,1\} \qquad \forall o \in O, l \in L, a \in A_l, t, t' \in TP \qquad (54)$$

The continuous-time RTN model is given by (20), (36)-(38), (41)-(44), and (46)-(54).

3.4.2. Discrete-time Formulation

Agent Resource Balance: The agent resource balance in the discrete-time formulation is analogous to the one in the continuous-time RTN. It should be noted that a single time index is used in the binary variables $W_{o,l,a,t}$. Due to the discretization Δt, the processing times $t_{o,l,a}$ are rounded up to the nearest multiple of Δt: $\tau'_{o,l,a} = \lceil \tau_{o,l,a}/\Delta t \rceil$. This ensures that approximation errors are handled in a conservative fashion, which avoids infeasibilities due to the discretization of time. The release and due dates for each order are rounded up and down, respectively: $\overline{t_o^r} = \lceil t_o^r/\Delta t \rceil + 1$ and $\underline{t_o^d} = \lfloor t_o^d/\Delta t \rfloor + 1$. All time values are expressed in terms of timepoints, rather than absolute times and the $+1$ in each expression is used to account for the fact that the first timepoint ($t = 1$) corresponds to the beginning of the scheduling horizon ($T = 0$). The agent resources are initialized with $AR_{a,0} = 1 \;\; \forall a \in A$ since all agents are available at the start of the scheduling horizon.

$$AR_{a,t} = AR_{a,t-1} - \sum_{o \in O} \sum_{l \in L} W_{o,l,a,t} + \sum_{o \in O} \sum_{l \in L} W_{o,l,a,t-\tau'_{o,l,a}} \big|_{t \geq \tau'_{o,l,a}} \qquad \forall a \in A, t \in TP \tag{55}$$

Order Resource Balance: The order resource balance is also analogous to the one used in the continuous-time RTN. A key difference is that $E_{o,t}$ is treated as a parameter such that $E_{o,\overline{t_o^r}} = 1$. Order resources are initialized with $OR_{o,l,0} = 0 \;\; \forall o \in O, l \in \{0\} \cup L$ since there are no orders in the system at the start of the scheduling horizon. Deliveries are calculated as in the continuous-time RTN formulation.

$$OR_{o,l,t} = OR_{o,l,t-1} - \sum_{\substack{a \in A_{l+1} \\ l < |L|}} W_{o,l+1,a,t} + \sum_{\substack{a \in A_l \\ l > 0}} W_{o,l,a,t-\tau'_{o,l,a}} \big|_{t \geq \tau'_{o,l,a}} + E_{o,t} \big|_{l=0} - D_{o,t} \big|_{l=|L|} \qquad \forall o \in O, l \in \{0\} \cup L, t \in TP \tag{56}$$

$$D_{o,t} = \sum_{a \in A_{|L|}} W_{o,|L|,a,t-\tau'_{o,|L|,a}} \big|_{t \geq \tau'_{o,|L|,a}} \qquad \forall o \in O, t \in TP \tag{57}$$

Order Fulfillment Calculation: An order is fulfilled when it is delivered, even if it is late. It should be noted that since $W_{o,l,a,t}$ can be fixed to 0 when $t < \overline{t_o^r}$, the sum in (58) can begin at $\overline{t_o^r}$.

$$F_o = \sum_{t=\overline{t_o^r}}^{|TP|} D_{o,t} \qquad \forall o \in O \tag{58}$$

Backlog Calculation: Determining if an order is backlogged is very straightforward in the discrete-time RTN model since each on-time delivery must occur within the allowed window of timepoints, $\left[\overline{t_o^r}, \underline{t_o^d}\right]$.

$$B_o = 1 - \sum_{t=\overline{t_o^r}}^{\underline{t_o^d}} D_{o,t} \qquad \forall o \in O \tag{59}$$

Variable Domains: The domains of the variables are given in (60)-(64).

$$0 \leq AR_{a,t} \leq 1 \qquad \forall a \in A, t \in TP \tag{60}$$

$$0 \leq D_{o,t} \leq 1 \qquad \forall o \in O, t \in TP \tag{61}$$

$$0 \leq B_o, F_o \leq 1 \qquad \forall o \in O \tag{62}$$

$$0 \leq OR_{o,l,t} \leq 1 \qquad \forall o \in O, l \in L, t \in TP \tag{63}$$

$$W_{o,l,a,t} \in \{0,1\} \qquad \forall o \in O, l \in L, a \in A_l, t \in TP \tag{64}$$

The discrete-time RTN model is given by (20) and (55)-(64).

3.5. Discrete-time State-Task Network (STN)

Due to the structure of the system being modeled, the discrete-time STN has the same formulation as the discrete-time RTN model with one change: constraint (55) is replaced by (65). Since agent resources are not modeled in STN, (65) is used to ensure that each agent can process only one order at a time. The graphical representation of the STN for the OTC process is the same as that of the RTN (**Figure 5**), with the agent resources removed.

$$\sum_{o\in O}\sum_{l\in L}\sum_{t'=t-\tau_{o,l,a}+1}^{t} W_{o,l,a,t'} \leq 1 \qquad \forall a \in A, t \in TP \qquad (65)$$

The discrete-time STN model is given by (20) and (56)-(65).

4. Case Studies

All case studies were programmed in JuMP 0.21.1 (Julia 1.3.0) and solved using IBM CPLEX 12.9 as the MIP solver on an Ubuntu 18.04 server with 98 GB of RAM. All runs were restricted to single thread execution. The scheduling models compared in this case study are 1) general precedence, 2) queue slot, 3) continuous-time RTN with 10, 13, 15 and 20 timepoints, 4) discrete-time STN and RTN with 1.00, 0.50, 0.10, 0.05, 0.01 day resolutions. A 30% backlog penalty is imposed in all cases.

4.1. Case Study I: Small Instance

A small instance of the OTC process is used in which there are 3 stages with 4 agents (1 customer service representative [CSR], 2 inventory management agents, and 1 logistics agent) that process 10 orders during a 10 period horizon (**Figure 7**). The order release dates, due dates, prices, and backlog penalties are given in **Table 1**. The processing times for the agents are given in **Table 2**. All data were generated at random.

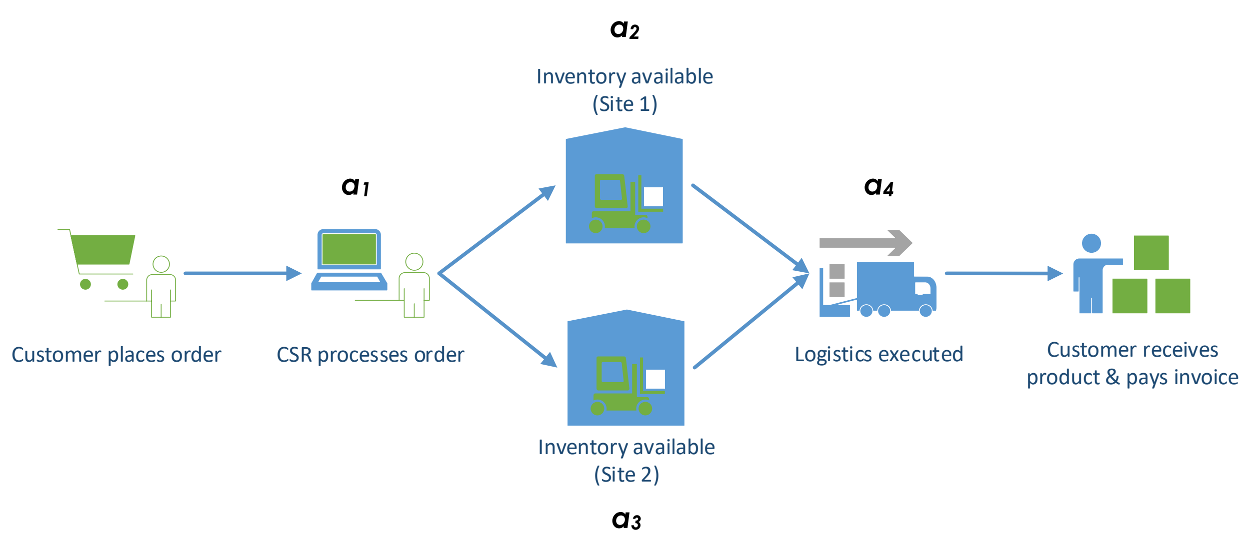

Figure 7. OTC process structure for case study I

Table 1. Order release dates, due dates, prices, and backlog penalties for case study I

Order #	Release Date *Day*	Release Date *Time*	Due Date *Day*	Due Date *Time*	On-time Revenue (thousands)	Backlog Penalty (thousands)
1	3	7:42 PM	7	6:58 AM	$511	$153
2	5	2:20 PM	9	8:42 PM	$803	$241
3	2	4:05 AM	5	8:36 AM	$671	$201
4	6	8:23 AM	9	5:34 PM	$86	$26
5	6	12:57 AM	9	8:35 PM	$362	$109
6	3	6:36 PM	9	1:45 PM	$388	$117
7	4	2:46 AM	9	8:04 PM	$401	$120
8	6	5:03 PM	9	12:44 PM	$507	$152
9	0	7:38 AM	9	8:52 PM	$550	$165
10	0	8:25 PM	6	4:39 AM	$67	$20

Table 2. Processing times for case study I

Stage: Agent	Processing Time (days)
1: CSR	0.91
2: Warehouse 1	0.79
3: Warehouse 2	0.78
4: Logistics	1.11

4.2. Case Study II: Intermediate Instance

The intermediate case is a 10-stage case (**Figure 8**), which is a typical number of activities in an industrial OTC process. A scheduling horizon of 30 periods is used, in which 20 orders are received. 3 agents from a set of 30 agents are assigned to each stage. Order and agent details are given in **Table 3** and **Table 4**, respectively.

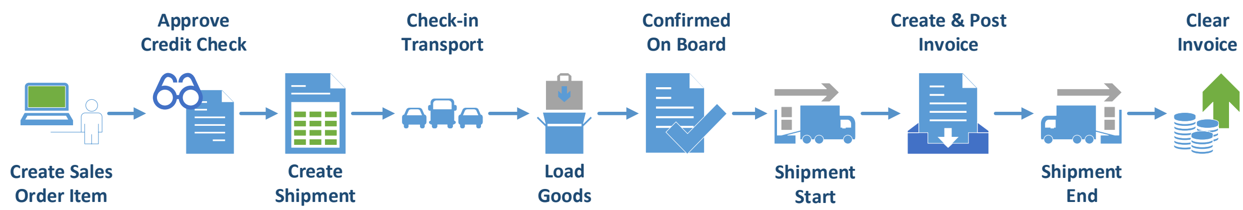

Figure 8. OTC process structure for case study II

Table 3. Order release dates, due dates, prices, and backlog penalties for case study II

Order #	Release Date *Day*	Release Date *Time*	Due Date *Day*	Due Date *Time*	On-time Revenue (thousands)	Backlog Penalty (thousands)
1	11	8:43 PM	25	1:32 AM	$624	$187
3	17	9:08 AM	27	10:34 PM	$981	$294
4	6	5:45 PM	19	4:55 AM	$356	$107

5	19	5:09 PM	27	5:16 PM	$963	$289
6	18	6:04 PM	28	1:03 PM	$105	$32
7	11	5:20 PM	29	2:19 PM	$205	$62
8	12	6:41 PM	26	3:33 AM	$615	$184
9	20	8:05 PM	29	8:46 PM	$659	$198
10	0	11:44 PM	18	2:04 PM	$980	$294
11	2	3:25 PM	10	10:35 PM	$941	$282
12	11	10:35 AM	23	3:10 AM	$530	$159
13	17	11:04 PM	26	7:59 PM	$341	$102
14	15	12:23 AM	22	6:56 PM	$438	$132
15	1	9:59 PM	22	6:08 AM	$977	$293
16	8	2:35 AM	24	8:58 AM	$645	$193
17	8	4:30 PM	24	1:30 PM	$160	$48
18	8	11:19 PM	20	7:29 PM	$295	$88
19	11	8:25 AM	25	12:58 PM	$727	$218
20	12	7:01 AM	25	12:53 PM	$297	$89

Table 4. Agent assignments and processing times for case study II

Stage #	Assigned Agents	Processing Times (days)
1	3, 10, 12	0.59, 1.05, 0.74
2	8, 18, 28	0.82, 0.84, 0.96
3	2, 5, 27	0.90, 0.89, 1.30
4	1, 20, 30	1.47, 1.03, 0.59
5	22, 24, 27	1.14, 0.63, 1.32
6	12, 16, 26	0.99, 1.42, 1.37
7	1, 14, 23	1.19, 1.3, 0.66
8	5, 12, 25	1.18, 0.56, 1.14
9	5, 8, 15	1.08, 1.06, 0.98
10	4, 16, 21	1.28, 1.28, 0.91

4.3. Case Study III: Larger Instance

Case study III has the same overall structure as case study II, but with an increased number of orders (50). See **Table 8** and **Table 9** in the Appendix for order and stage configuration details.

5. Results

5.1. Case Study I

All models were compared in the small case study instance and solved to optimality (0% gap). **Figure 9** shows the optimal profit found for each model and the computational time spent finding it. It can be observed, that only continuous-time models (general precedence and queue-slot) find the true optimum of $3.11 million. Compared to a first-in-first-out (FIFO) solution of $2.71 million, using optimization yields a 15% improvement. However, the continuous-time RTN model fails to reach the optimum, due to an insufficient number of timepoints. At 13 timepoints, the optimization times out after 1 hour of run time with a feasible schedule comparable to the FIFO solution. When the number of timepoints is increased to 15, the final optimality gap after 1 hour increases due to the effects of the increased model size on the computational performance.

For the discrete-time models, the best solution found improves as the discretization becomes finer. As the discretization provides only an approximate solution, the best solution found ($3.02 million) is 3% lower than the true optimum (approx. 10% higher than using a FIFO approach). The Gantt charts for the best continuous-time and the best discrete-time solutions (**Figure 10** and **Figure 11**) show that the discrete-time model fails to schedule order #10 because of the model precision. It should be noted that the general precedence model outperforms the queue-slot model in terms of speed, as does the discrete-time STN when compared to the discrete-time RTN. In both cases, the model size is the main source for these differences (see **Table 5**). It is also interesting that after presolve, the reduced models for the discrete-time formulations are 0-1 integer programs. The discrete-time models have the tightest LP relaxations of all the formulations. For this small instance, the general precedence model outperforms the discrete-time models in both solution quality and computational performance.

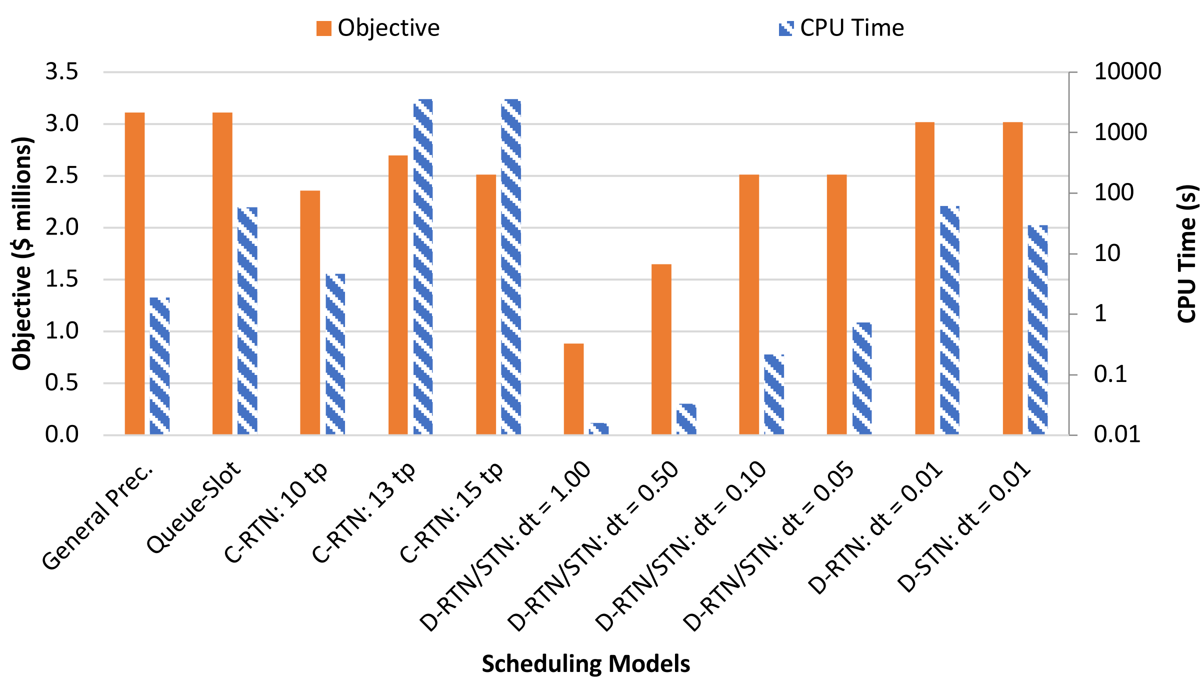

Figure 9. Objective function value (left vertical axis) and CPU time (right vertical axis) for each scheduling model run in case study I

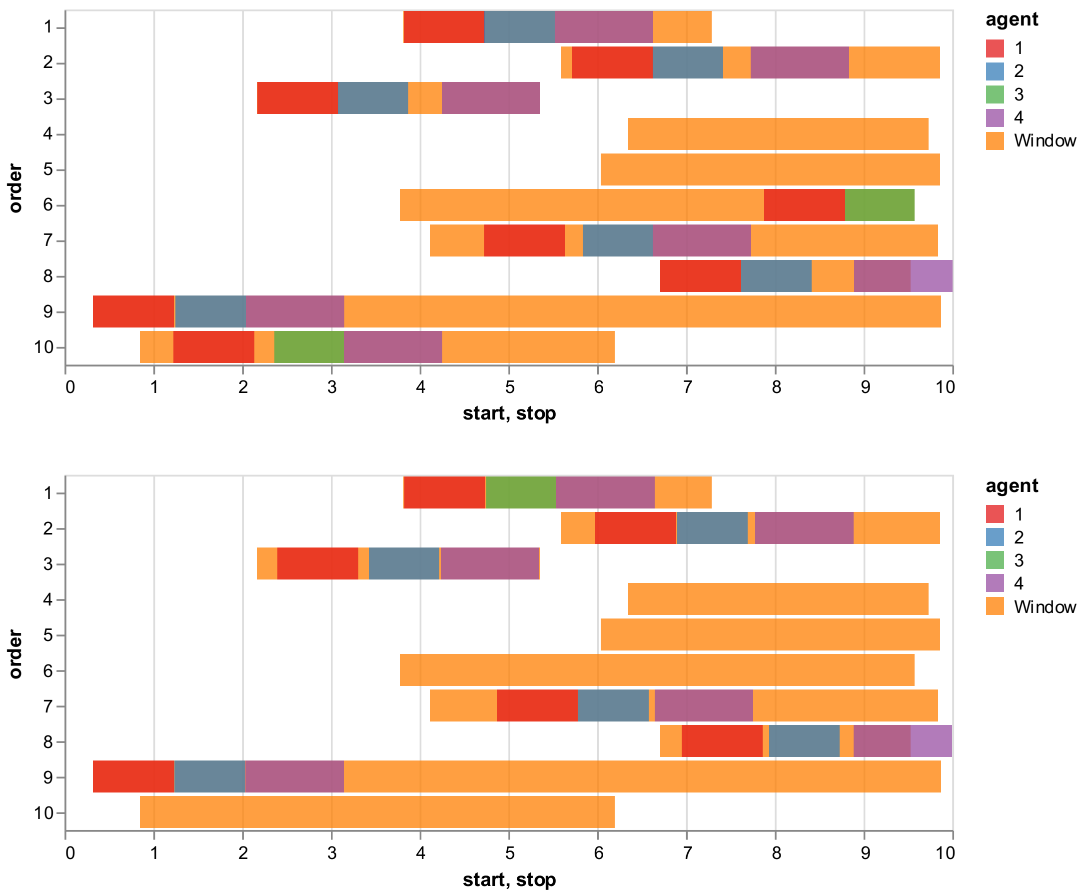

Figure 10. Gantt chart solutions for orders in case study I. Top: general precedence solution. Bottom: discrete network with 0.01 timesteps solution. Windows designate the window between the release and due date for each order.

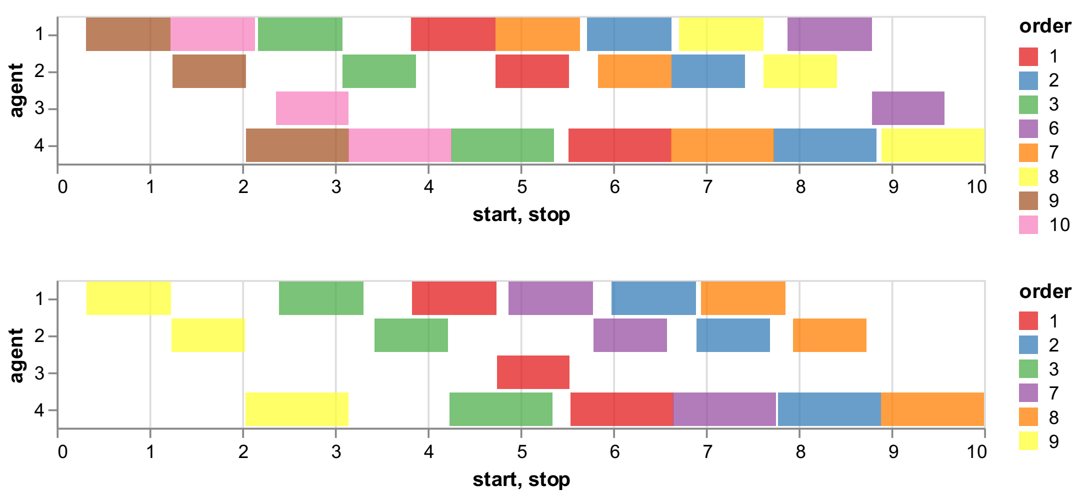

Figure 11. Gantt charts for each agent's schedule in case study I. Top: general precedence solution. Bottom: 0.01 timestep discrete network solution.

Table 5. Model sizes and performance metrics for case study I

	General Precedence	Continuous Queuing	Continuous RTN			Discrete RTN / STN				
			10 tp	*13 tp*	*15 tp*	*dt = 1.00*	*dt = 0.50*	*dt = 0.10*	*dt = 0.05*	*dt = 0.01*
CPU Time (s)	1.889	58.313	4.703	3600	3600	0.017 / 0.016	0.034 / 0.033	0.384 / 0.217	1.099 / 0.724	61.261 / 29.665
Relaxation*	4.348	4.326	3.164	4.104	4.28	0.884	1.647	2.513	2.513	3.02
Objective*	3.108	3.108	2.36	2.7	2.513	0.884	1.647	2.513	2.513	3.02
Nodes	1572	1415	2396	209674	99284	0	0	0	0	0
Gap	0%	0%	0%	25%	54%	0%	0%	0%	0%	0%
Original										
Constraints	2,140	7,082	12,022	19,576	25,632	4,037 / 3,949	7,617 / 7,449	36,247 / 35,451	72,038 / 70,454	358,358 / 350,474
Binary Vars.	1,040	1,220	5,700	9,750	13,050	1,320 / 1,320	2,520 / 2,520	12,120 / 12,120	24,120 / 24,120	120,120
Cont. Vars.	70	2,410	614	779	889	658 / 610	1,198 / 1,110	5,518 / 5,110	10,918 / 10,110	54,118 / 50,110
Reduced										
Constraints	840	1,732	1,031	1,511	1,855	139 / 139	278 / 275	1,279 / 1,268	2,542 / 2,522	12,622 / 12,528
Binary Vars.	330	420	1,664	2,966	4,038	312 / 305	688 / 599	3,317 / 2,899	6,642 / 5,817	33,182 / 29,096
Cont. Vars.	60	760	8	11	13	- / -	- / -	- / -	- / -	- / -

*$ millions

5.2. Case Study II

In the second case study, some of the formulations drop out as potential candidates for tractable models. The queue-slot model times out without having found a feasible solution. The same is true for the 20 timepoint continuous-time RTN. The 15 timepoint continuous-time RTN proves to yield a quite suboptimal result that is about one fourth of the optimal profit for this case study. The discrete-time network models prove to be faster than the general precedence model, with the STN model solving faster than the RTN model, but yield suboptimal results due to the time discretization (20% below the optimal profit). Increasing the level of discretization by 5x, however, results in larger models that time out after 1 hour with no solution.

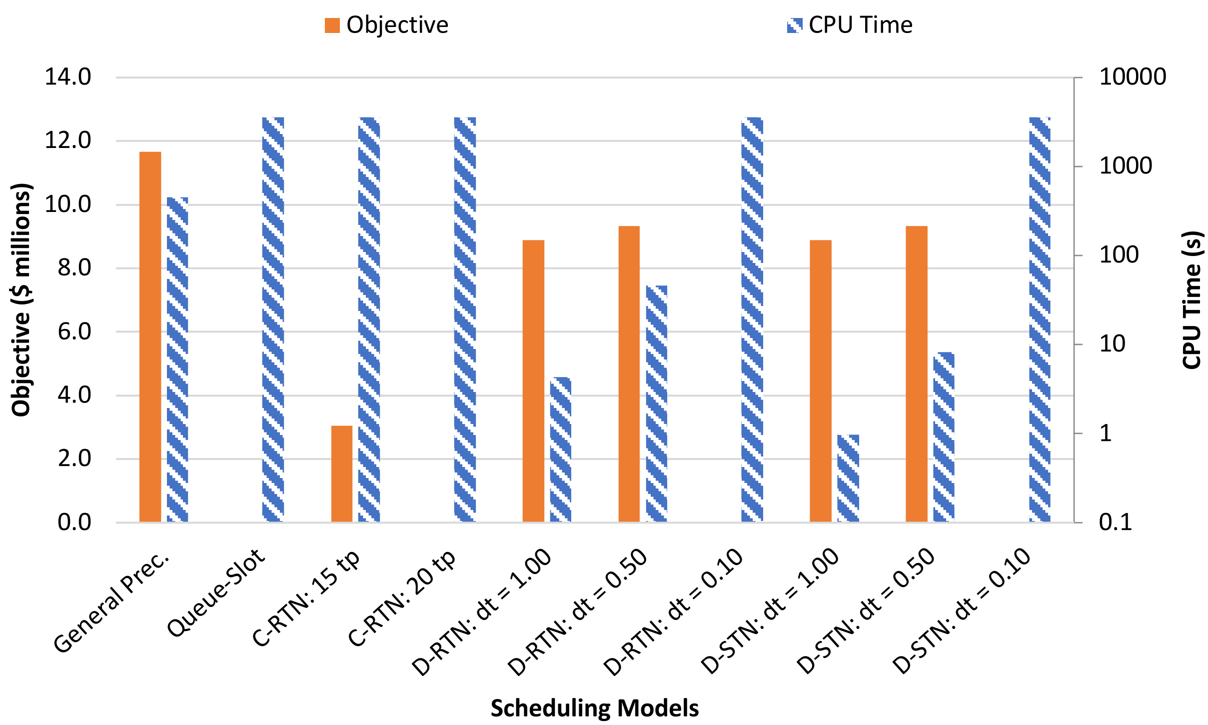

Figure 12. Objective function value (left vertical axis) and CPU time (right vertical axis) for each scheduling model run in case study II

Table 6. Model sizes and performance metrics for case study II

	General Precedence	Continuous Queuing	Continuous RTN		Discrete RTN / STN		
			15 tp	*20 tp*	*dt = 1.0*	*dt = 0.5*	*dt = 0.1*
CPU Time (s)	453.22	3600	3600	3600	4.31 / 0.97	46.31 / 8.30	3600
Objective*	11.663	-	3.053	-	8.893	9.323	-
Nodes	813	-	2163	-	0	0	-
Gap	0%	-	102%	-	0%	0%	-
Original							
Constraints	91,160	582,516	1,215,998	2,192,708	382,954 / 381,094	753,220 / 749,590	3,715,186 / 3,697,396
Binary Vars.	46,040	120,040	630,900	1,141,200	186,000 / 186,000	366,000 / 366,000	1,806,000 / 1,806,000
Cont. Vars.	420	240,020	4,355	5,710	8,660 / 7,700	16,760 / 14,900	81,560 / 72500
Reduced							
Constraints	38,860	-	5,051	-	3,148 / 3,145	6,284 / 6,270	- / -
Binary Vars.	12,060	-	29,485	-	11,875 / 11,531	23,963 / 22,706	- / -
Cont. Vars.	400	-	13	-	- / -	- / -	- / -

*$ millions

5.3. <u>Case Study III</u>

In the last case study, the best continuous-time model (general precedence) and the best discrete-time model (STN) were compared. The general precedence model timed out after 1 hour with no solution. The best solution found after 1 hour by the discrete-time STN was $9.62 million (2.6% optimality gap). It should be noted that before branching at the root node, the optimality gap for the STN model was 3.4% (after 128 seconds). Thus, discrete-time STN proves to be the most tractable model, finding near-optimal solutions quickly.

Table 7. Model sizes and performance metrics for case study III

	General Precedence	**Discrete STN dt = 1.0**
CPU Time (s)	3600	3600
Objective*	-	9.618
Nodes	-	10,958
Gap	-	2.6%
Original		
Constraints	655,300	949,107
Binary Vars.	265,100	465,000
Cont. Vars.	1,050	17,600
Reduced		
Constraints	-	5,546
Binary Vars.	-	22,871
Cont. Vars.	-	-

*$ millions

6. Conclusions

Five scheduling frameworks from the PSE community were adapted and applied to business process operations. This work serves as an initial study to incorporate techniques from the PSE community and the top-down supply chain approach into the transactional bottom-up supply chain modeling approach. Thus, mathematical models are shown to be valuable in obtaining optimal schedules for supply chain operations from a business process side. This work complements the data driven approaches utilized currently in BPM and provides optimality guarantees that are not available from the data driven approaches. Of the scheduling frameworks studied, the continuous-time general precedence model proves to be the most accurate, but presents issues with scaling. On the other hand, the discrete-time STN model has shown to be more tractable despite introducing suboptimal schedules due to approximation errors. However, these errors are handled in a conservative fashion to ensure that the resulting schedules are feasible. The superior computational performance of the discrete-time models is observed in the tightness of the models, which modern commercial solvers can exploit to find near-optimal solutions rapidly. Future directions of this approach to business process optimization include 1) using superstructure mathematical programming for optimizing business process designs, 2) addressing uncertainty and rework in the current approach, and 3) improving the computational performance and accuracy of the proposed methodology for large instances.

7. References

[1] G. Büyüközkan and F. Göçer, "Digital Supply Chain: Literature review and a proposed framework for future research," *Computers in Industry,* vol. 97, pp. 157-177, 2018.

[2] R. Cole, M. Stevenson and J. Aitken, "Blockchain Technology: Implications for operations and supply chain management," *Supply Chain Management,* vol. 24, no. 4, pp. 469-483, 2019.

[3] W. M. P. van der Aalst, A. H. M. ter Hofstede and M. Weske, "Business process management: a survey," *Business Process Management,* pp. 1-12, 2003.

[4] M. Dumas, M. L. Rosa, J. Mendling and H. A. Reijers, Fundamentals of Business Process Management, 2013.

[5] W. M. P. van der Aalst, "Business Process Management: A Comprehensive Survey," *International Scholarly Research Notices,* vol. 2013, pp. 1-37, 2013.

[6] P. Samaranayake, "Business process integration, automation, and optimization in ERP: Integrated approach using enhanced process models," *Business Process Management Journal,* vol. 15, no. 4, pp. 504-526, 2009.

[7] F. Niedermann, S. Radeschütz and B. Mitschang, "Business Process Optimization Using Formalized Optimization Patterns," in *International Conference on Business Information Systems*, 2011.

[8] K. Vergidis, D. Saxena and A. Tiwari, "An evolutionary multi-objective framework for business process optimisation," *Applied Soft Computing,* vol. 12, no. 8, pp. 2638-2653, 2012.

[9] P. Afflerbach, M. Hohendorf and J. Manderscheid, "Design it like Darwin - A value-based application of evolutionary algorithms for proper and unambiguous business process redesign," *Information Systems Frontiers,* vol. 19, no. 5, pp. 1101-1121, 2017.

[10] N. Mahammed and S. M. Benslimane, "An Evolutionary Algorithm Based Approach for Business Process Multi-Criteria Optimization," *International Journal of Organizational and Collective Intelligence,* vol. 7, no. 2, pp. 34-53, 2017.

[11] C. Gröger, H. Schwarz and B. Mitschang, "Prescriptive Analytics for Recommendation-Based Business Process Optimization," in *International Conference on Business Information Systems*, 2014.

[12] W. M. P. van der Aalst, M. Bichler and A. Heinzl, "Robotic Process Automation," *web intelligence,* vol. 60, no. 4, pp. 269-272, 2018.

[13] J. Krumeich, D. Werth and P. Loos, "Prescriptive Control of Business Processes," *Business & Information Systems Engineering,* vol. 58, no. 4, pp. 261-280, 2016.

[14] J. F. Shapiro, "Bottom-Up Vs. Top-Down Approaches to Supply Chain Modeling," *Quantitative Models For Supply Chain Management,* vol. 2, pp. 737-759, 1999.

[15] I. E. Grossmann, "Advances in mathematical programming models for enterprise-wide optimization," *Computers & Chemical Engineering,* vol. 47, pp. 2-18, 2012.

[16] I. Harjunkoski, C. T. Maravelias, P. Bongers, P. M. Castro, S. Engell, I. E. Grossmann, J. N. Hooker, C. A. Méndez, G. Sand and J. M. Wassick, "Scope for industrial applications of production scheduling models and solution methods," *Computers & Chemical Engineering,* vol. 62, pp. 161-193, 2014.

[17] N. Shah, "Process industry supply chains: Advances and challenges," *Computers & Chemical Engineering,* vol. 29, no. 6, pp. 1225-1235, 2005.

[18] B. Mota, M. I. Gomes, A. Carvalho and A. P. Barbosa-Povoa, "Towards supply chain sustainability: economic, environmental and social design and planning," *Journal of Cleaner Production,* vol. 105, pp. 14-27, 2015.

[19] V. A. Varma, G. V. Reklaitis, G. E. Blau and J. F. Pekny, "Enterprise-wide modeling & optimization—An overview of emerging research challenges and opportunities," *Computers & Chemical Engineering,* vol. 31, pp. 692-711, 2007.

[20] J. M. Laínez and L. Puigjaner, "Prospective and perspective review in integrated supply chain modelling for the chemical process industry," *Current Opinion in Chemical Engineering,* vol. 1, no. 4, pp. 430-445, 2012.

[21] C. A. Floudas and X. Lin, "Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review," *Computers & Chemical Engineering,* vol. 28, no. 11, pp. 2109-2129, 2004.

[22] C. A. Méndez, J. Cerdá, I. E. Grossmann, I. Harjunkoski and M. Fahl, "State-of-the-art review of optimization methods for short-term scheduling of batch processes," *Computers & Chemical Engineering,* vol. 30, no. 6, pp. 913-946, 2006.

[23] B. Brunaud, H. D. Perez, S. Amaran, S. J. Bury, J. M. Wassick and I. E. Grossmann, "Batch scheduling with quality-based changeovers," *Computers & Chemical Engineering,* vol. 132, p. 106617, 2020.

[24] J. M. Pinto and I. E. Grossmann, "A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants," *Industrial & Engineering Chemistry Research,* vol. 34, no. 9, pp. 3037-3051, 1995.

[25] C. Méndez, G. Henning and J. Cerdá, "An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities," *Computers & Chemical Engineering,* vol. 25, no. 4, pp. 701-711, 2001.

[26] C. T. Maravelias and I. E. Grossmann, "On the relation of continuous- and discrete-time state–task network formulations," *AIChE Journal,* vol. 52, no. 2, pp. 843-849, 2006.

[27] P. M. Castro, A. P. Barbosa-Póvoa and H. A. Matos, "Optimal periodic scheduling of batch plants using RTN-based discrete and continuous-time formulations: A case study approach," *Industrial & Engineering Chemistry Research,* vol. 42, no. 14, pp. 3346-3360, 2003.

[28] J. Ostrowski, M. F. Anjos and A. Vannelli, "Tight Mixed Integer Linear Programming Formulations for the Unit Commitment Problem," *IEEE Transactions on Power Systems,* vol. 27, no. 1, pp. 39-46, 2012.

[29] P. M. Castro, A. P. Barbosa-Póvoa, H. A. Matos and A. Q. Novais, "Simple Continuous-Time Formulation for Short-Term Scheduling of Batch and Continuous Processes," *Industrial & Engineering Chemistry Research,* vol. 43, no. 1, pp. 105-118, 2004.

8. Appendix: Data for Case Study III

Table 8. Order release dates, due dates, prices, and backlog penalties for case study III

Order #	Release Date Day	Release Date Time	Due Date Day	Due Date Time	On-time Revenue (thousands)	Backlog Penalty (thousands)
1	11	3:40 PM	24	3:49 PM	$332	$100
2	17	1:44 AM	25	2:52 PM	$59	$18
3	6	2:53 PM	17	2:57 PM	$908	$272
4	19	8:45 AM	27	7:43 PM	$107	$32
5	18	10:05 AM	28	9:12 PM	$517	$155
6	11	12:21 PM	24	10:42 AM	$29	$9
7	12	1:14 PM	21	6:54 AM	$608	$182
8	20	11:13 AM	28	2:08 PM	$690	$207
9	0	11:19 PM	19	3:48 PM	$275	$82
10	2	2:18 PM	13	10:15 PM	$215	$64
11	11	5:43 AM	21	6:51 AM	$898	$269
12	17	3:25 PM	26	9:41 AM	$740	$222
13	14	6:00 PM	28	7:44 PM	$791	$237
14	1	9:10 PM	15	12:09 AM	$506	$152
15	7	11:08 PM	25	8:13 PM	$348	$104
16	8	12:48 PM	19	4:17 AM	$935	$280
17	8	7:30 PM	22	6:07 AM	$982	$295
18	11	3:35 AM	23	2:52 AM	$472	$142
19	12	1:47 AM	25	5:33 PM	$485	$146
20	1	11:28 AM	29	9:55 PM	$100	$30
21	13	4:59 PM	23	2:25 PM	$203	$61
22	21	1:19 PM	29	7:43 PM	$828	$248
23	7	7:35 PM	20	2:53 AM	$38	$11
24	21	3:43 AM	29	9:18 AM	$203	$61
25	2	7:32 AM	21	5:02 PM	$422	$127
26	4	12:10 PM	24	5:16 AM	$561	$168
27	13	12:11 PM	26	12:52 PM	$737	$221
28	14	11:17 AM	29	1:38 AM	$219	$66
29	21	12:56 PM	29	11:06 PM	$357	$107
30	20	4:05 PM	29	8:30 PM	$721	$216
31	11	3:33 PM	27	5:45 PM	$574	$172
32	7	11:34 AM	18	4:45 PM	$390	$117
33	9	3:13 PM	20	2:40 AM	$312	$94
34	21	11:22 AM	29	6:59 PM	$741	$222
35	14	3:57 AM	27	6:42 PM	$826	$248
36	3	12:06 PM	27	10:51 PM	$994	$298
37	6	11:29 AM	20	5:06 PM	$490	$147
38	15	11:12 PM	26	3:07 AM	$148	$44
39	6	12:40 PM	24	10:08 PM	$693	$208

40	18	2:58 AM	28	10:33 AM	$140	$42
41	7	10:38 AM	24	6:44 AM	$357	$107
42	3	8:58 PM	25	2:39 PM	$790	$237
43	17	2:58 AM	26	10:24 PM	$218	$66
44	10	12:44 AM	27	10:39 PM	$858	$258
45	13	3:32 PM	26	12:02 PM	$152	$45
46	15	2:49 PM	24	5:22 PM	$617	$185
47	16	4:18 PM	26	12:37 PM	$678	$203
48	11	9:30 PM	25	8:29 AM	$654	$196
49	7	1:57 PM	29	2:55 PM	$13	$4
50	10	7:40 AM	20	8:33 AM	$748	$225

Table 9. Agent assignments and processing times for case study III

Stage #	Assigned Agents	Processing Times (days)
1	12, 27, 28	0.82, 0.95, 0.78
2	19, 22, 27	0.92, 1.3, 0.5
3	1, 11, 12	0.73, 0.87, 1.07
4	7, 9, 23	1.47, 0.57, 0.87
5	3, 13, 28	1.18, 1.33, 0.87
6	4, 15, 16	0.81, 0.87, 1.11
7	2, 12, 25	0.83, 0.91, 0.84
8	6, 11, 28	0.96, 0.94, 1.4
9	15, 16, 21	1.46, 1.21, 1.26
10	4, 22, 23	1.01, 1.18, 0.79