Mathematical Programming Representation of Discrete-Event Simulation

ARTICLE HISTORY
Compiled August 9, 2020

ABSTRACT

1. Introduction

Discrete Event Simulation (DES) is one of the most used tool for performance evaluation of discrete event dynamic systems, and hence, simulation optimization algorithms are widely developed for optimizing the parameters of such systems. Most of the state-of-the-art simulation optimization algorithms consider DES as a black-box function, and the structure of DES has been seldom studied. Under the black-box setting, simulation optimization algorithms always require huge amount of iterations to explore the search region of the optimization problem, and can be computationally inefficient. On the contrary, a minority of the simulation optimization literature explores the structure of the DES, and that research is referred to as white-box simulation optimization. The benefit of white-box simulation optimization is saving of simulation budget since optimization procedure is guided by the information contained in the structure of DES. However, the barrier of white-box simulation optimization is how to model DES as white box, so that it eventually favors the optimization. In this work, we present how to establish an equivalent Mathematical Programming Representation (MPR) for certain types of DES. We deploy this work with the modeling approach, conditions under which the approach can be applied and some examples.

Chan and Schruben (2008) is the first work proposing a modeling framework to translate DES into MPR in a general sense. Their modeling framework is based on the Event Relationship Graph (ERG) of the system dynamics. To derive the MP model, an ERG of the discrete event system has to be constructed and expanded to an elementary ERG (EERG) model, and a routined approach can be applied to translate the EERG model into an MP model. However, the work of Chan and Schruben (2008) has some limitations. First, deriving an ERG is not an easy task, and the user has to pay quite much attention to detect all the event relationships and complete the triggering conditions between each pair of related events. That limits the wide spread of such approach. Second, Chan and Schruben (2008) proposes the modeling approach case-by-case, which means the user has to first identify which situations he faces by analyzing the EERG and then choose the appropriate model. This is quite a burden, since EERG is an expansion of ERG and could be huge. This work proposes a user-friendly modeling approach without plotting the ERG and the modeling procedure can be automatically generated in a general-purpose programming language. Despite taking different paths, the MPRs proposed in this work and Chan and Schruben (2008) lead to the equivalent results, which are both equivalent to a simulation realization.

When there is already a DES model, the benefit of developing an MPR is not obvious due to the high complexity of solving MPR. However, the MPR of a simulation model favors the optimal design and control of the DES thanks to the vast theory and methodological works in MP. Many works in literature show the potentiality of this research direction. For instance, the gradient can be conveniently estimated from the simulation model, if the MPR is approximated into Linear Programming (LP) and solving the dual of the LP (Chan & Schruben, 2008; Zhang, Matta, & Alfieri, 2020). Moreover, if some of the parameters in the MPR are changed to decision variables, the MPR becomes an integrated simulation optimization model. Solving the integrated model provides the optimal solution of the optimization problem (Matta, 2008). MP-based algorithms, such as linear programming approximation (Alfieri & Matta, 2012), Benders decomposition (Weiss & Stolletz, 2015), column generation (Alfieri, Matta, & Pastore, 2020), has been applied to improve the efficiency of solving the integrated MP models.

The application of MPR-based simulation optimization approaches is usually found in operations management of manufacturing and service systems. The integrated simulation optimization model has proved itself to be well suited in solving the buffer allocation problem. The most efficient approach to finding the sample-path global optimal of the buffer allocation problem of serial production line is developed based on it (Zhang, Pastore, Matta, & Alfieri, n.d.). Thanks to the flexibility of DES in evaluating complex systems, the buffer allocation problem of production systems with complex blocking mechanism, such as kanban control, base stock control, extended kanban control, can be managed (Pedrielli, Alfieri, & Matta, 2015). Thanks to the flexibility of MPR in modeling optimization problems, problems involving real-valued decision variables such as optimal production rate (Tan, 2015), bottleneck detection (Zhang & Matta, 2018) and throughput improvement problem (Zhang & Matta, 2020) have all been well addressed and the sample-path global optimal solution can be obtained. It is worthy to notice that, before the above mentioned works were proposed, there were many state-of-the-art heuristic approaches addressing those problems, but without guarantee of global or local optimality. Thus, the development of MPR-base simulation optimization has made much contribution in the research area of manufacturing and service system optimization.

The rest of the paper is organized as follows.

2. Event-scheduling worldview of DES

The event-scheduling approach is the logic behind all major simulation software and used by people when programming their own model with general purpose languages Law, Kelton, and Kelton (2000). The logic is briefly shown in Figure 1. The fundamental elements of the algorithm is system state and events. The system state is a collection of state variables to describe the system at a particular time. An event execution leads to the change of system state, the events of the same type change the system state with the same function. The event list contains the scheduled but not executed events together with occurring times. When simulation is launched, the system state and the event list is initialized with user-defined values, and the simulation clock is set to zero. The event with the earliest occurring time in the event list will be picked to execute, and the system evolves into a new state together with the simulation clock. New state may enable to schedule new events, i.e., adding new event executions to the event list. Event cancellation may also occur, i.e., removing some event executions from the event list. There is usually a delay between the scheduling time and the execution time of the event, but the delay is equal to zero when necessary. The algorithm will terminate with certain conditions.

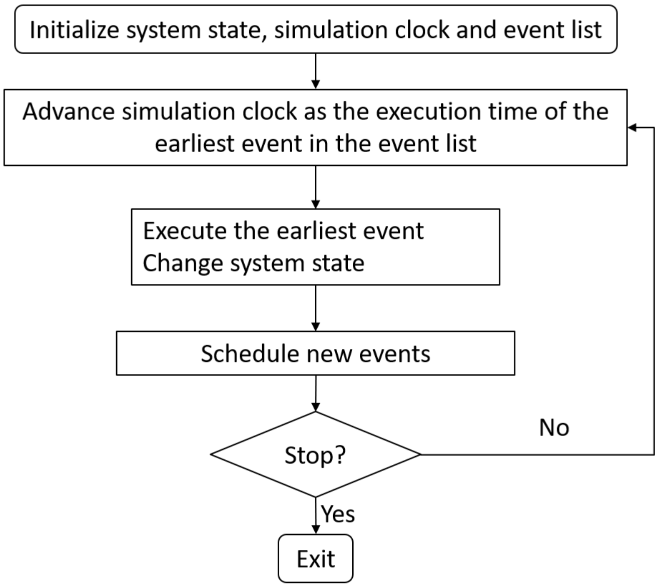

Figure 1. Event-scheduling simulation algorithm.

3. Mathematical programming representation of DES

In this section, we introduce a routined approach to translate DES into MPR. The proposed approach is applicable when the DES satisfies several conditions, and we first describe those conditions. The formulation approach is presented afterwards.

3.1. Conditions

Section 2 describes the event-scheduling algorithm for DES. To apply the approach proposed in this work, the following conditions must be satisfied.

(1) State variables are integer.
(2) For all event e^ξ of type ξ, the *scheduling condition* are in the form of $a_s^\xi \leq s \leq b_s^\xi$ for certain state variable s, and combined with logic operator "AND" when multiple state variables are involved, where $a^{\xi,s}$ and $b^{\xi,s}$ are given.
(3) The delay between scheduling and execution time of an event e^ξ are independent and identically distributed random variables.
(4) When more than one events in the event list have the same execution time as the earliest execution time, the system state, the event list and the number of past executions of each event type are independent of the sequence of execution when the simulation clock is advanced.

(5) Simulation terminates when the number of execution of each event e^ξ reaches N^ξ, which is known before launching the simulation.

Integer state variables, such as buffer level, idle servers, widely exist in practical simulation models. Moreover, integer variables include binary variables, which is a powerful tool in modeling the natural behavior and control of the system. The second condition is the most strict condition to satisfy. However, if one model does not satisfy this condition, more binary variables could be introduced to make it satisfied. As for the third condition, if the delay is not I.I.D. random variate, it should be splitted into several events so that each event has I.I.D delay. For instance, if the distribution service time depends on the job type, the event of *finish* a job should be splitted so that each job type is represented by one event. The forth condition saying that the execution time is the only attribute of priority for the event executions in the event list. One can specify the priority in scheduling stage, instead of execution stage. The last condition specifies the termination condition of the DES.

3.2. Mathematical programming model

The MPR represents the dynamics of the simulated system. Specifically, event scheduling time, event execution time and state variable changes along the simulation time line can all be seen in the MPR as decision variables. The i-th scheduling time and the i-th execution time of event e^ξ is denoted by $e_i^{\xi,0}$ and $e_i^{\xi,1}$ respectively. $\mathcal{E}_k$ denotes the time of the k-th execution of any event type, and the simulation clock is initialized as zero with $\mathcal{E}_k = 0$. $e_i^{\xi,0}$, $e_i^{\xi,1}$ and $\mathcal{E}_k$ are all real-valued. We use u_k^s to denote the value of state variable s just after the k-th execution, i.e., just after $\mathcal{E}_k$. The domain of variable u_k^s should be integer. The initial system state is given as u_0^s. Some binary variables are also used in the MPR, but they are introduced when it appears in some constraints.

To implement the event-schedule algorithm, the condition to schedule it, the distribution of the delay between scheduling and execution, the state variable changes upon execution and the canceled events of each event type should be provided. In the event list, multiple executions of the same event type are allowed to coexist. For instance, in a G/G/m queue system, when all the servers are occupied, there are m executions of departure event in the event list. The number of coexisting executions α^ξ of the each event type e^ξ is also mandatory to develop the MPR.

The first group of constraints, denoted by group A, is the constraints binding $e_i^{\xi,1}$ and $\mathcal{E}_k$. Binary variables $w_{i,k}^\xi$ are used, and $e_i^{\xi,1}$ and $\mathcal{E}_k$ are binded if $w_{i,k}^\xi$ is equal to one, as shown in constraints (A1) and (A2). Constraints (A3) and (A4) states that each $e_i^{\xi,1}$ can be binded to one and only one $\mathcal{E}_k$. Constraints (A5) imply that the index executions of the same event type is sequenced along with the index of $\mathcal{E}_k$.

$$
\begin{aligned}
e_i^{\xi,1} - \mathcal{E}_k \geq M(w_{i,k}^{\xi} - 1) \quad & \forall\ \xi, i, k \quad (A1)\\
\mathcal{E}_k - e_i^{\xi,1} \geq M(w_{i,k}^{\xi} - 1) \quad & \forall\ \xi, i, k \quad (A2)\\
\sum_k w_{i,k}^{\xi} = 1 \quad & \forall\ \xi, i \quad (A3)\\
\sum_{\xi,i} w_{i,k}^{\xi} = 1 \quad & \forall\ k \quad (A4)\\
\sum_k k w_{i+1,k}^{\xi} - \sum_k k w_{i,k}^{\xi} \geq 1 \quad & \forall\ \xi, i \quad (A5)
\end{aligned}
$$

The second group of constraints, denoted by group B, is the constraints binding $e_i^{\xi,0}$ and $e_i^{\xi,1}$. If the event is a single-execution event, i.e., the maximal number of executions coexisting in the event list is equal to one, the i-th execution $e_i^{\xi,1}$ is binded to the i-th scheduling with a delay t_i^{ξ}, as in constraints (B1).

$$
e_i^{\xi,1} - e_i^{\xi,0} = t_i^{\xi} \quad \forall \xi, i = 1, ..., N^{\xi} \quad (B1) \tag{1}
$$

If event e^{ξ} is a multiple-execution event, i.e., the maximal number of executions coexisting in the event list is equal to α^{ξ}, which is at least two, the a late scheduled execution could overtake an early scheduled one, since the delay time between scheduling and execution is random variate. Thus, we introduce the binary variable $y_{i,i'}^{\xi}$, and the i-th scheduling of event e^{ξ} is the i'-th execution when $y_{i,i'}^{\xi}$ is equal to one, as in constraints (B2) and (B3). Constraints (B4) and (B5) show that each $e_{i'}^{\xi,1}$ can be binded to one and only one $e_i^{\xi,0}$. Constraints (B6) imply that the i-th scheduling cannot be binded to an execution earlier than the $(i + \alpha^{\xi})$-th, since at most α^{ξ} executions are allowed to coexist in the list. For the same reason, constraints (B7) state that i'-th execution cannot be binded to the scheduling later than the $(i' - \alpha^{\xi})$-th.

$$
\begin{aligned}
e_{i'}^{\xi,1} - e_i^{\xi,0} \geq t_i^{\xi} + M(y_{i,i'}^{\xi} - 1) \qquad & \forall \xi, i, i' = 1, ..., N^{\xi} \qquad (B2)\\
e_i^{\xi,0} - e_{i'}^{\xi,1} \geq -t_i^{\xi} + M(y_{i,i'}^{\xi} - 1) \qquad & \forall \xi, i, i' = 1, ..., N^{\xi} \qquad (B3)\\
\sum_{i=1}^{N^{\xi}} y_{i,i'}^{\xi} = 1 \qquad & \forall\ \xi, i' = 1, ..., N^{\xi} \qquad (B4)\\
\sum_{i'=1}^{N^{\xi}} y_{i,i'}^{\xi} = 1 \qquad & \forall\ \xi, i = 1, ..., N^{\xi} \qquad (B5)\\
\sum_{i'=i+\alpha^{\xi}}^{N^{\xi}} y_{i,i'}^{\xi} = 0 \quad & \forall\ \xi,\ i = 1, ..., N^{\xi} - \alpha^{\xi} \quad (B6)\\
\sum_{i=1}^{i'-\alpha^{\xi}} y_{i,i'}^{\xi} = 0 \quad & \forall\ \xi,\ i' = \alpha^{\xi} + 1, ..., N^{\xi} \quad (B7)
\end{aligned}
$$

The third group of constraints, denoted by grtroup C, state that event e^{ξ} can be scheduled right after $\mathcal{E}_k$ if the condition for scheduling an event e^ξ is true. We first introduce binary variables $x^\xi_{i,k}$, and $x^\xi_{i,k}$ equal to one represents that the i-th scheduling of event e^ξ is enabled by $\mathcal{E}_k$, as in constraints (C1) and (C2). Constraints (C3) show that each scheduling is enabled by one and only one $\mathcal{E}_k$. As stated in Section 3.1, the condition for scheduling event e^ξ is a set of inequalities in the form of $a^\xi_s \leq s \leq b^\xi_s$ combined with operator "AND". Binary variables z^ξ_k are used to verify that condition, and if z^ξ_k is equal to one, the condition is true, as in constraints (C4) and (C5). Moreover, a set of binary variables $v^{\xi,s,a}_k$ and $v^{\xi,s,b}_k$ are used to verify if the condition is false. Specifically, constraints (C6) state that if $v^{\xi,s,a}_k$ is equal to one, u^s_k will be smaller than $a^{\xi,s}$, and hence, the inequality $a^\xi_s \leq s$ is violated. Similar for constraints (C7), if $v^{\xi,s,b}_k$ is equal to one, $s \leq b^\xi_s$ is violated. Constraints (C8) imply that z^ξ_k is equal to one if and only if $a^\xi_s \leq s \leq b^\xi_s$, and that z^ξ_k is equal to zero if and only if $a^\xi_s - 1 \geq s$ or $s \geq b^\xi_s + 1$. Constraints (C9) state that if the condition to schedule e^ξ is true, an execution of e^ξ has to be added into the event list. Constraints (C10) show that the index of scheduling is sequenced along with the index of event execution $\mathcal{E}_k$. Constraints (C11) imply that the $(i+\alpha^\xi, k)$-th scheduling of event e^ξ must occur after the i-th execution.

For events with multiple executions, several rules should be follow. First, multiple execution events should all have positive delay. In fact, when the delay between scheduling and execution of an event is zero, multiple executions is equivalent to sequential scheduling a single-execution event. For the event e^ξ with strictly positive delay and multiple executions, the following routine should be followed to formulate a correct MPR. The number of executions in the event list should be treated as state variable, and an event $e^{\tilde{\xi}}$ that updates the number of executions should be included artificially, and this event should have the same scheduling condition as e^ξ and zero delay. However, the reason is not to limit the number of executions, but to allow scheduling of multiple executions, instead. According to constraints (C9), at most one execution can be scheduled after execution of $\mathcal{E}_k$. To schedule multiple executions, we use the artificial event to make sure the number of executions of event e^ξ can reach α^ξ if the condition to schedule is satisfied. Constraints (C12) states that the i-th scheduling of event e^{ξ} and $e^{\tilde{\xi}}$ should be enabled by the same execution.

$$
\begin{aligned}
e_i^{\xi,0} - \mathcal{E}_k &\geq M(x_{i,k}^{\xi} - 1) \quad \forall\, \xi, k, i \quad (C1)\\
\mathcal{E}_k - e_i^{\xi,0} &\geq M(x_{i,k}^{\xi} - 1) \quad \forall\, \xi, k, i \quad (C2)\\
\sum_k x_{i,k}^{\xi} &= 1 \quad \forall\, \xi, i \quad (C3)\\
u_k^s - a^{\xi,s} &\geq M(z_k^{\xi} - 1) \quad \forall\, \xi, k, s \quad (C4)\\
b^{\xi,s} - u_k^s &\geq M(z_k^{\xi} - 1) \quad \forall\, \xi, k, s \quad (C5)\\
(a^{\xi,s} - 1) - u_k^s &\geq M(v_k^{\xi,s,0} - 1) \quad \forall\, \xi, k, s \quad (C6)\\
u_k^s - (b^{\xi,s} + 1) &\geq M(v_k^{\xi,s,1} - 1) \quad \forall\, \xi, k, s \quad (C7)\\
1 - z_k^{\xi} &\leq \sum_{s \in S^{\xi}} v_k^{\xi,s,0} + \sum_{s \in S^{\xi}} v_k^{\xi,s,1} \quad \forall\, \xi, k \quad (C8)\\
\sum_{i=1}^{N^{\xi}} x_{i,k}^{\xi} &= z_k^{\xi} \quad \forall\, \xi, k \quad (C9)\\
\sum_k k x_{i+1,k}^{\xi} - \sum_k k x_{i,k}^{\xi} &\geq 1 \quad \forall\, \xi, i \quad (C10)\\
\sum_k k x_{i+\alpha^{\xi},k}^{\xi} - \sum_k k w_{i,k}^{\xi} &\geq 0 \quad \forall\, \xi, i \quad (C11)\\
\sum_k k(x_{i,k}^{\xi} - x_{i,k}^{\tilde{\xi}}) &= 0 \quad \forall\, \xi, i \quad (C12)
\end{aligned}
$$

The constraints (D1) represent the evolution of state variables. If the $\mathcal{E}_k$ is of event type ξ, the state variable s is changed by function $f^{\xi}(u_{k-1}^s)$.

$$
u_k^s = \sum_{\xi} \sum_{i=1}^{N^{\xi}} w_{i,k}^{\xi} f^{\xi}(u_{k-1}^s) \quad \forall\, s, k \quad (D1)
$$

If all the events changes the state variables with a fixed increment or decrement, constraints (D1) will be changed to (D2), which is linear constraints, and the MPR is an MILP.

$$
u_k^s = u_{k-1}^s + \sum_{\xi} \sum_{i=1}^{N^{\xi}} w_{i,k}^{\xi} \Delta^{\xi,s} \quad \forall\, s, k \quad (D2)
$$

Constraints (E1) show that the $\mathcal{E}_k$ are temporally sequenced with index k.

$$
\mathcal{E}_k - \mathcal{E}_{k-1} \geq 0 \quad \forall\, k \quad (E1)
$$

With the constraints above, there is a unique feasible solution in terms of event occurring times (solution of the binary variables could be multiple in case of multiple simultaneous events). Thus, the objective can be arbitrarily.

4. Examples

In this section, we present several examples. For each example, we introduce the necessary information to be provided, and then the derivation of MPR is trivial since the approach proposed in Section 3 is well routined.

4.1. Single server merge

The first example we model is a queueing system composed of three servers within a merge architecture. Jobs arrive at server 1 or server 2, and the buffer in front of the two servers have infinite capacity. We assume that all jobs arrive at time zero. After processing a job, server 1 or 2 can release the job to the finite buffer in front of server 3, if the buffer is not full. Thus, the blocking policy is block-after-service. If there is only one space available in the buffer, and both server 1 and 2 are holding a finished job, server 1 has higher priority to release. After processing a job, server 3 releases the job from the system immediately.

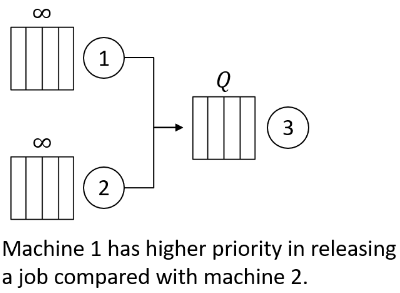

Figure 2. Example: single server merge.

We first define the state variables. Variable m^i and q represents the state of server i and the available space in buffer 3 respectively. For i equal to 1 or 2, the server have three states, namely *idle*, *working* and *holding a finished job*, and each state refers to m^i equal to 0, 1 and 2 respectively. Server 3 has only two states, *idle* and *working*, and refers to m^3 equal to 0 and 1, respectively. The servers are initially idle, and buffer 3 is empty, i.e., $q = Q$ where Q denotes the capacity of buffer 3.

We then define the events composing the DES as shown in Table 1. Event $e^{s,1}$ represents the starting of service of server 1, scheduling of such event is enabled if server 1 is idle, i.e., $m^1 \leq 0$. Executing $e^{s,1}$ will increase state variable m^1 from 0 to 1. Event $e^{f,1}$ represents the finishing of server 1, scheduling of it requires that server 1 is working. The delay between scheduling and execution is random variate t^1, which is the service time. Executing $e^{f,1}$ makes the server hold a finished job and hence increase state variable m^1 from 1 to 2. Events $e^{s,2}$ and $e^{f,2}$ are similar with events $e^{s,1}$ and $e^{f,1}$. The departure event of server 1 and server 2 differ from each other. As long as there is a space available in buffer 3 and server 1 holds a finished job, an departure event of server 1 $e^{d,1}$ can be scheduled. For the departure event of server 2 $e^{d,2}$, one more condition to verify is that server 1 does not hold a finished job due to the priority rule. The execution of departure event will occupy an available space in buffer 3, and make the server idle. The start event of server 3 $e^{s,2}$ is enabled if it is idle and there is at least one job waiting in buffer 3. After execution of $e^{s,3}$, the server becomes busy and the available space in buffer 3 is added by one. The departure event $e^{d,3}$ is can be scheduled if server 3 is working with delay t^3, which is a random variate representing the service time of server 3. Execution of $e^{d,3}$ makes server 3 to be idle. Since all the stations are composed of a single server, the number of coexisting executions of all events is equal to one.

Variable	Event	Condition to schedule	Delay	α^ξ	State change
$e^{s,1}$	Start m1	$m^1 \leq 0$	0	1	$m^1 ++$
$e^{f,1}$	Finish m1	$1 \leq m^1 \leq 1$	t^1	1	$m^1 ++$
$e^{d,1}$	Depart m1	$m^1 \geq 2$ & $q \geq 1$	0	1	$m^1 = m^1 - 2,\ q --$
$e^{s,2}$	Start m2	$m^2 \leq 0$	0	1	$m^2 ++$
$e^{f,2}$	Finish m2	$1 \leq m^2 \leq 1$	t^2	1	$m^2 ++$
$e^{d,2}$	Depart m2	$m^2 \geq 2$ & $q \geq 1$ $m^1 \leq 1$	0	1	$m^2 = m^2 - 2,\ q --$
$e^{s,3}$	Start m3	$m^3 \leq 0$ & $q \leq Q-1$	0	1	$m^3 ++,\ q ++$
$e^{d,3}$	Depart m3	$m^3 \geq 1$	t^3	1	$m^3 --$

Table 1. Events to simulate a merge queueing system.

4.2. Multi-server merge

We now study a multi-server merge queueing system, as shown in Figure 3. The multi-server merge queue is a generalization of the system presented in Section 4.1, where the number of parallel servers in station 1, 2 and 3 is equal to s^1, s^2 and s^3 respectively. The state variables are changed accordingly. To describe the state of multi-server station j, now we need two state variables g^j and h^j to represent the number of idle servers and number of finished jobs of each the station. The state variable q is used to present the number of available space in buffer 3.

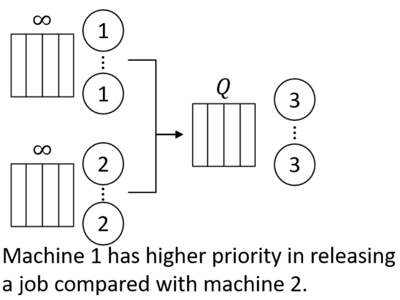

Figure 3. Example: multi-server merge.

The events composing the DES model are shown in Table 2. The start event of station 1 and 2 is scheduled when there are at least one empty server, and their execution decreases the number of empty servers by one. The scheduling condition of finish event $e^{f,j}$ is the same as $e^{s,j}$, but its execution will increase the number of finished jobs by one. Event $e^{f,j}$ is a multi-execution event with positive delays, an event to count the number of executions of it should be introduced. However, the event $e^{s,j}$ plays that role and the number of executions in the event list is equal to $(s^j - g^j)$. Similarly with single-server system, the departure of station 1 requires that there is at least one finished job in the station and there is at least one space available in buffer 3, but the departure of station 2 also requires that there is no finished job in station

1. The departure of station 1 and 2 will increase the number of empty servers by one, decrease the number of finished jobs by one and decrease the number of available space by one. As for station 3, the start and departure event can be scheduled if there is at least one empty server and one job in buffer 3. Thus, $e^{s,3}$ is used to count the number executions in the event list of $e^{d,3}$. Execution of $e^{s,3}$ will decrease the number of empty server by one and increase the available space in buffer 3 by one.

Variable	Event	Condition to schedule	Delay	α^{ξ}	State change
$e^{s,1}$	Start 1	$1 \leq g^1$	0	1	$g^1 --$
$e^{f,1}$	Finish 1	$1 \leq g^1$	t^1	s^1	$h^1 ++$
$e^{d,1}$	Depart 1	$1 \leq h^1$& $q \geq 1$	0	1	$g^1 ++, h^1 --,\ q --$
$e^{s,2}$	Start 2	$1 \leq g^2$	0	1	$g^2 --$
$e^{f,2}$	Finish 2	$1 \leq g^2$	t^2	s^2	$h^2 ++$
$e^{d,2}$	Depart 2	$1 \leq h^2$ & $q \geq 1$ & $h^1 \leq 0$	0	1	$g^2 ++, h^2 --,\ q --$
$e^{s,3}$	Start 3	$1 \leq g^3$ & $q \leq Q-1$	0	1	$g^3 --,\ q ++$
$e^{d,3}$	Depart 3	$1 \leq g^3$ & $q \leq Q-1$	t^3	s^3	$g^3 ++$

Table 2. Events to simulate a multi-server merge queueing system.

MP model of single server merge model:

$$\min \sum_k \mathcal{E}_k \tag{2}$$

$$e_i^{(\xi,j),1} - \mathcal{E}_k \geq M(w_{i,k}^{\xi,j} - 1) \qquad \xi \in \{s,f,d\}, j \in \{1,2,3\}, \forall i,k \tag{3}$$

$$\mathcal{E}_k - e_i^{(\xi,j),1} \geq M(w_{i,k}^{\xi,j} - 1) \qquad \xi \in \{s,f,d\}, j \in \{1,2,3\}, \forall i,k \tag{4}$$

$$\sum_k w_{i,k}^{\xi,j} = 1 \qquad \forall\ \xi \in \{s,f,d\}, j \in \{1,2,3\}, i \tag{5}$$

$$\sum_{(\xi,j),i} w_{i,k}^{\xi,j} = 1 \qquad \forall\ k \tag{6}$$

$$\sum_k k w_{i+1,k}^{\xi,j} - \sum_k k w_{i,k}^{\xi,j} \geq 1 \qquad \forall\ \xi \in \{s,f,d\}, j \in \{1,2,3\}, i \tag{7}$$

$$e_i^{s,j,1} - e_i^{s,j,0} \geq 0 \qquad j = 1,2,3, \forall\ i \tag{8}$$

$$e_i^{f,j,1} - e_i^{f,j,0} \geq t_i^j \qquad j = 1,2, \forall\ i \tag{9}$$

$$e_i^{d,j,1} - e_i^{d,j,0} \geq 0 \qquad j = 1,2, \forall\ i \tag{10}$$

$$e_i^{d,3,1} - e_i^{d,3,0} \geq t_i^3 \qquad \forall\ i \tag{11}$$

$$e_i^{\xi,j,0} - \mathcal{E}_k \geq M(x_{i,k}^{\xi,j} - 1) \qquad \forall\ \xi \in \{s,f,d\}, j = 1,2,3, k, i \tag{12}$$

$$\mathcal{E}_k - e_i^{\xi,j,0} \geq M(x_{i,k}^{\xi,j} - 1) \qquad \forall\ \xi \in \{s,f,d\}, j = 1,2,3, k, i \tag{13}$$

$$m_k^j = m_{k-1}^j + \sum_{i=1}^{N^j} (w_{i,k}^{s,j} + w_{i,k}^{f,j} - 2w_{i,k}^{d,j}) \qquad j = 1,2, \forall\ k \tag{14}$$

$$m_k^3 = m_{k-1}^3 + \sum_{i=1}^{N^3} (w_{i,k}^{s,3} - w_{i,k}^{d,3}) \qquad \forall\ k \tag{15}$$

$$q_k = q_{k-1} + \sum_{i=1}^{N^3} w_{i,k}^{s,3} - \sum_{i=1}^{N^1} w_{i,k}^{d,1} - \sum_{i=1}^{N^2} w_{i,k}^{d,2} \tag{16}$$

$$m_k^j \geq M(z_k^{s,j} - 1) \qquad j = 1,2,3, \forall\ k \tag{17}$$

$$1 - m_k^j \geq M(z_k^{f,j} - 1) \qquad j = 1,2, \forall\ k \tag{18}$$

$$m_k^j - 1 \geq M(z_k^{f,j} - 1) \qquad j = 1,2, \forall\ k \tag{19}$$

$$m_k^j - 2 \geq M(z_k^{d,j} - 1) \qquad j = 1,2, \forall\ k \tag{20}$$

$$q_k - 1 \geq M(z_k^{d,j} - 1) \qquad j = 1,2, \forall\ k \tag{21}$$

$$1 - m_k^1 \geq M(z_k^{d,2} - 1) \qquad \forall\ k \tag{22}$$

$$m_k^3 - 1 \geq M(z_k^{d,3} - 1) \qquad \forall\ k \tag{23}$$

$$(Q - 1) - q_k \geq M(z_k^{s,3} - 1) \qquad \forall\ k \tag{24}$$

$$\sum_k x_{i,k}^{\xi,j} = 1 \qquad \forall\ \xi \in \{s,f,d\}, j = 1,2,3, \forall\ i,k \tag{25}$$

$$\sum_{i=1}^{N^j} x_{i,k}^{\xi,j} \leq z_k^{\xi,j} \qquad \forall\ \xi \in \{s,f,d\}, j = 1,2,3, k \tag{26}$$

$$\sum_k k x_{i+1,k}^{\xi,j} - \sum_k k x_{i,k}^{\xi,j} \geq 0 \qquad \forall\ \xi \in \{s,f,d\}, j = 1,2,3, i \tag{27}$$

5. Resource allocation problem of queueing systems

5.1. Mathematical programming representation of simulation model

We study only the system that the occurrence of an event will lead to the increment or decrement of one unit of the state variables. A simulation model is called a *natural* simulator if the following assumptions all hold:

(1) An event e^{ξ} can be triggered if the state variables $\mathbf{s}$ satisfy specific conditions *at that time*, regardless of the history of the state or event occurrence, and the condition is not changed along time, i.e., condition is static. It could be possible to define more state variables in case of history dependence and variant triggering conditions.
(2) Natural triggering relationship: if and only if e^{ξ} is an s-increment event, e^{ξ} triggers an s-decrement event, vice versa.
(3) Natural triggering condition: the condition for triggering an e^{ξ} is that each state variable s must within its predefined domain, i.e., $\mathbf{l} \leq \mathbf{s} \leq \mathbf{u}$, regardless of event type ξ.
(4) For all e^{ξ}, the number of execution N^{ξ} is known before simulation, and the simulation terminate when all types of events have been triggered for that number.

Assumptions for a variable x to be resource-type:

(1) For all e^{ξ}, $\mathbf{u}$ is monotonically increasing on x, and $\mathbf{l}$ is monotonically decreasing on x.

Formulate the MP model of simulation:

$e_i^{\xi} \geq 0$	time of the i-th occurrence of event e^{ξ}.
$\tau_l^{s+} \geq 0$	time of the l-th occurrence of events that increments state variable s.
$\tau_l^{s-} \geq 0$	time of the l-th occurrence of events that decrements state variable s.
$x_{i,l}^{\xi,s+} \in \{0,1\}$	equal to 1 if e_i^{ξ} is the l-th increment of s.
$x_{i,l}^{\xi,s-} \in \{0,1\}$	equal to 1 if e_i^{ξ} is the l-th decrement of s.

The MP model of simulation is

$$\min\{\textstyle\sum_{\xi,i} e_i^{\xi}\} \tag{28}$$

$$s.t. \tag{29}$$

$$\tau_l^{s+} - \tau_{l+s_0-u_s}^{s-} \geq 0 \qquad \forall\ s \tag{30}$$

$$\tau_l^{s-} - \tau_{l-s_0+l_s}^{s+} \geq 0 \qquad \forall\ s \tag{31}$$

$$\tau_l^{s+} - e_i^{\xi} \geq M(x_{i,l}^{\xi,s+} - 1) \quad \forall\ s\ and\ e^{\xi}\ with\ increment\ of\ s \tag{32}$$

$$\tau_l^{s-} - e_i^{\xi} \geq M(x_{i,l}^{\xi,s-} - 1) \quad \forall\ s\ and\ e^{\xi}\ with\ decrement\ of\ s \tag{33}$$

$$e_i^{\xi} - \tau_l^{s+} \geq M(x_{i,l}^{\xi,s+} - 1) \quad \forall\ s\ and\ e^{\xi}\ with\ increment\ of\ s \tag{34}$$

$$e_i^{\xi} - \tau_l^{s-} \geq M(x_{i,l}^{\xi,s-} - 1) \quad \forall\ s\ and\ e^{\xi}\ with\ decrement\ of\ s \tag{35}$$

$$\textstyle\sum_{\xi,i} x_{i,l}^{\xi,s+} = 1 \qquad \forall\ s, l \tag{36}$$

$$\textstyle\sum_{\xi,i} x_{i,l}^{\xi,s-} = 1 \qquad \forall\ s, l \tag{37}$$

$$\textstyle\sum_{s,l} x_{i,l}^{\xi,s+} = 1 \qquad \forall\ \xi, i \tag{38}$$

$$\textstyle\sum_{s,l} x_{i,l}^{\xi,s-} = 1 \qquad \forall\ \xi, i \tag{39}$$

$$\tag{40}$$

5.2. Mathematical programming representation of simulation model - V2

Revise event-based simulation algorithm.

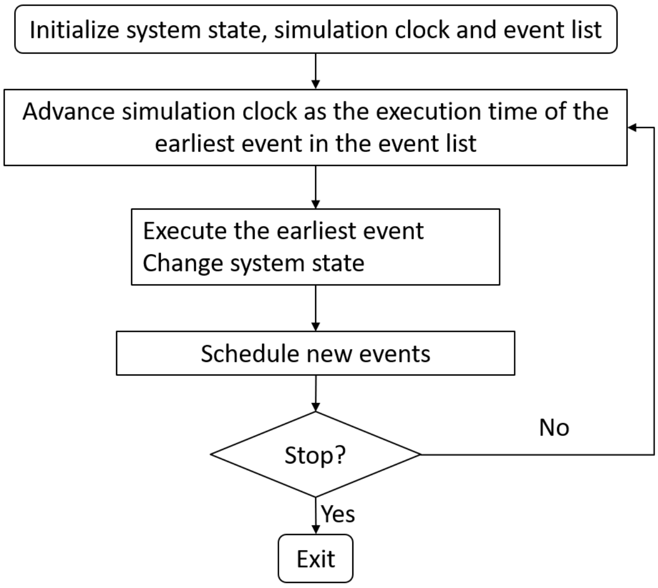

Figure 4. Event-based simulation algorithm.

An equivalent mathematical programming model exists if the following assumptions are satisfied:

(1) State variables are integer.
(2) For all event e^ξ, the *scheduling conditions* are in the form of $a_s^\xi \leq s \leq b_s^\xi$ combined with logic operator "AND", where s is a state variable, and a_s^ξ and b_s^ξ are lower and upper bounds.
(3) The scheduling conditions is independent of the history and not changed along time. (It could be possible to define more state variables in case of history dependence and time-variant scheduling conditions.)
(4) An event execution of e^ξ leads to integer increment or decrement equal to Δ_s^ξ of certain state variables s, and Δ_s^ξ is not changed along time.
(5) The delay between scheduling and execution time of an event e^ξ, denoted by t^ξ, is random variate. They can be generated independently from the simulation run. (*This point is different from ERG. In ERG, the delay is dependent on the edge, i.e, a couple of events, but I consider delay dependent on a single event.*)
(6) For all events e^ξ, the number of executions I^ξ is known before simulation.

Preparation Event e^ξ is expanded into a series of events $e^{\xi,0}$, $e^{\xi,1}$, ..., e^{ξ,Δ^ξ}, where Δ^ξ is equal to the maximum among Δ_s^ξ for all $s \in \Theta^\xi$. The expansion is conducted as follows. First, event $e^{\xi,0}$ is executed as soon as all the scheduling conditions are satisfied, and the state variables $s \in \Theta^\xi$ are not changed. Then, event $e^{\xi,1}$ is executed after t^ξ time unit after an execution of $e^{\xi,0}$. For all $s \in \Theta^\xi$, if $\Delta_s^\xi \geq \delta$, $e^{\xi,\delta}$ will increase or decrease s by one, for all $\delta = 1, ..., \Delta^\xi$. The i-th execution of event $e^{\xi,\delta}$ for $\delta = 1, ..., \Delta^\xi$ are simultaneous.

Constraints (A) The constraints below imply that event $e^{\xi,1}$ is scheduled to exe-

e^{ξ}	event of type ξ
s	state variable
S	set of all state variables
S^{ξ}	set of state variables whose value is conditioned for scheduling event e^{ξ}.
$\Theta^{\xi+}$	the set of state variables that event e^{ξ} will increase its value.
$\Theta^{\xi-}$	the set of state variables that event e^{ξ} will decrease its value.
Θ^{ξ}	$\Theta^{\xi+} \cap \Theta^{\xi-}$
E^{s+}	set of events whose execution increases the value of state variable s.
E^{s-}	set of events whose execution decreases the value of state variable s.
Δ_s^{ξ}	increment or decrement of state variable s when event e^{ξ} is executed.
I^{ξ}	total number of executions of event e^{ξ}
L^{s+}	total number of times that state variable s is increased.
L^{s-}	total number of times that state variable s is decreased.
t^{ξ}	delay between scheduling and execution of event e^{ξ}.
t_i^{ξ}	delay between i-th scheduling and its execution of event e^{ξ}.

Table 3. Notations

$e_i^{\xi,\delta} \geq 0$	time of i-th execution of event $e^{\xi,\delta}$
$\tau_l^{s+} \geq 0$	time when state variable s is increased for the l-th time.
$\tau_l^{s-} \geq 0$	time when state variable s is decreased for the l-th time.
$x_{i,i'}^{\xi} \in \{0,1\}$	equal to 1 if the i' execution of event e^{ξ} is the i-th scheduled one.
$y_{i,l}^{\xi,\delta,s+} \in \{0,1\}$	equal to 1 if the i-th execution of event e^{ξ} is the l-th time that state variable s in incre
$y_{i,l}^{\xi,\delta,s-} \in \{0,1\}$	equal to 1 if the i-th execution of event e^{ξ} is the l-th time that state variable s in decre
$z_{i,l}^{\xi,s+} \in \{0,1\}$	equal to 1 if the i-th scheduling of event e^{ξ} is later than τ_l^{s+}.

Table 4. Decision variables

cute with a delay t^{ξ}, after an execution of $e^{\xi,0}$.

$$e_{i'}^{\xi,1} - e_i^{\xi,0} \geq t_i^{\xi} + M(x_{i,i'}^{\xi} - 1) \qquad \forall \xi, i, i' = 1, ..., I^{\xi} \tag{41}$$

It should be noticed that, if multiple executions of the same event e^{ξ} are allow to exist in the future event list simultaneously, the execution of $e^{\xi,1}$ scheduled by the i-th execution of $e^{\xi,0}$ may be not the i-th execution of $e^{\xi,1}$. Thus, binary variables $x_{i,i'}^{\xi}$ are introduced, and it is equal to one if the i' execution of event $e^{\xi,1}$ is scheduled by the i-th execution of event $e^{\xi,0}$. Since each execution of $e^{\xi,0}$ can schedule one and only one execution of $e^{\xi,1}$, the following constraints should also be satisfied:

$$\sum_{i=1}^{N^{\xi}} x_{i,i'}^{\xi} = 1 \qquad \forall\ \xi, i' = 1, ..., I^{\xi} \tag{42}$$

$$\sum_{i'=1}^{N^{\xi}} x_{i,i'}^{\xi} = 1 \qquad \forall\ \xi, i = 1, ..., I^{\xi} \tag{43}$$

If up to α^{ξ} multiple executions of event e^{ξ} are allowed, the following constraints can be added:

$$e_i^{\xi,0} - e_{i-\alpha^\xi}^{\xi,1} \geq 0 \qquad \forall\ \xi,\ i = \alpha^\xi + 1, ..., I^\xi \tag{44}$$

$$\sum_{i'=i+\alpha^\xi}^{I^\xi} x_{i,i'}^\xi = 0 \qquad \forall\ \xi,\ i = 1, ..., I^\xi - \alpha^\xi \tag{45}$$

$$\sum_{i=1}^{i'-\alpha^\xi} x_{i,i'}^\xi = 0 \qquad \forall\ \xi,\ i' = \alpha^\xi + 1, ..., I^\xi \tag{46}$$

If α^ξ is equal to one, the constraints (A) are reduced to:

$$e_i^{\xi,1} - e_i^{\xi,0} \geq t_i^\xi \qquad \forall \xi, i = 1, ..., I^\xi \tag{47}$$

$$e_i^{\xi,0} - e_{i-1}^{\xi,1} \geq 0 \qquad \forall\ \xi,\ i = 2, ..., I^\xi \tag{48}$$

Constraints (B) Binding $e_i^{\xi,\delta}$ and $\tau_l^{s+},\ \tau_l^{s-}$:

$$\tau_l^{s+} - e_i^{\xi,\delta} \geq M(y_{i,l}^{\xi,\delta,s+} - 1) \qquad \forall\ s \in S, e^\xi \in E^{s+}, \delta = 1, .., \Delta_s^\xi, i = 1, .., I^\xi, l = 1, .., L^{s+} \tag{49}$$

$$e_i^{\xi,\delta} - \tau_l^{s+} \geq M(y_{i,l}^{\xi,\delta,s+} - 1) \qquad \forall\ s \in S, e^\xi \in E^{s+}, \delta = 1, .., \Delta_s^\xi, i = 1, .., I^\xi, l = 1, .., L^{s+} \tag{50}$$

$$\tau_l^{s-} - e_i^{\xi,\delta} \geq M(y_{i,l}^{\xi,\delta,s-} - 1) \qquad \forall\ s \in S, e^\xi \in E^{s-}, \delta = 1, .., \Delta_s^\xi, i = 1, .., I^\xi, l = 1, .., L^{s-} \tag{51}$$

$$e_i^{\xi,\delta} - \tau_l^{s-} \geq M(y_{i,l}^{\xi,\delta,s-} - 1) \qquad \forall\ s \in S, e^\xi \in E^{s-}, \delta = 1, .., \Delta_s^\xi, i = 1, .., I^\xi, l = 1, .., L^{s-} \tag{52}$$

$$\sum_{\substack{\xi: e^\xi \in E^{s+} \\ i=1,...,I^\xi \\ \Delta=1,...,\Delta_s^\xi}} y_{i,l}^{\xi,\delta,s+} = 1 \qquad \forall\ s \in S, l = 1, ..., L^{s+} \tag{53}$$

$$\sum_{l=1,...,L^{s+}} y_{i,l}^{\xi,\delta,s+} = 1 \qquad \forall\ \xi, s \in \Theta^{\xi+}, i = 1, ..., I^\xi, \delta = 1, ..., \Delta_s^\xi \tag{54}$$

$$\sum_{\substack{\xi: e^\xi \in E^{s-} \\ i=1,...,I^\xi \\ \Delta=1,...,\Delta_s^\xi}} y_{i,l}^{\xi,\delta,s-} = 1 \qquad \forall\ s \in S, l = 1, ..., L^{s-} \tag{55}$$

$$\sum_{l=1,...,L^{s-}} y_{i,l}^{\xi,\delta,s-} = 1 \qquad \forall\ \xi, s \in \Theta^{\xi-}, i = 1, ..., I^\xi, \delta = 1, ..., \Delta_s^\xi \tag{56}$$

$$\tag{57}$$

Binary variables $y_{i,l}^{\xi,\delta,s+} \in \{0, 1\}$ are equal to one if the i-th execution of event $e^{\xi,\delta}$ is the l-th time that state variable s in increased. Since events $e^{\xi,1}, ..., e^{\xi,\Delta_s^\xi}$ are expanded from one event e^ξ, and they are executed simultaneously, the following constraints are also added:

$$e_i^{\xi,\delta} = e_i^{\xi,1} \qquad \forall\ \xi,\ i = 1, ..., I^\xi, \delta = 1, ..., \Delta^\xi \tag{58}$$

$$y_{i,l+\delta-1}^{\xi,\delta,s+} = y_{i,l}^{\xi,1,s+} \qquad \forall\ \xi,\ s \in \Theta^{\xi+},\ \delta = 1, ..., \Delta_s^\xi,\ i = 1, ..., I^\xi,\ l = 1, ..., L^{s+} - \Delta_s^\xi + 1 \tag{59}$$

$$y_{i,l+\delta-1}^{\xi,\delta,s-} = y_{i,l}^{\xi,1,s-} \qquad \forall\ \xi,\ s \in \Theta^{\xi-},\ \delta = 1, ..., \Delta_s^\xi,\ i = 1, ..., I^\xi,\ l = 1, ..., L^{s-} - \Delta_s^\xi + 1 \tag{60}$$

Constraints (C) To trigger event $e^{\xi,0}$, the conditions $a_s^\xi \le s \le b_s^\xi$ for all state variable $s \in S^\xi$ should be satisfied. $s \in S^\xi$ can be categorized into one of the following three situations:

- event e^ξ does not change the value of s, i.e., $s \notin \Theta^\xi$.
- event e^ξ increases the value of s, i.e., $s \in \Theta^{\xi+}$.
- event e^ξ decreases the value of s, i.e., $s \in \Theta^{\xi-}$.

If event e^ξ does not change the value of s, or if it is executed after being scheduled with positive delay, i.e., $s \notin \Theta^\xi$ or $t^\xi > 0$, the following constraints are applied:

$$e_i^{\xi,0} - \tau_l^{s+} \le M z_{i,l}^{\xi,s+} \quad \forall\, \xi, i = 1, ..., I^\xi,\; s \in S^\xi, l = 1, ..., L^{s+} \tag{61}$$
$$e_i^{\xi,0} - \tau_l^{s-} \le M z_{i,l}^{\xi,s-} \quad \forall\, \xi,\; i = 1, ..., I^\xi,\; s \in S^\xi, l = 1, ..., L^{s-} \tag{62}$$
$$e_i^{\xi,0} - \tau_l^{s-} \ge -M \hat{z}_{i,l}^{\xi,s-} \quad \forall\, \xi,\; i = 1, ..., I^\xi,\; s \in S^\xi, l = 1, ..., L^{s-} \tag{63}$$
$$e_i^{\xi,0} - \tau_l^{s+} \ge -M \hat{z}_{i,l}^{\xi,s+} \quad \forall\, \xi,\; i = 1, ..., I^\xi,\; s \in S^\xi, l = 1, ..., L^{s+} \tag{64}$$
$$z_{i,l}^{\xi,s+} + \hat{z}_{i,s_0+l-b_s^\xi}^{\xi,s-} \le 1 \quad ?? \tag{65}$$
$$z_{i,l}^{\xi,s-} + \hat{z}_{i,-s_0+l+a_s^\xi}^{\xi,s+} \le 1 \quad ?? \tag{66}$$
$$\tag{67}$$

$$e_i^{\xi,0} - \tau_l^{s+} \ge M(z_{i,l}^{\xi,s+} - 1) \quad \forall\, \xi, i = 1, ..., I^\xi,\; s \in S^\xi, l = 1, ..., L^{s+} \tag{68}$$
$$\tau_l^{s+} - e_i^{\xi,0} > -M z_{i,l}^{\xi,s+} \quad \forall\, \xi,\; i = 1, ..., I^\xi,\; s \in S^\xi, l = 1, ..., L^{s+} \tag{69}$$
$$e_i^{\xi,0} - \tau_l^{s-} \ge M(z_{i,l}^{\xi,s-} - 1) \quad \forall\, \xi,\; i = 1, ..., I^\xi,\; s \in S^\xi, l = 1, ..., L^{s-} \tag{70}$$
$$\tau_l^{s-} - e_i^{\xi,0} > -M z_{i,l}^{\xi,s-} \quad \forall\, \xi,\; i = 1, ..., I^\xi,\; s \in S^\xi, l = 1, ..., L^{s-} \tag{71}$$
$$\tag{72}$$

If $e_i^{\xi,0}$ is executed after $\tau_l^{s+}(\tau_l^{s-})$, $z_{i,l}^{\xi,s+}(z_{i,l}^{\xi,s-})$ is equal to one. If $e_i^{\xi,0}$ is executed before $\tau_l^{s+}(\tau_l^{s-})$, $\hat{z}_{i,l}^{\xi,s+}(\hat{z}_{i,l}^{\xi,s-})$ is equal to one.

If event e^ξ increases the value of s, and it is executed immediately when scheduled, i.e., $s \in \Theta^{\xi+}$ and $t^\xi = 0$, the following constraint should be applied:

$$e^{\xi,0} - \tau_{s_0+l-1-b}^{s-} \ge M(y_{i,l}^{\xi,1,s+} - 1) \tag{73}$$
$$e^{\xi,0} - \tau_{s_0+l-1-a}^{s-} \le M(1 - y_{i,l}^{\xi,1,s+}) \tag{74}$$

If event e^ξ decreases the value of s, and it is executed immediately when scheduled, i.e., $s \in \Theta^{\xi-}$ and $t^\xi = 0$, the following constraint should be applied:

$$e^{\xi,0} - \tau_{-s_0+l-1+a}^{s+} \ge M(y_{i,l}^{\xi,1,s-} - 1) \tag{75}$$
$$e^{\xi,0} - \tau_{-s_0+l-1+b}^{s+} \le M(1 - y_{i,l}^{\xi,1,s-}) \tag{76}$$

5.3. Mathematical programming representation of simulation model - V3

Event-scheduling algorithm for DES.

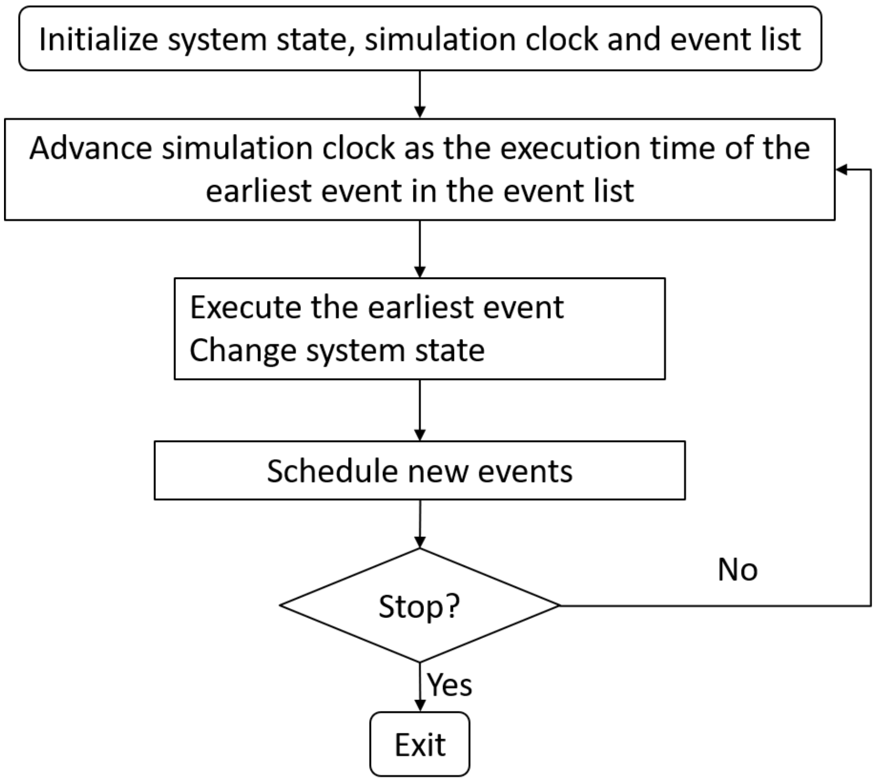

Figure 5. Event-based simulation algorithm.

An equivalent mathematical programming model exists if the following assumptions are satisfied:

(1) For all event e^{ξ}, the *scheduling conditions* are in the form of $a_s^{\xi} \leq s \leq b_s^{\xi}$ combined with logic operator "AND", where s is a state variable, and a_s^{ξ} and b_s^{ξ} are lower and upper bounds.

(2) The scheduling conditions is independent of the history and not changed along time. (It could be possible to define more state variables in case of history dependence and time-variant scheduling conditions.)

(3) An event execution of e^{ξ} leads to (integer) increment or decrement equal to Δ_s^{ξ} of certain state variables s, and Δ_s^{ξ} is not changed along time. (A direct evaluation can be modeled in this way.)

(4) The delay between scheduling and execution time of an event e^{ξ}, denoted by t^{ξ}, is random variate. They can be generated independently from the simulation run. (*This point is different from ERG. In ERG, the delay is dependent on the edge, i.e, a couple of events, but I consider delay dependent on a single event.*)

(5) For all events e^{ξ}, the number of executions N^{ξ} is known before simulation.

$e_i^{\xi,0} \geq 0$	i=1,...,I^ξ	the i-th scheduling time of event e^ξ.
$e_i^{\xi,1} \geq 0$	i=1,...,I^ξ	the i-th execution time of event e^ξ.
$\mathcal{E}_k \geq 0$	k=0,...,K	time of the k-th execution of any events.
$u_k^s \in \mathbb{Z}$	k=0,...,K	value of state variable s just after the k-th event.
$w_{i,k}^\xi \in \{0,1\}$	k=1,...,K	binding $e_i^{\xi,1}$ and $\mathcal{E}_k$.
$x_{i,k}^\xi \in \{0,1\}$	k=0,...,K	equal to one if $\mathcal{E}_k$ schedules $e^{\xi,0}$.
$y_{i,i'}^\xi \in \{0,1\}$		binding $e_i^{\xi,0}$ and $e_{i'}^{\xi,1}$ in case of overtaking.
$z_k^\xi \in \{0,1\}$	k=0,...,K	equal to one if the condition for scheduling e^ξ is true right after $\mathcal{E}_k$.
$v_k^{\xi,s,0} \in \{0,1\}$	k=0,...,K	equal to one if $s_k \leq a^{\xi,s} - 1$
$v_k^{\xi,s,1} \in \{0,1\}$	k=0,...,K	equal to one if $s_k \geq b^{\xi,s} - 1$
$r_k^\xi \in \mathbb{Z}$	k=0,...,K	number of existing parallel executions of $e_i^{\xi,1}$ after $\mathcal{E}_k$ before scheduling.
$n_k^\xi \in \mathbb{Z}$	k=0,...,K	number of scheduled executions of $e_i^{\xi,1}$ after $\mathcal{E}_k$ before scheduling.

Table 5. Notation

Constraints (A): binding $e_i^{\xi,1}$ and $\mathcal{E}_k$:

$$e_i^{\xi,1} - \mathcal{E}_k \geq M(w_{i,k}^\xi - 1) \quad A1 \quad \forall\, \xi, i, k \tag{77}$$

$$\mathcal{E}_k - e_i^{\xi,1} \geq M(w_{i,k}^\xi - 1) \quad A2 \quad \forall\, \xi, i, k \tag{78}$$

$$\sum_k w_{i,k}^\xi = 1 \quad A3 \quad \forall\, \xi, i \tag{79}$$

$$\sum_{\xi,i} w_{i,k}^\xi = 1 \quad A4 \quad \forall\, k \tag{80}$$

$$\sum_k k w_{i+1,k}^\xi - \sum_k k w_{i,k}^\xi \geq 1 \quad A5 \quad \forall\, \xi, i \tag{81}$$

Constraints (B): binding $e_i^{\xi,0}$ and $e_{i'}^{\xi,1}$, where α^ξ is the maximal number of executions existing simultaneously in the event list:

$$e_{i'}^{\xi,1} - e_i^{\xi,0} \geq t_i^\xi + M(y_{i,i'}^\xi - 1) \quad B1 \quad \forall \xi, i, i' = 1, ..., N^\xi \tag{82}$$

$$e_i^{\xi,0} - e_{i'}^{\xi,1} \geq -t_i^\xi + M(y_{i,i'}^\xi - 1) \quad B2 \quad \forall \xi, i, i' = 1, ..., N^\xi \tag{83}$$

$$\sum_{i=1}^{N^\xi} y_{i,i'}^\xi = 1 \quad B3 \quad \forall\, \xi, i' = 1, ..., N^\xi \tag{84}$$

$$\sum_{i'=1}^{N^\xi} y_{i,i'}^\xi = 1 \quad B4 \quad \forall\, \xi, i = 1, ..., N^\xi \tag{85}$$

$$\sum_{i'=i+\alpha^\xi}^{N^\xi} y_{i,i'}^\xi = 0 \quad B5 \quad \forall\, \xi,\ i = 1, ..., N^\xi - \alpha^\xi \tag{86}$$

$$\sum_{i=1}^{i'-\alpha^\xi} y_{i,i'}^\xi = 0 \quad B6 \quad \forall\, \xi,\ i' = \alpha^\xi + 1, ..., N^\xi \tag{87}$$

If $\alpha^\xi = 1$, variables $y^\xi_{i,i'}$ are redundant and constraints (B) are reduced to:

$$e_i^{\xi,1} - e_i^{\xi,0} = t_i^\xi \quad B1 \quad \forall \xi, i = 1, ..., N^\xi \tag{88}$$

Number of executions of event e^ξ waiting in the event list can be a state variable n^ξ, and one condition for scheduling an e^ξ is $n^\xi \leq \alpha^\xi$. Thus, it can be managed as a generic scheduling condition.

Constraints (C): event e^ξ can be scheduled right after $\mathcal{E}_k$ if all state variables s satisfies condition $a^\xi_s \leq s_k \leq b^\xi_s$.

$$e_i^{\xi,0} - \mathcal{E}_k \geq M(x^\xi_{i,k} - 1) \quad C1 \quad \forall\, \xi, k, i \tag{89}$$

$$\mathcal{E}_k - e_i^{\xi,0} \geq M(x^\xi_{i,k} - 1) \quad C2 \quad \forall\, \xi, k, i \tag{90}$$

$$\sum_k x^\xi_{i,k} = 1 \quad C3 \quad \forall\, \xi, i \tag{91}$$

$$b^{\xi,s} - u^s_k \geq M(z^\xi_k - 1) \quad C4 \quad \forall\, \xi, k, s \tag{92}$$

$$u^s_k - a^{\xi,s} \geq M(z^\xi_k - 1) \quad C5 \quad \forall\, \xi, k, s \tag{93}$$

$$u^s_k - (b^{\xi,s} + 1) \geq M(v_k^{\xi,s,1} - 1) \quad C6 \quad \forall\, \xi, k, s \tag{94}$$

$$(a^{\xi,s} - 1) - u^s_k \geq M(v_k^{\xi,s,0} - 1) \quad C7 \quad \forall\, \xi, k, s \tag{95}$$

$$1 - z^\xi_k \leq \sum_{s \in S^\xi} v_k^{\xi,s,0} + \sum_{s \in S^\xi} v_k^{\xi,s,1} + v_k^{\xi,r} + v_k^{\xi,N} \quad C8 \quad \forall\, \xi, k \tag{96}$$

$$\sum_{i=1}^{N^\xi} x^\xi_{i,k} = z^\xi_k \quad C9 \quad \forall\, \xi, k \tag{97}$$

$$\sum_k k x^\xi_{i+1,k} - \sum_k k x^\xi_{i,k} \geq 1 \quad C10 \quad \forall\, \xi, i \tag{98}$$

Constraints (D): evolution of state variables

$$u^s_k = u^s_{k-1} + \sum_\xi \sum_{i=1}^{N^\xi} w^\xi_{i,k} \Delta^{\xi,s} \quad D1 \quad \forall\, s, k \tag{99}$$

$$r^\xi_k = r^\xi_{k-1} + X^\xi_{k-1} - \sum_i w^\xi_{i,k} \quad D2 \quad \forall\, \xi, k \tag{100}$$

$$R^\xi - r^\xi_k \geq z^\xi_k \quad D3 \quad \forall\, \xi, k \tag{101}$$

$$r^\xi_k \geq R^\xi v_k^{\xi,r} \quad D4 \quad \forall\, \xi, k \tag{102}$$

$$n^\xi_k = n^\xi_{k-1} + X^\xi_{k-1} \quad D5 \quad \forall\, \xi, k \tag{103}$$

$$N^\xi - n^\xi_k \geq z^\xi_k \quad D6 \quad \forall\, \xi, k \tag{104}$$

$$n^\xi_k \geq N^\xi v_k^{\xi,N} \quad D7 \quad \forall\, \xi, k \tag{105}$$

Constraints (E): others

$$\mathcal{E}_0 = 0 \quad E1 \tag{106}$$

$$\mathcal{E}_k - \mathcal{E}_{k-1} \geq 0 \quad E1 \quad \forall\, k \tag{107}$$

Objective function: with the constraints above, there is a unique solution in terms of event occurring times (solution of the binary variables could be multiple in case of multiple simultaneous events). Thus, the objective can be any function of event occurring time. I tried minimize/maximize the sum of $\mathcal{E}_k$, and they give the same solution.

Conditions for a variable x to be *resource-type* are not valid any more.

(1) $\forall\ \xi$ and s, upper bound b_s^ξ is monotonically increasing on x.
(2) $\forall\ \xi$ and s, lower bound a_s^ξ is monotonically decreasing on x.

The reason is that increasing b_s^ξ or decreasing a_s^ξ will tighten constraints C6 and C7. To be simple, we consider b only.

$$b - u_k^s \geq M(z_k^\xi - 1) \quad C4 - b \quad \forall\ \xi, k, s \tag{108}$$
$$u_k^s - (b+1) \geq M(v_k^{\xi,s,1} - 1) \quad C6 - b \quad \forall\ \xi, k, s \tag{109}$$

(when $u = b + 1$, event e^ξ cannot be scheduled.)

If b is increased to $b + 1$:

$$(b+1) - u_k^s \geq M(z_k^\xi - 1) \quad C4 - (b+1) \quad \forall\ \xi, k, s \tag{110}$$
$$u_k^s - (b+2) \geq M(v_k^{\xi,s,1} - 1) \quad C6 - (b+1) \quad \forall\ \xi, k, s \tag{111}$$

(when $u = b + 1$, event e^ξ must be scheduled.)

A group of relaxed constraints are:

$$(b+1) - u_k^s \geq M(z_k^\xi - 1) \quad C4 - (b+1) \quad \forall\ \xi, k, s \tag{112}$$
$$u_k^s - (b+1) \geq M(v_k^{\xi,s,1} - 1) \qquad C6 - (b) \qquad \forall\ \xi, k, s \tag{113}$$

(when $u = b + 1$, event e^ξ can be scheduled or not.)

Todo:

(1) What kind of performance indicators can be used? (Regular function of time, in scheduling area. Weighted sum, maximum. Refer to book on scheduling.)

5.4. Merge

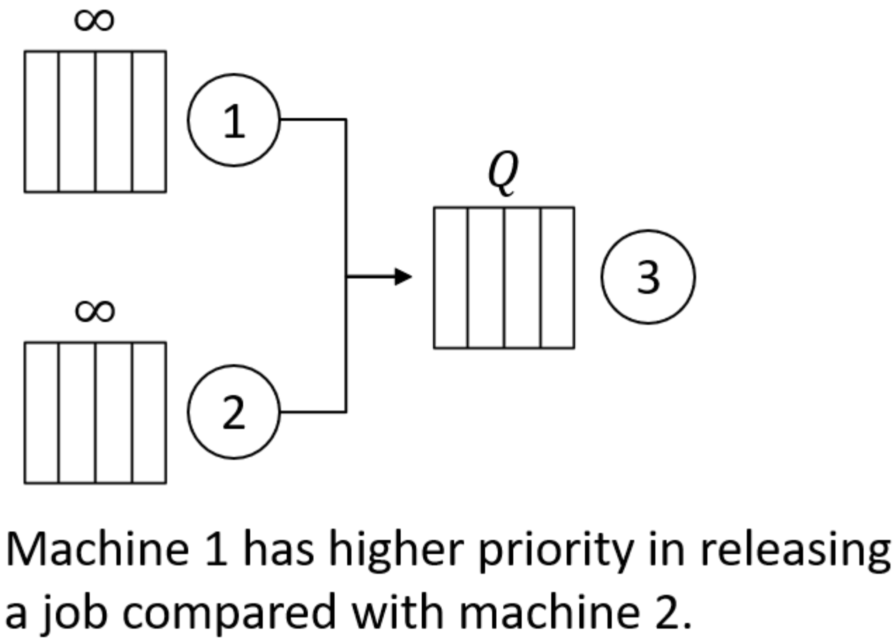

Figure 6. Example: merge.

Variable	Event	Condition to schedule	Delay	# executions	State change
$e^{s,1}$	Start m1	$m^1 \leq 0$	0	1	m^1++
$e^{f,1}$	Finish m1	$1 \leq m^1 \leq 1$	t^1	1	m^1++
$e^{d,1}$	Depart m1	$m^1 \geq 2\ AND$ $q \geq 1$	0	1	$m^1 = m^1 - 2,\ q--$
$e^{s,2}$	Start m2	$m^2 \leq 0$	0	1	m^2++
$e^{f,2}$	Finish m2	$1 \leq m^2 \leq 1$	t^2	1	m^2++
$e^{d,2}$	Depart m2	$m^2 \geq 2\ AND$ $q \geq 1\ AND$ $m^1 \leq 1$	0	1	$m^2 = m^2 - 2,\ q--$
$e^{s,3}$	Start m3	$m^3 \leq 0\ AND$ $q \leq Q - 1$	0	1	$m^3++,\ q++$
$e^{d,3}$	Depart m3	$m^3 \geq 1$	t^3	1	m^3--

Table 6. Merge-S3M111

MP model:

$$\min \sum_k \mathcal{E}_k \tag{114}$$

$$e_i^{(\xi,j),1} - \mathcal{E}_k \geq M(w_{i,k}^{\xi,j} - 1) \qquad \xi \in \{s,f,d\}, j \in \{1,2,3\}, \forall i,k \tag{115}$$

$$\mathcal{E}_k - e_i^{(\xi,j),1} \geq M(w_{i,k}^{\xi,j} - 1) \qquad \xi \in \{s,f,d\}, j \in \{1,2,3\}, \forall i,k \tag{116}$$

$$\sum_k w_{i,k}^{\xi,j} = 1 \qquad \forall\, \xi \in \{s,f,d\}, j \in \{1,2,3\}, i \tag{117}$$

$$\sum_{(\xi,j),i} w_{i,k}^{\xi,j} = 1 \qquad \forall\, k \tag{118}$$

$$\sum_k k w_{i+1,k}^{\xi,j} - \sum_k k w_{i,k}^{\xi,j} \geq 1 \qquad \forall\, \xi \in \{s,f,d\}, j \in \{1,2,3\}, i \tag{119}$$

$$e_i^{s,j,1} - e_i^{s,j,0} \geq 0 \qquad j = 1,2,3, \forall\, i \tag{120}$$

$$e_i^{f,j,1} - e_i^{f,j,0} \geq t_i^j \qquad j = 1,2, \forall\, i \tag{121}$$

$$e_i^{d,j,1} - e_i^{d,j,0} \geq 0 \qquad j = 1,2, \forall\, i \tag{122}$$

$$e_i^{d,3,1} - e_i^{d,3,0} \geq t_i^3 \qquad \forall\, i \tag{123}$$

$$e_i^{\xi,j,0} - \mathcal{E}_k \geq M(x_{i,k}^{\xi,j} - 1) \qquad \forall\, \xi \in \{s,f,d\}, j = 1,2,3, k, i \tag{124}$$

$$\mathcal{E}_k - e_i^{\xi,j,0} \geq M(x_{i,k}^{\xi,j} - 1) \qquad \forall\, \xi \in \{s,f,d\}, j = 1,2,3, k, i \tag{125}$$

$$m_k^j = m_{k-1}^j + \sum_{i=1}^{N^j} (w_{i,k}^{s,j} + w_{i,k}^{f,j} - 2w_{i,k}^{d,j}) \qquad j = 1,2, \forall\, k \tag{126}$$

$$m_k^3 = m_{k-1}^3 + \sum_{i=1}^{N^3} (w_{i,k}^{s,3} - w_{i,k}^{d,3}) \qquad \forall\, k \tag{127}$$

$$q_k = q_{k-1} + \sum_{i=1}^{N^3} w_{i,k}^{s,3} - \sum_{i=1}^{N^1} w_{i,k}^{d,1} - \sum_{i=1}^{N^2} w_{i,k}^{d,2} \tag{128}$$

$$m_k^j \geq M(z_k^{s,j} - 1) \qquad j = 1,2,3, \forall\, k \tag{129}$$

$$1 - m_k^j \geq M(z_k^{f,j} - 1) \qquad j = 1,2, \forall\, k \tag{130}$$

$$m_k^j - 1 \geq M(z_k^{f,j} - 1) \qquad j = 1,2, \forall\, k \tag{131}$$

$$m_k^j - 2 \geq M(z_k^{d,j} - 1) \qquad j = 1,2, \forall\, k \tag{132}$$

$$q_k - 1 \geq M(z_k^{d,j} - 1) \qquad j = 1,2, \forall\, k \tag{133}$$

$$1 - m_k^1 \geq M(z_k^{d,2} - 1) \qquad \forall\, k \tag{134}$$

$$m_k^3 - 1 \geq M(z_k^{d,3} - 1) \qquad \forall\, k \tag{135}$$

$$(Q-1) - q_k \geq M(z_k^{s,3} - 1) \qquad \forall\, k \tag{136}$$

$$\sum_k x_{i,k}^{\xi,j} = 1 \qquad \forall\, \xi \in \{s,f,d\}, j = 1,2,3, \forall\, i,k \tag{137}$$

$$\sum_{i=1}^{N^j} x_{i,k}^{\xi,j} \leq z_k^{\xi,j} \qquad \forall\, \xi \in \{s,f,d\}, j = 1,2,3, k \tag{138}$$

$$\sum_k k x_{i+1,k}^{\xi,j} - \sum_k k x_{i,k}^{\xi,j} \geq 0 \qquad \forall\, \xi \in \{s,f,d\}, j = 1,2,3, i \tag{139}$$

5.5. Merge - 2 machines in station 3

Variable	Value	Initialization	Description
e^1	0,1,2	2	number of empty machines in station 1
f^1	0,1,2	0	number of finished jobs in station 1
e^2	0,1	1	number of empty machines in station 2
f^2	0,1	0	number of finished jobs in station 2
e^3	0,1	1	number of empty machines in station 3
q	0,...,Q	Q	number of available spaces in queue

Table 7. State variables: Merge-S3M211

Variable	Event	Condition to schedule	Delay	# executions	State change
ai $e^{f,1}$	Finish m1	$1 \leq e^1 \leq 2$	t^1	2	$f^1 ++$
$e^{d,1}$	Depart m1	$1 \leq f^1 \leq 2\ AND$ $1 \leq q \leq Q$	0	1	$e^1 ++,\ f^1 --,\ q--$
$e^{s,2}$	Start m2	$1 \leq e^2 \leq 1$	0	1	$e^2 --$
$e^{f,2}$	Finish m2	$1 \leq e^2 \leq 1$	t^2	1	$f^2 ++$
$e^{d,2}$	Depart m2	$1 \leq f^2 \leq 1\ AND$ $1 \leq q \leq Q\ AND$ $0 \leq f^1 \leq 0$	0	1	$f^2 --, e^2 ++,\ q--$
$e^{s,3}$	Start m3	$1 \leq e^3 \leq 1\ AND$ $0 \leq q \leq Q-1$	0	1	$e^3 --,\ q++$
$e^{d,3}$	Depart m3	$1 \leq e^3 \leq 1\ AND$	t^3	1	$e^3 ++$
		$0 \leq q \leq Q-1$			

Table 8. Events: Merge-S3M211

5.6. Failure

5.7. Jobshop

5.8. Identifying Resource-type variables

6. Gradient-based approximate cut

6.1. Gradient estimation

6.2. Gradient-based feasibility cut

7. Combinatorial cut generation

7.1. Combinatorial cut

7.2. Heuristic for tightening Exact combinatorial cut

8. Feasibility-cut-based algorithm

The complete algorithm for solving RAP–PC is summarized in Algorithm 1. The resource capacities are initialized to the lower bound. The searching region of RAP–PC–MIP is initialized to $\mathbb{X}$, and the lower and upper bounds of the objective function, C^L and C^U, respectively, are set considering the upper bound and lower bound of the capacity of each resource. Lines 7 to 11 show that approximate cuts are generated and used in the model when infeasible solutions are found. Once a feasible solution is found, the upper bound C^U, which is also the incumbent solution, can be updated after comparing the value of the found feasible solution and that of the current incumbent. Then, all the currently used approximate cuts are replaced by exact cuts of the DIS. If there are only exact cuts in RAP–PC–MIP, the solution is the new lower bound C^L. The algorithm terminates when the gap between the upper bound and lower bound is within a tolerance or the time limit is exceeded.

9. Numerical analysis

10. Conclusion

References

Alfieri, A., & Matta, A. (2012). Mathematical programming formulations for approximate simulation of multistage production systems. *European Journal of Operational Research*, *219*(3), 773–783.

Alfieri, A., Matta, A., & Pastore, E. (2020). The time buffer approximated buffer allocation problem: A row–column generation approach. *Computers & Operations Research*, *115*, 104835.

Chan, W. K., & Schruben, L. (2008). Optimization models of discrete-event system dynamics. *Operations Research*, *56*(5), 1218–1237.

Law, A. M., Kelton, W. D., & Kelton, W. D. (2000). *Simulation modeling and analysis* (Vol. 3). McGraw-Hill New York.

Matta, A. (2008). Simulation optimization with mathematical programming representation of discrete event systems. In *2008 winter simulation conference* (pp. 1393–1400).

Pedrielli, G., Alfieri, A., & Matta, A. (2015). Integrated simulation–optimisation of pull control systems. *International Journal of Production Research*, *53*(14), 4317–4336.

Tan, B. (2015). Mathematical programming representations of the dynamics of continuous-flow production systems. *IIE Transactions*, *47*(2), 173–189.

Algorithm 1 MIP–based algorithm.

Input:

Lower bound $\mathbf{a} = [a_1, ..., a_J]$ and upper bound $\mathbf{b} = [b_1, ..., b_J]$ of resource capacity $\mathbf{x}$, such that $a_j \leq x_j \leq b_j \ \forall \ j = 1, ..., J$.

Tolerance of optimality gap ε_{opt}.

Optional input: time limit of the algorithm T_{lim}.

Ensure:

Sample-path global optimal $\mathbf{x}^*$.

```
Initialize system with lower bound x ← a
Initialize incumbent with upper bound x* ← b.
Initialize lower bound of the objective C^L ← c^T a.
Initialize upper bound of the objective C^U ← c^T b.
Add initial constraints which defines 𝕏 to the RAP–PC–MIP.
while C^U − C^L > ε_opt and T_lim is not exceeded. do
   while There exists at least one violated performance constraint do
      Generate one approximate cut CA(x̄, l) for each violated constraints l and
      add all the generated cuts to the RAP–PC–MIP.
      x̄ ← solution of the RAP–PC–MIP.
      Simulate the system of x̄.
   end while
   Update upper bound and incumbent C^U ← c^T x̄, x* ← x̄ if c^T x̄ < C^U.
   if There exist approximate cuts in RAP–PC–MIP then
      For all the currently used approximate cuts CA(x̄^r, l), find dominating infea-
      sible solution x̄_d(x̄^r) and replace approximate cuts CA(x̄^r, l) by exact cuts
      CE(x̄_d(x̄^r), l) of the DIS.
      x̄ ← solution of the RAP–PC–MIP.
      Simulate the system of x̄.
      Update lower bound C^L ← max{c^T x̄, C^L}.
   end if
end while
```


Weiss, S., & Stolletz, R. (2015). Buffer allocation in stochastic flow lines via sample-based optimization with initial bounds. *OR spectrum*, *37*(4), 869–902.

Zhang, M., & Matta, A. (2018). Data-driven downtime bottleneck detection in open flow lines. In *2018 ieee 14th international conference on automation science and engineering (case)* (pp. 1513–1518).

Zhang, M., & Matta, A. (2020). Models and algorithms for throughput improvement problem of serial production lines via downtime reduction. *IISE Transactions*, 1–15.

Zhang, M., Matta, A., & Alfieri, A. (2020). Simulation optimization with mathematical programming representation of discrete event systems. In *2020 winter simulation conference* (p. accepted).

Zhang, M., Pastore, E., Matta, A., & Alfieri, A. (n.d.). Buffer allocation problem in production flow lines: a new benders decomposition based exact solution approach. *under review*.