Adjustable Robust Optimization for Multi-tasking Scheduling with Reprocessing due to Imperfect Tasks

Nikos H. Lappas,[1] Luis A. Ricardez-Sandoval,[2] Ricardo Fukasawa,[3] Chrysanthos E. Gounaris[1,*]

[1]*Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213*
[2]*Department of Chemical Engineering, University of Waterloo, Waterloo, Canada*
[3]*Combinatorics & Optimization Department, University of Waterloo, Waterloo, Canada*

Abstract

This work contemplates the optimal scheduling of multi-tasking production environments where the processing tasks are subject to uncertain success rates. Such problems arise in many industrial applications that have the potential to yield non compliant products, which must then be reprocessed. We address this problem by mapping the multi-tasking sequential recipe into a State-Task Network representation that includes suitably defined recycle streams to accommodate the option for reprocessing. This allows us to utilize a variant of an established global-event continuous time scheduling formulation to model the overall problem, as well as to employ an Adjustable Robust Optimization framework to account for the uncertainty in the production yields associated with each processing task. We assess the computational performance of the proposed approach via a comprehensive study that involves a large database of multi-tasking scheduling benchmark problems, and we demonstrate that instances involving more than 100 uncertain parameters can be addressed within reasonable computational times. Our results also help elucidate the expected amount of cost premium to insure against various levels of uncertainty in the production success rates.

Keywords: Process scheduling, Multi-tasking scheduling, Robust optimization.

[1] Author to whom all correspondence should be addressed (gounaris@cmu.edu).

1 Introduction

The optimal scheduling of operations has progressively become more challenging due to the ever-increasing complexity of production networks and the agile competition that necessitates the transition to more versatile production schemes [Harjunkoski et al., 2014]. As a result, a variety of short-term, multi-purpose process scheduling models have been proposed in the literature to address numerous real-life production environments that feature limited, yet flexible resources [Kallrath, 2002, Floudas and Lin, 2005, Méndez et al., 2006, Maravelias, 2012]. Despite this progress, however, one notable setting that had eluded investigation up until recently pertained to production environments that utilize processing units in which multiple tasks can be processed simultaneously.

Among other applications, this setting is common in the scientific services industry, where samples stemming from different orders must undergo a number of processing steps (tasks) that are typically specific to each order. These tasks are executed by a common pool of processing units (machines), some or all of which may be capable to simultaneously handle samples that originate from different orders. This unique capability of allowing a machine to undergo concurrently a set of distinct tasks grants the characterization of *multi-tasking* for the corresponding scheduling problem. Furthermore, an important requirement in this context is the preservation of each order's *batch integrity* throughout the recipe, a feature that is also crucial for the pharmaceutical and biotech industry [Sundaramoorthy and Maravelias, 2011]. Given the above complexities, multi-tasking scheduling was initially treated as a multi-commodity flow problem with a discrete time grid by Patil et al. [2015], and more recently, as a special case of a multi-purpose sequential environment using a modified continuous time, slot-based formulation by Lagzi et al. [2017a,b]. While these approaches yielded promising results for the deterministic case of the problem, parameter uncertainty was not considered.

Such uncertainty may originate from variations in raw materials properties, fluctuations in operating conditions of the processing units, or inconsistencies in the quality of supplied utilities. These factors may in turn result in imperfect products that require reprocessing (complete or partial) until they eventually meet the specifications. In fact, the efficient handling of parameter uncertainty is vital in many process scheduling contexts, and since optimization under nominal conditions can lead to suboptimal or infeasible solutions in view of the actual realized values of the uncertain parameters, a systematic uncertainty mitigation methodology is necessary.

Multiple approaches have been proposed in the literature to address uncertainty in process scheduling problems [Li and Ierapetritou, 2008a, Verderame et al., 2010, Grossmann et al., 2016]. The most straightforward way to do so is to perform rescheduling (either on a regular basis, or after "trigger" events), reacting upon observed deviations from the nominal schedule [Mendez and Cerdá, 2004]. In practice, such an approach would rely on heuristic rules, since the available time for executing the full-scale optimization can be prohibitive in light of the frequency with which the plant must react [Gupta et al., 2016]. A different class of solution methods relies on the fundamental approach of Stochastic Programming, where uncertainty distributions are discretized to create representative scenarios with corresponding probability weights, after which one solves a single optimization model that aims to optimize performance in expectation [Ierapetritou and Pistikopoulos, 1996, Grossmann and Balasubramanian, 2004]. However, for cases where detailed (joint) probability distributions are not available, this approach becomes less practical. One alternative approach is Multi-parametric Programming that maps the uncertain parameters space to critical regions with dedicated optimal solutions for each such region [Ryu et al., 2007, Li and Ierapetritou, 2008b], offering operators with the opportunity to a-priori select one of these available solutions that corresponds to the best trade-off between cost and risk. Depending on the application, however, the size and number of these critical regions can pose hurdles from both a computational tractability and a solution implementation point of view. In addition to the above methodologies, traditional Robust Optimization (RO) has also been applied in the context of process scheduling under uncertainty [Lin et al., 2004, Li and Ierapetritou, 2008c, Guzman et al., 2016]. Here, one seeks production plans that are to remain feasible under any realization of uncertainty that one assumes to be possible via an a-priori chosen uncertainty set. Whereas the RO approach ensures one's ability to implement the resulting solution in a non-anticipative manner, this solution may often be overly conservative from a cost perspective. Furthermore, the single-stage nature of this approach makes it unsuitable to address uncertainty in parameters that participate in equality constraints (see, e.g., Gorissen et al. [2015]), such as task production yields, which are typically referenced as coefficients in various mass balances.

In order to alleviate the inherent conservatism of traditional RO solutions, the concept of Adjustable Robust Optimization (ARO) was introduced by Ben-Tal et al. [2004]. The fundamental idea behind ARO is the classification of decisions as either *here-and-now*, which have to be made before any uncertain parameter is realized, or as *wait-and-see*, which are committed to at later points in time, and thus, can be adapted to the observed values of those uncertain parameters that have by then already been realized. Continuous wait-and-see decisions, in particular, can be chosen based on an optimal affine relationship to these observed values, commonly referred to as the *affine decision rules* approach. The latter was recently applied in the context of multi-stage process scheduling by Lappas and Gounaris [2016], where it was shown to perform up to 50% better compared to traditional RO in terms of worst-case costs, while providing additional improvements for costs in expectation. Importantly, the multi-stage ARO framework made it possible to accommodate uncertainty in production yields [Lappas and Gounaris, 2018a] for the first time in a robust optimization approach. This ability of the ARO framework makes it an ideal candidate to address the setting of multi-tasking scheduling with variability in task outcomes, which is effectively a form of production yield uncertainty.

This work builds upon and extends this ARO framework for process scheduling. Our specific contributions are as follows:

- We extend the State-Task Network representation as well as adapt a global event-based process scheduling model to support multi-tasking production environments.
- We introduce a systematic way of modeling imperfect tasks via the introduction of recycle streams in conjunction with uncertain variability in the associated production yield parameters.
- We augment deterministic multi-tasking scheduling instances from the literature to accommodate imperfect tasks, and we construct realistic decision-dependent polyhedral uncertainty sets to capture the observed variability of their success rates.
- We perform a comprehensive computational study that elucidates our proposed approach's computational performance, establishing for the first time robust optimal solutions and quantifying the associated price of robustness in such problems.

The remainder of this study is organized as follows. In Section 2, we provide an overview of multi-tasking scheduling, discuss the construction of State-Task Networks from multi-tasking sequential recipes, and present a deterministic global event-based model that is capable of accommodating multi-tasking operations. In Section 3, we discuss how to model the need for reprocessing due to uncertain success rates as uncertainty in production yields, showcase the construction of decision-dependent uncertainty sets that are suitable for this context, and investigate approaches for reducing the number of uncertain parameters. We then present our comprehensive computational study in Section 4, and lastly, we conclude in Section 5 with some final remarks.

2 Multi-tasking Scheduling

In traditional multi-purpose plant scheduling, units are capable of performing different processing tasks (e.g., originating from different recipe paths or orders) but are limited to executing at most one task at any given time. Multi-tasking scheduling (MTS) generalizes this setting by allowing units to execute multiple tasks at the same time. For example, consider the scientific services industry and a laboratory's sterilizing oven. In this context, the laboratory handles samples from different origins that have to undergo different sets of analysis tasks, yet all of them need to be sterilized; hence, it is common for samples from various batches to undergo sterilization in the same oven at the same time, sharing the unit's total capacity. Another common example of multi-tasking is found in applications that involve chambers to create vacuum or some other controlled environment. Arguably, MTS is relevant to a multitude of discrete manufacturing contexts. It should be highlighted, however, that modeling these cases within the confines of traditional multi-purpose plant scheduling models can lead to suboptimal solutions, since the equipment would be underutilized, as shown by Lagzi et al. [2017a]. Therefore, explicit consideration of a unit's multi-tasking capabilities in the model is necessary.

Furthermore, it is important to work with multi-tasking scheduling models that allow for batch integrity to be maintained, in the sense that material flows belonging to one order cannot be utilized to satisfy the demand for a different order. While similar problems have been addressed by sequential-environment formulations [Méndez and Cerdá, 2003, Ferrer-Nadal et al., 2008, Lee and Maravelias, 2017], these approaches are based on the assumption that the batching problem (the division of orders to an optimal number of batches) has been solved a-priori. In this work, we will leverage the ability of network-environment scheduling formulations [Floudas and Lin, 2005, Méndez et al., 2006] to solve simultaneously the batching and sequencing problems, augmenting them as necessary to satisfy the additional requirement for multi-tasking.

2.1 Mapping of Sequential Multi-purpose Recipes to State-Task Networks

The first step towards utilizing a network-environment scheduling formulation is to properly set up the corresponding State-Task Network (STN). In order to better illustrate this procedure, we will utilize the motivating example from the work of Lagzi et al. [2017a]. In this example, there exist two orders, A and B, of sizes 100 and 120, respectively, which have to go through different processing paths, as shown in Figure 1. More specifically, order A has to go through all four processes (red path), while order B shall bypass the second process (green path). As mentioned above, it is essential for the model to keep track of the orders throughout the processing steps. While this is clearly represented in the Operations Figure, the transition to an STN representation requires some careful attention. Specifically, after the set of unique processes has been identified (in this example, four unique processes) and ordered arbitrarily, we need to introduce as many states as the cardinality of this set plus one (in this example, five states) so as to ensure that each process can be linked to a state from which it draws material and to a state to which it stores its products.[1] Next, each state has to be disaggregated to as many "sub-states" as the number of orders whose path passes through this state. In the example of Figure 1, each of the five states are disaggregated to two sub-states, with the exception of the state that feeds the second process, which can only have material originating from order A, and hence corresponds to a single sub-state. Finally, the processes have to be connected to the corresponding states as illustrated in Figure 1. Whereas this conversion of an Operations Figure to an STN does not offer more information, the STN representation comes with the additional benefit of being able to accommodate recycles, which as we shall see later will be important for handling uncertainty due to imperfect tasks. It should also be noted that the above conversion procedure is generic and can be utilized for any sequential-environment that needs to be mapped to an STN.

[1] Note that, typically, these states do not correspond to actual storage facilities, and thus, are not subject to storage capacity constraints. If actual storage limitations apply, these can be easily incorporated as upper bounds on the variables that shall represent the state levels.

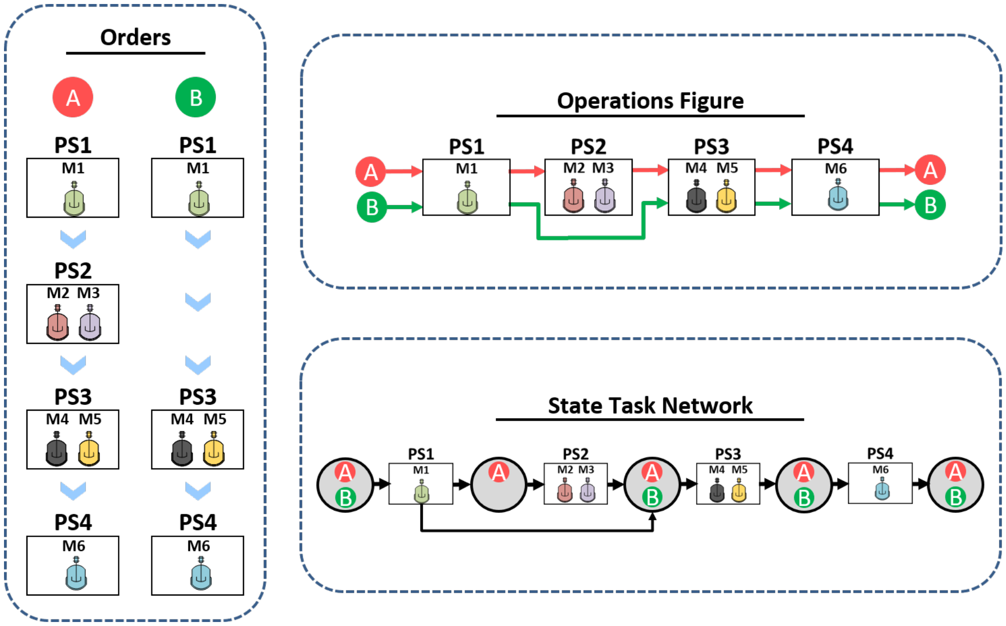


Figure 1: Motivating example from Lagzi et al. [2017a]. The orders with their corresponding processing steps and compatible machines are illustrated on the left. The resulting Operations Figure and its equivalent State-Task Network are shown on the right.

2.2 Proposed Deterministic Formulation

Given an STN network, any network-based, multi-purpose scheduling model can be utilized to solve sequential multi-purpose problems. In this work, we will utilize the continuous-time model by Castro et al. [2004], which we will adapt to capture the multi-tasking nature of the setting under investigation. The full model is described via Eqs. (1–18), and the corresponding nomenclature can be found in Appendix A. The main difference between our model and the original model presented in the cited reference pertains to the introduction of order-specific sub-batches and sub-states and their proper handling through the mass balances. It has to be highlighted that the adapted model features an increased number of continuous variables (sub-batch sizes and sub-state levels), but no new binary variables are introduced compared to the original formulation of Castro et al. [2004]. This is especially important as it preserves the overall tractability characteristics of the formulation. We shall also note that the proposed model can easily accommodate, if so desired, the full spectrum of the original model's capabilities, including multiple storage and zero-wait policies as well as the tracking of utilities consumption. However, for ease of exposition, as well as given the fact that the multi-tasking scheduling literature benchmarks typically do not require them, these aspects will not be covered in this work.

$$\min_{\substack{W,BO,SO\\G,T,z}} z \tag{1}$$

s.t.

$$z \geq \sum_{k\in\mathcal{K}} \sum_{s\in\mathcal{S}_k} P_{k,s} \left(SO_{k,s,0} - SO_{k,s,N} \right) \tag{2}$$

$$z \geq T_N - T_1 \tag{3}$$

$$G_{j,n} = G_{j,n-1} + \sum_{i\in\mathcal{I}_j} \left(\sum_{n'\in\mathcal{N}_n^+} W_{i,n,n'} - \sum_{n'\in\mathcal{N}_n^-} W_{i,n',n} \right) \quad \forall\, n \in \mathcal{N} : \{n > 1\}, \qquad \forall\, j \in \mathcal{J} \tag{4}$$

$$T_{n'} - T_n \geq \sum_{i\in\mathcal{I}_j} \left(\alpha_i W_{i,n,n'} + \beta_i \sum_{k\in\mathcal{K}_i} BO_{i,k,n,n'} \right) \quad \forall\, n' \in \mathcal{N}_n^+, \quad \forall\, n \in \mathcal{N}, \quad \forall\, j \in \mathcal{J} \tag{5}$$

$$\sum_{i\in\mathcal{I}_j} \sum_{\substack{n'\in\mathcal{N}:\\ \{n'\geq n\}}} \sum_{n''\in\mathcal{N}_{n'}^+} \left(\alpha_i W_{i,n',n''} + \beta_i \sum_{k\in\mathcal{K}_i} BO_{i,k,n',n''} \right) \leq T_N - T_n \quad \forall\, n \in \mathcal{N}, \quad \forall\, j \in \mathcal{J} \tag{6}$$

$$\begin{aligned} SO_{k,s,n} &= SO_{k,s,n-1} \\ &\quad + \sum_{i\in\mathcal{I}_{k,s}^p} \rho_{i,k,s,n}^p \sum_{n'\in\mathcal{N}_n^-} BO_{i,k,n'n} \\ &\quad - \sum_{i\in\mathcal{I}_{k,s}^c} \rho_{i,k,s,n'}^c \sum_{n'\in\mathcal{N}_n^+} BO_{i,k,n,n'} \qquad \forall\, n \in \mathcal{N}, \quad \forall\, s \in \mathcal{S}_k, \quad \forall\, k \in \mathcal{K} \end{aligned} \tag{7}$$

$$W_{i,n,n'} B_i^{min} \leq \sum_{k\in\mathcal{K}_i} BO_{i,k,n,n'} \leq W_{i,n,n'} B_i^{max} \quad \forall\, n' \in \mathcal{N}_n^+, \quad \forall\, n \in \mathcal{N}, \quad \forall\, i \in \mathcal{I} \tag{8}$$

$$SO_{k,s,N} \geq D_{k,s} \quad \forall\, s \in \mathcal{S}_k, \quad \forall\, k \in \mathcal{K} \tag{9}$$

$$T_1 = 0 \tag{10}$$

$$T_N = H \tag{11}$$

$$G_{j,N} = 0 \quad \forall\, j \in \mathcal{J} \tag{12}$$

$$z \in \mathbb{R} \tag{13}$$

$$T_n \geq 0 \quad \forall\, n \in \mathcal{N} \tag{14}$$

$$0 \leq G_{j,n} \leq 1 \quad \forall\, n \in \mathcal{N}, \quad \forall\, j \in \mathcal{J} \tag{15}$$

$$SO_{k,s,n} \geq 0 \quad \forall\, n \in \mathcal{N}, \quad \forall\, s \in \mathcal{S}_k, \quad \forall\, k \in \mathcal{K} \tag{16}$$

$$BO_{i,k,n,n'} \geq 0 \quad \forall\, n' \in \mathcal{N}_n^+, \quad \forall\, n \in \mathcal{N}, \quad \forall\, k \in \mathcal{K}_i, \quad \forall\, i \in \mathcal{I} \tag{17}$$

$$W_{i,n,n'} \in \{0, 1\} \quad \forall\, n' \in \mathcal{N}_n^+, \quad \forall\, n \in \mathcal{N}, \quad \forall\, i \in \mathcal{I} \tag{18}$$

In the above model, we use the auxiliary variable z to encode the value of the objective function to be minimized (Eq. 1), which can flexibly capture either the maximization of profit (expressed equivalently as minimizing negative profit, Eq. 2) or the minimization of makespan (Eq. 3). Note that the two objectives are meant to be used exclusively to each other. Eqs. 4 ensure that the proper utilization status of each processing unit is set for each event point, depending on whether any processing task is taking place in that unit. Eqs. 5 are responsible for assigning actual time values to the available event points, whereas the model can be tightened by the (optional) use of Eqs. 6, which capture the fact that, for each event point and processing unit, the total duration of the remaining tasks cannot exceed the remaining scheduling horizon. Eqs. 7 correspond to mass balances for all relevant sub-states, appropriately accounting for the order-specific production and consumption yields of each task. Eqs. 7 restrict the batch-sizes to the applicable capacity limits, while Eqs. 9 provide final production targets for the case of makespan minimization. Eqs. 10–12 provide fixed values for certain variables, while Eqs. 13–18 declare the domains of the decision variables. We remark that Eqs. 11 and 2 are only to be accounted for in the case of maximization of profit objective, while Eqs. 3 and 9 are only to be accounted in the case of minimization of makespan objective. We also remark that the above model treats all batch sizes and state levels as continuous variables. For discrete manufacturing environments, batch sizes and state levels can be suitably rounded after the optimal solution has been determined, without significant loss of precision. Finally, we highlight that, as is common practice with such scheduling models, the tasks have been duplicated as necessary to be specific to given processing units; that is, $\mathcal{I}_j \cap \mathcal{I}_{j'} = \emptyset, \forall j \neq j'$. This approach, which can be easily implemented during an input preprocessing step, also readily allows for the utilization of non-identical machines where the various task-related parameters (e.g., processing times, yields, capacities) may differ from unit to unit.

3 Reprocessing of Imperfect Tasks

A typical assumption in process scheduling models is that tasks are perfect inasmuch as they always result in products of desired specifications. Nevertheless, this is not always the case in real life settings, where production quality can fluctuate over time due to a variety of reasons, ranging from equipment malfunctions to process dynamics, and leading to nonconforming products. For example, in the scientific services industry, it is frequently the case that a fraction of the analyzed samples are detected as nonconforming, necessitating that they repeat a subset of the preceding analysis steps before the results are finalized for the clients. The samples that need to be reprocessed impose to the system an additional workload, which if not accounted for in the overall schedule, could lead to infeasible operations, e.g., exceeding the capacity of a processing unit or missing a deadline constraint. To that end, it is necessary to develop a systematic methodology that accounts for imperfect tasks in a way that does not introduce severe reductions in the overall productivity.

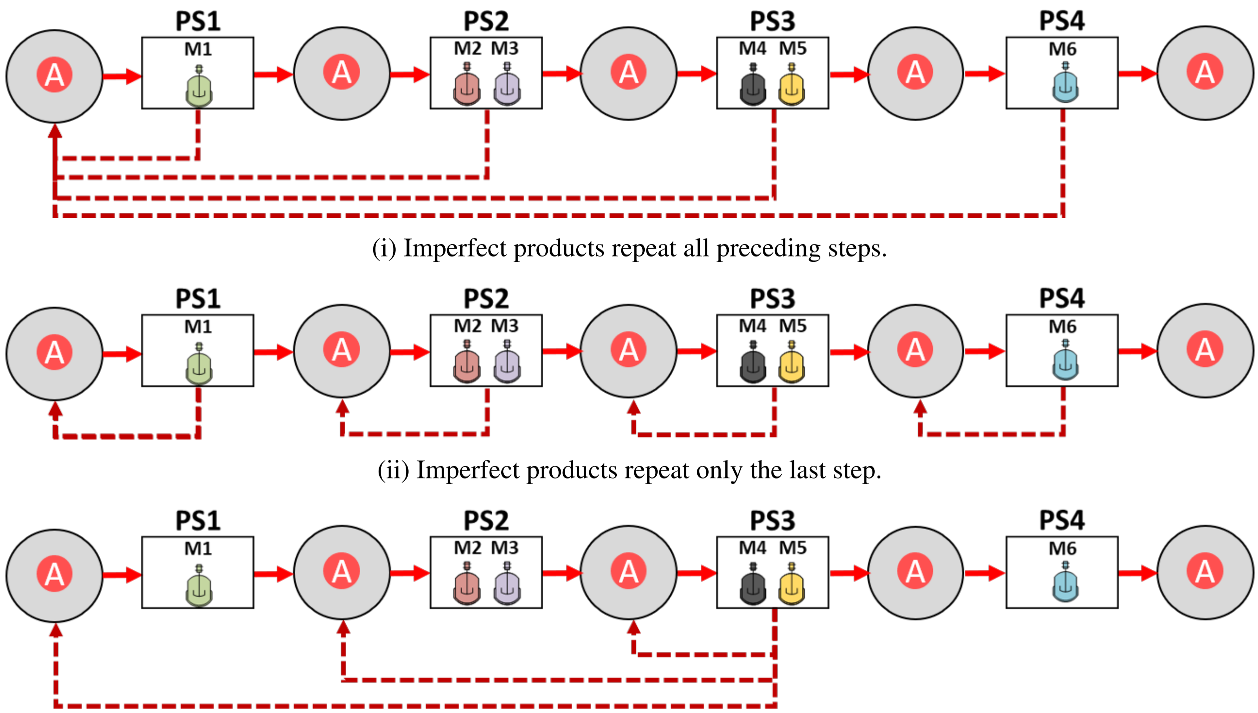
3.1 Modeling of Reprocessing Rates as Production Yields

In sequential environments, yields are not generally defined, since each task can only draw material from the task directly preceding in the recipe and can only discard the processed material to the subsequent task. On the other hand, network environments allow for multiple material inputs and outputs for each production task, necessitating the introduction of consumption and production yields, which is one of the main modeling capabilities enabled by the STN representation. In this work, we take advantage of this feature by introducing recycles that are directed to any preceding task of the production network, as appropriate, and by defining associated production yields to represent the fractions of total product output of each order that needs to be reprocessed through each given recycle path due to task failure.[2] This framework generalizes the standard, "perfect task" environment, which can be recovered by simply setting equal to one the production yields corresponding to the main recipe path.

Depending on the application, one or more recycle paths can be defined for nonconforming products, with Figure 2 illustrating the most common cases. Recycle type (i) refers to the case where the imperfect products at a stage need to undergo all of the processing steps from the very beginning. Since the feeding state of the first processing step is the recipient of all recycled material, this type of recycle is the one that can cause the most severe bottlenecks, due to the accumulation of material to be reprocessed at the early production stages. Recycle type (ii) refers to the case where the imperfect output of a task is directed to the input of that same task, repeating only the latest processing task where presumably the failure occurred. Finally, recycle type (iii) illustrates the case where non-conforming products exhibit a variety of defects and have to be routed through different reprocessing paths. This would be relevant in settings where detection of the failure is done only at specific points in the network, and the path through which to reroute the material shall depend on the type of failure detected and the appropriate remedy to be followed in each case. We remark that a mix of these common recycle paths may be present in certain real life applications, and this can be achieved easily by defining the applicable STN for each case. Furthermore, we highlight that, since in our model the yields are also indexed over the set of orders, we maintain the flexibility to assign different recycle paths for batches of material belonging to different orders, as applicable.

Since failure rates typically fluctuate and are unknown to the decision maker during the scheduling

[2] Note how this calls for augmenting sets $\mathcal{S}^{p}_{i,k}$ in the deterministic model.



(i) Imperfect products repeat all preceding steps.

(ii) Imperfect products repeat only the last step.

(iii) Imperfect products are rerouted to different preceding steps, depending on the nature of each product's imperfection (recycles shown here only for step PS3 failures).

Figure 2: Examples of common recycle paths to reprocess imperfect products.

phase, we assume uncertain production yields for each task that is potentially subject to a non-zero failure rate. For notational convenience, we use $\mathcal{I}^{imperf} \subseteq \mathcal{I}$ to denote the subset of tasks that may result into imperfections. We also denote as $s^0_{i,k} \in \mathcal{S}^p_{i,k}$ the sub-state that corresponds to the main recipe path and with $\mathcal{S}^{rec}_{i,k} \equiv \mathcal{S}^p_{i,k} \setminus \{s^0_{i,k}\}$ the sub-states that can receive recycled material.

Furthermore, in order to properly model potential variations in the production yields through time and across different executions of the same type of task, we apply the stage-splitting technique that was first presented in the work of Lappas and Gounaris [2016], where separate copies–one for each stage (in this case, event point)–are considered in the model for every uncertain parameter. More specifically, each production yield parameter $\rho^p_{i,k,s}$ will be augmented with index n, giving rise to a set of parameters $\rho^p_{i,k,s,n}$, for each $n \in \mathcal{N}$. Figure 3 provides a depiction of how these parameters relate to a given processing task.

3.2 Uncertainty Characterization

In order to employ an Adjustable Robust Optimization framework to address uncertainty in our context, we first need to define an appropriate uncertainty set. This set should attempt to avoid any unnecessary con-

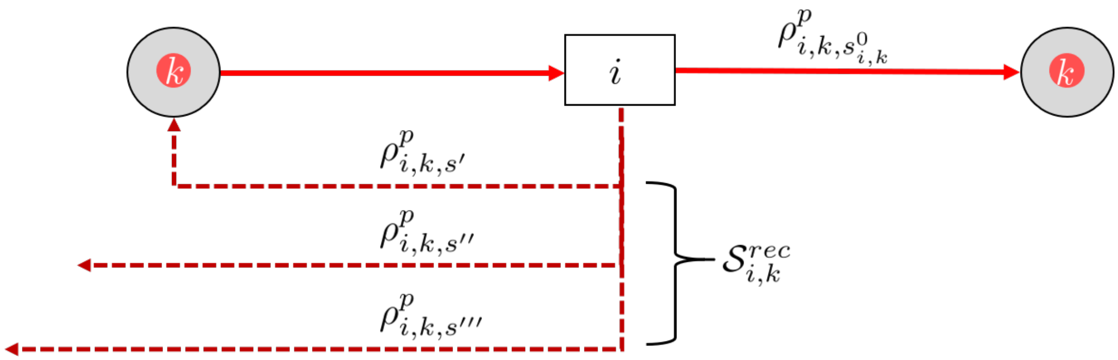


Figure 3: Output streams of an imperfect task i and associated production yields for a specific order k and event point n.

servatism, only reflecting the risk that the decision-maker is realistically exposed to. As pointed by Lappas and Gounaris [2016, 2018a], in a scheduling context this entails taking into account the endogenous nature of the uncertainty, whereby an uncertain production yield will be revealed (and will be of relevance) only if the optimal solution activates the corresponding task. To that end, we shall utilize decision-dependent uncertainty sets [Lappas and Gounaris, 2018b] where the uncertain parameters are to be restricted according to "materialization indicators." More specifically, for each task $i \in \mathcal{I}^{imperf}$ and for each event point $n \in \mathcal{N}$, we define a materialization indicator binary variable $v_{i,n} \in \{0, 1\}$ that is associated with every uncertain production yield parameter $\rho^{p}_{i,k,s,n}$ (for all $k \in \mathcal{K}_i$ and $s \in \mathcal{S}^{rec}_{i,k}$). This binary variable must be appropriately linked to the model's native decision variables to indicate whether the relevant processing task was indeed finalized–and hence exerted a realization of the production yields–at the given time point; therefore,

$$v_{i,n} := \sum_{n' \in \mathcal{N}_n^-} W_{i,n',n} \qquad \forall\, n \in \mathcal{N},\ \forall\, i \in \mathcal{I}^{imperf}. \tag{19}$$

Having defined these materialization indicators, we now propose a decision-dependent uncertainty set $\mathcal{P}\,(v_{i,n})$ as shown in Eq. 20. This set reflects restrictions on the magnitude of the possible failures, such as upper bounds on the fraction of any given order-specific output that is directed towards any recycle stream, as well as the "yield balance" restriction that ensures the totality of the task's production is directed towards some output state. The set also includes correlations among the parameters corresponding to all orders concurrently processed in a given task to reflect the fact that it is highly unlikely all the orders will fail at their maximum allowable rates at the same time. Note how the various restrictions on the uncertain parameters are only applied if the corresponding materialization indicators attain the value of one, i.e., only if the referenced parameters remain relevant for the optimal solution. The constants $\xi \in [0,1]$ and $\phi \in [0,1]$ in Eq. 20 are selected by the modeler to parameterize the size and shape, respectively, of the polyhedral uncertainty set.[3]

$$\mathcal{P}(v_{i,n}) := \left\{ \begin{array}{l} \rho^{p}_{i,k,s,n} \in \mathbb{R}_{+} : \\ \quad \{v_{i,n} = 1\} \Rightarrow \left\{ \begin{array}{ll} \rho^{p}_{i,k,s,n} \le \xi & \forall\, s \in \mathcal{S}^{rec}_{ik},\ \forall\, k \in \mathcal{K}_i \\ \sum\limits_{s \in \mathcal{S}^{p}_{ik}} \rho^{p}_{i,k,s,n} = 1 & \forall\, k \in \mathcal{K}_i \\ \sum\limits_{k \in \mathcal{K}_i} \sum\limits_{s \in \mathcal{S}^{rec}_{ik}} \rho^{p}_{i,k,s,n} \le \phi\,\xi & \end{array} \right\} \quad \forall\, n \in \mathcal{N},\ \forall\, i \in \mathcal{I}^{imperf} \end{array} \right\} \tag{20}$$

Furthermore, since the total number of uncertain parameters referenced in the uncertainty set is expected to impact the overall computational tractability of the robust counterpart, it is in our interest to keep this number as low as possible. To that end, we can take advantage of the mass conservation among the outputs and equivalently reformulate our model of uncertainty by eliminating one of the uncertain production yields for each task and event point from the uncertainty set. More specifically, we can eliminate the yields towards the main recipe path[4] by applying their definition,

$$\rho^{p}_{i,k,s^{0}_{i,k},n} := 1 - \sum_{s \in \mathcal{S}^{rec}_{i,k}} \rho^{p}_{i,k,s,n} \quad \forall\, n \in \mathcal{N},\ \forall\, k \in \mathcal{K}_i,\ \forall\, i \in \mathcal{I}^{imperf}, \tag{21}$$

while also performing similar parameter eliminations from the optimization model (specifically, from Eqs. 7) as well.

[3] We remark that these constants could in principle depend on the specific task and/or event point, but for ease of exposition we will consider them in the remainder of this work as common across all (i, n) combinations.

[4] In this context, the production yield towards the main recipe path can be viewed as the task's success rate.

The resulting uncertainty set we thus use is as follows:

$$\mathcal{P}(v_{i,n}) := \left\{ \begin{array}{l} \rho^{p}_{i,k,s,n} \in \mathbb{R}_{+} : \\ \qquad \{v_{i,n} = 1\} \Rightarrow \left\{ \begin{array}{ll} \rho^{p}_{i,k,s,n} \le \xi & \forall\, s \in \mathcal{S}^{rec}_{ik},\ \forall\, k \in \mathcal{K}_i \\ \sum\limits_{s \in \mathcal{S}^{rec}_{ik}} \rho^{p}_{i,k,s,n} \le 1 & \forall\, k \in \mathcal{K}_i : \left\{ |\mathcal{S}^{rec}_{ik}| \neq 0 \right\} \\ \sum\limits_{k \in \mathcal{K}_i} \sum\limits_{s \in \mathcal{S}^{rec}_{ik}} \rho^{p}_{i,k,s,n} \le \phi\, \xi & \end{array} \right\} \quad \forall\, n \in \mathcal{N},\ \forall\, i \in \mathcal{I}^{imperf} \end{array} \right\}. \tag{22}$$

3.3 Adjustable Robust Counterpart

We will now discuss the reformulation steps that are required to obtain an adjustable robust counterpart of the deterministic model from Section 2.2. The first step is to select which of the decision variables from the original model will be considered as *adjustable* variables. In our case, since the uncertain parameters (production yields) participate in the material balance constraints (Eqs. 7), it is the order-specific sub-state levels ($SO_{k,s,n}$) that have to be defined as adjustable variables, in order to allow for the satisfaction of the material balances for all uncertainty scenarios admitted by the uncertainty set. To that end, and as per the standard ARO procedure, the modeler a-priori postulates the functional form of the dependency of these decision variables on the realizations of the uncertain parameters. In our case, we restrict this dependency to the set of affine functions (i.e., we apply *affine decision rules* [Ben-Tal et al., 2004]), due to their proven performance in the context of scheduling ARO models [Lappas and Gounaris, 2016, 2018a].

More specifically, for each original variable $SO_{k,s,n}$ in our optimization model, we define a new set of unrestricted continuous variables $SO^{0}_{k,s,n} \in \mathbb{R}$ to encode the constant part (adjustment offset) of the affine decision rule and a set of unrestricted continuous variables $SO^{1}_{k,s,n,i',k',s',n'} \in \mathbb{R}$, one for each uncertain parameter $\rho^{p}_{i',k',s',n'}$ upon which we wish to adjust, to encode the linear part (adjustment coefficients). Given these new variable declarations, we perform the following replacements in the optimization model to eliminate the original $SO_{k,s,n}$ variables:

$$SO_{k,s,n} \longleftarrow \left(SO^{0}_{k,s,n} + \sum_{i' \in \mathcal{I}^{imperf}} \sum_{k' \in \mathcal{K}'_{i}} \sum_{s' \in \mathcal{S}^{rec}_{i'k'}} \sum_{n' \in \mathcal{N}} SO^{1}_{k,s,n,i',k',s',n'}\, \rho^{p}_{i',k',s',n'} \right) \quad \forall\, n \in \mathcal{N},\ \forall\, s \in \mathcal{S}_k,\ \forall\, k \in \mathcal{K}. \tag{23}$$

Special care is required to restrict the admissible values of the variables $SO^1_{k,s,n,i',k',s',n'}$ so as to ensure that the model complies with the principle of *non-anticipativity*. More specifically, we need to ensure that adjustment of an original variable $SO_{k,s,n}$ does not rely on a parameter $\rho^p_{i',k',s',n'}$ whose value has not yet been revealed by the time the decision-maker needs to evaluate the material balance (essentially, time T_n). Following the procedure previously demonstrated in Lappas and Gounaris [2016], we further add the below non-anticipativity constraints in our robust counterpart.

$$SO^1_{k,s,n,i',k',s',n'} = 0 \quad \forall\, n' \in \mathcal{N} : \{n' > n\},\ \forall\, s' \in \mathcal{S}^{rec}_{i'k'},\ \forall\, k' \in \mathcal{K}_{i'},\ \forall\, i' \in \mathcal{I}^{imperf},\ \forall\, n \in \mathcal{N},\ \forall\, s \in \mathcal{S}_k,\ \forall\, k \in \mathcal{K} \tag{24}$$

$$\{v_{i',n'} = 0\} \Rightarrow \left\{SO^1_{k,s,n,i',k',s',n'} = 0\right\} \quad \forall\, n' \in \mathcal{N} : \{n' \leq n\},\ \forall\, s' \in \mathcal{S}^{rec}_{i'k'},\ \forall\, k' \in \mathcal{K}_{i'},\ \forall\, i' \in \mathcal{I}^{imperf},\ \forall\, n \in \mathcal{N},\ \forall\, s \in \mathcal{S}_k,\ \forall\, k \in \mathcal{K} \tag{25}$$

Eqs. 24 correspond to variable fixings that are imposed to ensure that a decision variable related to event point n shall not be adjusted on information that is to be revealed at a later event point $n' > n$.[5] On the other hand, Eqs. 25 correspond to conditional variable fixings that are imposed to ensure that no variable adjustments will be done based on realizations of *non-materialized* uncertain parameters. These implication constraints can be implemented via the *indicator constraint* facility of modern solvers.

After the affine decision rule replacements (Eqs. 23), uncertain parameters $\rho^p_{i,k,s,n}$ are referenced in Eqs. 2, 9, 16, and 7. For the first three sets of constraints, which constitute inequalities, duality-based reformulations are applied. For the sake of brevity, we omit the details, but interested readers are referred to the work by Lappas and Gounaris [2018b] that discusses how such a reformulation is obtained in the context of a decision-dependent uncertainty set of the form of Eq. 22. With regards to Eqs. 7, which constitute equality constraints, these are reformulated via the coefficient matching approach, as previously shown by Gorissen et al. [2015] and Lappas and Gounaris [2018a].

We remark that, due to the fact that the uncertainty set (Eq. 22) is polyhedral, the adjustable robust model retains the mixed-integer linear structure of its deterministic counterpart. Moreover, we note that the indicator variables $v_{i,n}$ need not be declared as separate variables in the model, rather than be directly

[5] Obviously, such variable fixings can be simply implemented by not including these variables in the summations of Eqs. 23, alleviating the need to clutter the model with additional variables.

replaced by their definitions (Eq. 19) that reference the model's native binary variables $W_{i,n',n}$.

4 Computational Studies

We assess the performance of the proposed framework on the set of 9 benchmark problems utilized in the work of Lagzi et al. [2017a], which constitute representative process configurations of an actual facility in the scientific services sector. In order to account for the possibility of reprocessing, we augmented the original deterministic multi-tasking instances of Lagzi et al. [2017a] with appropriately defined recycles, as illustrated in Appendix B. The detailed input data for all instances can be found in Appendix C. For each input network, we considered 15 different levels of uncertainty stemming from combinations of values for ξ and ϕ, as follows: $\xi \in \{0.10, 0.20, 0.30\}$ and $\phi \in \{0.00, 0.25, 0.50, 0.75, 1.00\}$. For completeness, we also considered the combination $(\xi, \phi) = (0.00, 0.00)$, which corresponds to the original deterministic case under a nominal realization where there is no possibility of task failure, and hence, no reprocessing. The instances were solved for both the objectives of profit maximization and makespan minimization, resulting in a total of 288 optimization runs. Our framework was implemented within CPLEX Optimization Studio 12.8, and the runs were conducted using 4 parallel threads (limited via software option) in an Intel Xeon CPU E5-2689v4@3.10GHz with 4 GB of available RAM. For the full details of our computational results, the readers are referred to Appendix D.

4.1 Representative Solution Analysis

In order to gain insight on the solutions obtained from our proposed framework, let us initially focus on the motivating example P1 from Lagzi et al. [2017a]. As shown in Appendix B, this instance considers two orders. Whereas the red order has to undergo all tasks, the green order's recipe calls for bypassing the second processing unit. It further assumes that all of the processing tasks exhibit the potential to produce non-compliant products, which have to be reprocessed by repeating only the last step in which the failure occurred. The risk in this instance arises from the fact that the extra work load, which is introduced to the system by the reprocessing tasks, may lead to cases where the capacity of the machines is not enough to accommodate the extra material in a single batch, thus making the operation increasingly unprofitable or overly delayed. In Figure 4, we plot the optimal schedules for the objective of maximizing profit, considering the deterministic instance as well as three different robust instances, namely the (ξ, ϕ) settings of $(0.10, 0.00)$, (0.20, 0.50) and (0.30, 1.00), which are reported here as representatives of low, medium and high levels of uncertainty, respectively.

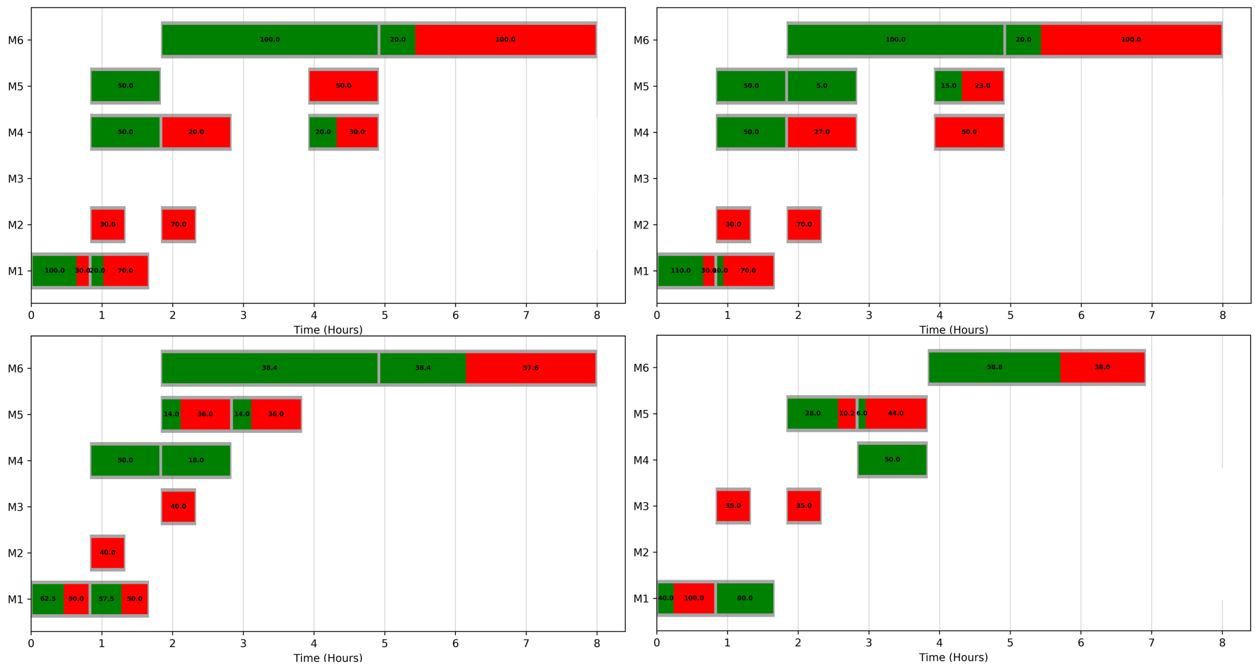


Figure 4: Optimal schedules for problem P1 with maximization of profit objective for various levels of uncertainty; *top left*: deterministic solution (obj. = 1,640); *top right*: low uncertainty (w.-c. obj. = 1,640); *bottom left*: medium uncertainty (w.-c. obj. = 852.5); *bottom right*: high uncertainty (w.-c. obj. = 480.2). All robust schedules are plotted for the nominal realizations of the uncertain parameters. The numbers within the bars correspond to the batch sizes of each order.

A close examination of the solutions depicted in Figure 4 leads to interesting findings. More specifically, as we account for uncertainty in the success rates, the worst-case objective is reduced from 1,640 to 852.5 and 480.2 for medium and high level of uncertainty, respectively. This expected behavior reflects the risk premium to be paid in order to guarantee the feasibility of the solution for the worst case scenarios. Another interesting observation is that, for low level of uncertainty, the solution has the same worst-case objective value as the deterministic one, which hints to the fact that the system has enough capacity to accommodate some small amount of material reprocessing. On the other hand, when we attempt to consider medium and high levels of uncertainty, we can see that the robust solutions utilize only a portion of the initial orders, in order to ensure that the processing machines will not be overloaded by the reprocessing tasks that may have to be accommodated. Although this imposes a tax on the profits, it guarantees the feasibility of the provided solution for a wider range of scenarios, and the decision maker might in fact want to adopt these more conservative robust solutions instead.

Interesting observations can also be made for the solutions from Figure 5, which depicts optimal schedules for the objective of makespan minimization under various levels of uncertainty. In this case, the worst-case makespan of 5.4h does not increase until only after a high level of uncertainty is postulated, at which point it increases to 5.8h. Arguably, there is no reason for the decision maker to pick the deterministic solution over the robust solution corresponding to medium level of uncertainty, which offers the same objective but with the extra benefit of insuring against some considerable level of risk.

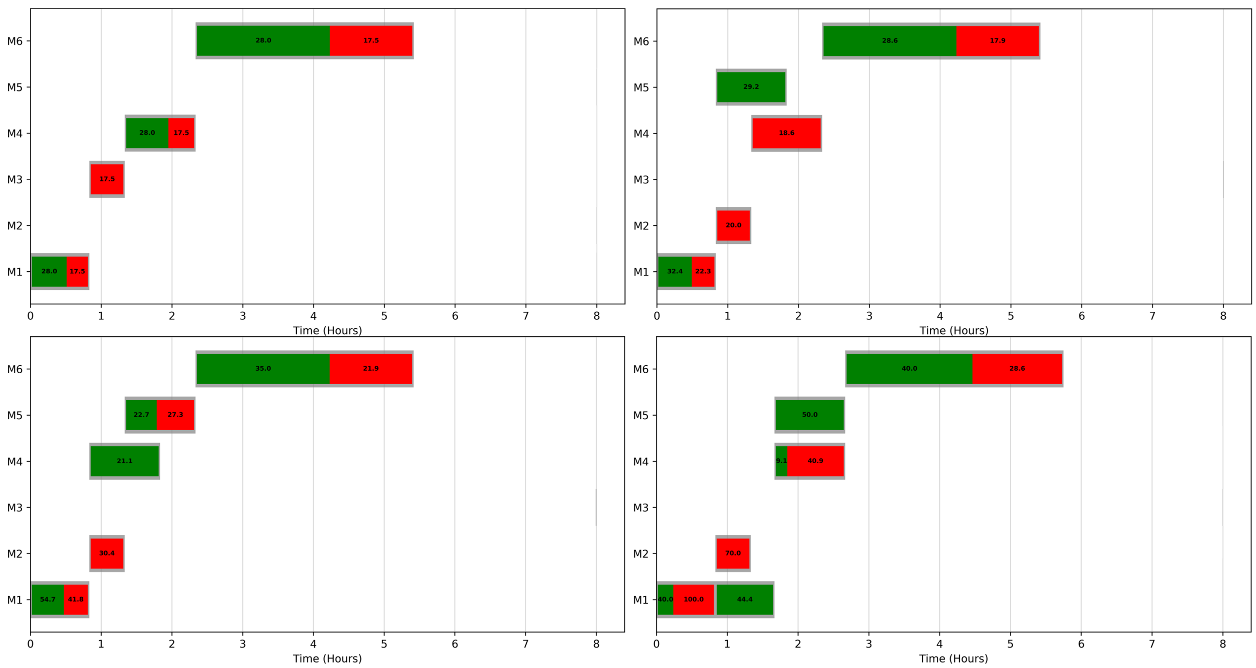


Figure 5: Optimal schedules for problem P1 with minimization of makespan objective for various levels of uncertainty; *top left*: deterministic solution (obj. = 5.4h); *top right*: low uncertainty (w.-c. obj. = 5.4h); *bottom left*: medium uncertainty (w.-c. obj. = 5.4h); *bottom right*: high uncertainty (w.-c. obj. = 5.8h). All robust schedules are plotted for the nominal realizations of the uncertain parameters. The numbers within the bars correspond to the batch sizes of each order.

4.2 Price of Robustness

In this section, we obtain an estimate of the price of robustness that one is expected to pay in this context so as to insure the solutions against the risk of imperfect tasks. To that end, Tables 1 and 2 show the worst-case optimal profit and makespan, respectively, averaged across all problem instances and normalized to the corresponding deterministic optimal objective values (= 1.000), for each considered values of ξ and ϕ.

Table 1: Average normalized worst-case profit for various levels of uncertainty.

$\xi \setminus \phi$	0.00	0.25	0.50	0.75	1.00
0.10	0.999	0.965	0.944	0.934	0.930
0.20	0.999	0.934	0.894	0.876	0.867
0.30	0.999	0.908	0.856	0.825	0.806

Table 2: Average normalized worst-case makespan for various levels of uncertainty.

$\xi \setminus \phi$	0.00	0.25	0.50	0.75	1.00
0.10	1.000	1.000	1.000	1.000	1.000
0.20	1.000	1.000	1.000	1.000	1.000
0.30	1.000	1.005	1.013	1.013	1.013

For the objective of profit maximization, it can be seen from Table 1 that, as the uncertainty set size increases (reflected by increasing parameter ξ), the worst-case profit decreases. On the contrary, as the correlations among success rates referring to the same processing units become stronger (reflected by decreasing parameter ϕ), the worst-case profit increases. It can be concluded that, when strong correlations among the uncertain parameters are justified, we can expect to obtain robust solutions with objective value similar or comparable to that of the deterministic solution, a fact that we first observed in the motivating example. Even in the most conservative case of large uncertainty set size and no correlations among the uncertain parameters, $(\xi, \phi) = (0.30, 1.00)$, a risk premium of only 19.4% needs to be paid.

In contrast, for the objective of makespan minimization, it can be seen from Table 2 that uncertainty in the success rates has minimal impact. This can be attributed to the fact that, as per the original data from Lagzi et al. [2017a], the processing times do not generally depend on overall batch sizes; that is, $\beta_i = 0$ in most instances. As a result, the larger batch sizes that are necessary when recycled material is being processed has no effect on the total time to process the batches. Furthermore, the capacity limits B_i^{max} were generally large enough to accommodate the recycled material in the nominally scheduled tasks, and hence there was little–if any–need to schedule additional tasks that could have pushed the worst-case makespan to larger values.

4.3 Formulation Sizes and Overall Numerical Tractability

Regarding how the size of the robust counterpart formulation scales with the size of the problem input, the pattern is similar to the one described in previous ARO works on scheduling by Lappas and Gounaris [2016, 2018a]. Tables 3 and 4 illustrate statistics about the sizes of the deterministic and adjustable robust counterpart formulations used in this study. It can be observed that, as one moves from the deterministic formulation to the robust one, there is a significant increase in the number of continuous variables and constraints that occur due to the robustification procedure based on the affine decision rules approach. This is not surprising, especially if one factors in the fact that we are considering robust optimization instances with a relatively large number of parameters (e.g., more than 100, in the case of P7). On the other hand, we should underline the fact that, for the same number of event points, there is no increase in the number of binary variables between the deterministic and the robust models, which is a significant contributor to our ability to solve the robust formulations. We remark that the sizes of the adjustable robust counterparts do not depend on the choices of (ξ, ϕ) constants for the uncertainty set.

Table 3: Deterministic Formulation Sizes.

	Problem	# Ev. Points	# Bin. Var.	# Cont. Var.	# Constraints
Max Profit	P1	6	72	288	343
	P2	7	60	267	289
	P3	8	90	453	466
	P4	8	144	672	673
	P5	6	84	431	438
	P6	10	168	1,027	878
	P7	8	180	2,142	1,251
	P8	9	210	2,180	1,665
	P9	8	198	2,647	1,912
Min Makespan	P1	6	72	288	353
	P2	8	72	316	349
	P3	8	90	453	479
	P4	6	96	460	480
	P5	7	105	528	550
	P6	10	168	1,027	899
	P7	8	180	2,142	1,311
	P8	9	210	2,180	1,727
	P9	7	165	2,236	1,731

Table 4: Adjustable Robust Formulation Sizes.

	Problem	# Ev. Points	# Uncert. Param.	# Bin. Var.	# Cont. Var.	# Constraints
Max Profit	P1	6	60	72	105,380	57,883
	P2	7	84	60	60,263	75,581
	P3	8	40	90	88,366	67,586
	P4	8	64	144	194,660	107,921
	P5	6	42	84	73,067	68,988
	P6	10	80	168	215,189	271,198
	P7	8	104	180	1,646,630	802,859
	P8	9	63	210	601,724	579,528
	P9	8	88	198	1,492,337	980,576
Min Makespan	P1	6	60	72	113,048	61,673
	P2	8	96	72	82,556	103,389
	P3	8	40	90	93,262	70,959
	P4	6	48	96	118,468	64,788
	P5	7	49	105	105,427	99,390
	P6	10	80	168	224,869	282,419
	P7	8	104	180	1,745,632	845,871
	P8	9	63	210	632,834	606,491
	P9	7	77	165	1,222,264	794,691

The noticeable increase in the size of the robust formulations, as compared to their deterministic counterparts, has a somewhat adverse (yet not prohibitive) effect on the CPU times required to solve these instances to guaranteed optimality. Tables 5 and 6 summarize the results by showing the geometric average computational times across all instances for different uncertainty levels and for each of the two objectives. We can observe that, while deterministic solutions can be obtained almost instantaneously, robust solutions require more time, in the order of one to three minutes, on average.

Table 5: Geometric average CPU times (in s) for the objective of profit maximization.

$\xi \setminus \phi$	0.00	0.25	0.50	0.75	1.00
0.00	0.2	0.2	0.2	0.2	0.2
0.10	118.9	90.4	83.8	71.6	25.1
0.20	138.9	96.4	99.8	73.9	13.1
0.30	137.4	117.5	103.2	75.9	10.6

Table 6: Geometric average CPU times (in s) for the objective of makespan minimization.

$\xi \setminus \phi$	0.00	0.25	0.50	0.75	1.00
0.00	0.2	0.2	0.2	0.2	0.2
0.10	44.7	53.3	50.9	61.5	36.8
0.20	51.6	53.4	55.5	63.6	36.9
0.30	45.6	84.3	64.4	71.1	51.1

5 Conclusions

We introduced a novel framework for the mitigation of risk that stems from the possibility of task failure and the need to reprocess material in multi-tasking scheduling environments. We presented a systematic way of mapping sequential multi-tasking scheduling instances to State-Task Network representations that allowed us to model failure rates of tasks as uncertainty in model parameters representing production yields. In this regard, our approach is not strictly bound to a specific scheduling formulation and can thus be easily utilized with various formulations as its basis. In this work, we used a specific global event-based model to showcase our approach and we combined it with the Adjustable Robust Optimization framework for scheduling under uncertainty by Lappas and Gounaris [2016] to obtain solutions that remain robust against various task failures scenarios that could be encountered in practice. Our computational studies highlighted the favorable computational tractability characteristics of the proposed approach, as the well as the quality of the robust solutions for various levels of uncertainty.

Acknowledgments

C.E.G. and N.H.L. gratefully acknowledge support from the National Science Foundation (grant No. CBET-1510787). N.H.L. further acknowledges support from the University of Patras via an Andreas Mentzelopoulos scholarship. L.R. and R.F. gratefully acknowledge the support provided by the Natural Sciences and Engineering Research Council of Canada. The authors would also like to acknowledge the support provided by a collaborating company in the scientific services sector.

References

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

J. Kallrath. Planning and scheduling in the process industry. *OR Spectrum*, 24(3):219–250, 2002. doi: 10.1007/s00291-002-0101-7.

C.A. Floudas and X. Lin. Mixed integer linear programming in process scheduling: modeling, algorithms, and applications. *Annals of Operations Research*, 139(1):131–162, 2005. doi: 10.1007/s10479-005-3446-x.

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*, 30(6-7):913–946, 2006. doi: 10.1016/j.compchemeng.2006.02.008.

C.T. Maravelias. General framework and modeling approach classification for chemical production scheduling. *AIChE Journal*, 58 (6):1812–1828, 2012. doi: 10.1002/aic.13801.

A. Sundaramoorthy and C.T. Maravelias. A general framework for process scheduling. *AIChE Journal*, 57(3):695–710, 2011. doi: 10.1002/aic.12300.

B.P. Patil, R. Fukasawa, and L.A. Ricardez-Sandoval. Scheduling of operations in a large-scale scientific services facility via multicommodity flow and an optimization-based algorithm. *Industrial & Engineering Chemistry Research*, 54(5):1628–1639, 2015. doi: 10.1021/ie503660f.

S. Lagzi, R. Fukasawa, and L.A. Ricardez-Sandoval. A multitasking continuous time formulation for short-term scheduling of operations in multipurpose plants. *Computers & Chemical Engineering*, 97:135–146, 2017a. doi: 10.1016/j.compchemeng.2016.11.012.

S. Lagzi, D.Y. Lee, R. Fukasawa, and L. Ricardez-Sandoval. A computational study of continuous and discrete time formulations for a class of short-term scheduling problems for multipurpose plants. *Industrial & Engineering Chemistry Research*, 56(31): 8940–8953, 2017b. doi: 10.1021/acs.iecr.7b01718.

Z. Li and M.G. Ierapetritou. Process scheduling under uncertainty: Review and challenges. *Computers & Chemical Engineering*, 32(4-5):715–727, 2008a. doi: 10.1016/j.compchemeng.2007.03.001.

P.M. Verderame, Josephine A. Elia, Jie Li, and C.A. Floudas. Planning and scheduling under uncertainty: a review across multiple sectors. *Industrial & Engineering Chemistry Research*, 49(9):3993–4017, 2010. doi: 10.1021/ie902009k.

I.E. Grossmann, R.M. Apap, B.A. Calfa, P. García-Herreros, and Q. Zhang. Recent advances in mathematical programming techniques for the optimization of process systems under uncertainty. *Computers & Chemical Engineering*, 91:3–14, 2016. doi: 10.1016/j.compchemeng.2016.03.002.

C.A. Mendez and J. Cerdá. An MILP framework for batch reactive scheduling with limited discrete resources. *Computers & Chemical Engineering*, 28(6-7):1059–1068, 2004. doi: 10.1016/j.compchemeng.2003.09.008.

D. Gupta, C.T. Maravelias, and J.M. Wassick. From rescheduling to online scheduling. *Chemical Engineering Research and Design*, 116:83–97, 2016.

M.G. Ierapetritou and E.N. Pistikopoulos. Batch plant design and operations under uncertainty. *Industrial & Engineering Chemistry Research*, 35(3):772–787, 1996. doi: 10.1021/ie950263f.

I.E. Grossmann and J. Balasubramanian. Approximation to multistage stochastic optimization in multiperiod batch plant scheduling under demand uncertainty. *Industrial & Engineering Chemistry Research*, 43(14):3695–3713, 2004. doi: 10.1021/ie030308+.

J. Ryu, V. Dua, and E.N. Pistikopoulos. Proactive scheduling under uncertainty: a parametric optimization approach. *Industrial & Engineering Chemistry Research*, 46(24):8044–8049, 2007. doi: 10.1021/ie070018j.

Z. Li and M.G. Ierapetritou. Reactive scheduling using parametric programming. *AIChE Journal*, 54(10):2610–2623, 2008b. doi: 10.1002/aic.11593.

X. Lin, S.L. Janak, and C.A. Floudas. A new robust optimization approach for scheduling under uncertainty: I. Bounded uncertainty. *Computers & Chemical Engineering*, 28(6-7):1069–1085, 2004. doi: 10.1016/j.compchemeng.2003.09.020.

Z. Li and M.G. Ierapetritou. Robust optimization for process scheduling under uncertainty. *Industrial & Engineering Chemistry Research*, 47(12):4148–4157, 2008c. doi: 10.1021/ie071431u. URL `http://pubs.acs.org/doi/abs/10.1021/ie071431u`.

Y.A. Guzman, L.R. Matthews, and C.A. Floudas. New a priori and a posteriori probabilistic bounds for robust counterpart optimization: I. Unknown probability distributions. *Computers & Chemical Engineering*, 84:568–598, 2016. doi: 10.1016/j.compchemeng.2015.09.014.

B.L. Gorissen, İ. Yanıkoğlu, and D. den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, 2015. doi: 10.1016/j.omega.2014.12.006.

A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004. doi: 10.1007/s10107-003-0454-y.

N.H. Lappas and C.E. Gounaris. Multi-stage adjustable robust optimization for process scheduling under uncertainty. *AIChE Journal*, 62(5):1646–1667, 2016. doi: 10.1002/aic.15183.

N.H. Lappas and C.E. Gounaris. Theoretical and computational comparison of continuous-time process scheduling models for adjustable robust optimization. *AIChE Journal*, 64(8):3055–3070, 2018a. doi: 10.1002/aic.16124.

C.A. Méndez and J. Cerdá. An MILP continuous-time framework for short-term scheduling of multipurpose batch processes under different operation strategies. *Optimization and Engineering*, 4(1/2):7–22, 2003. doi: 10.1023/A:1021856229236.

S. Ferrer-Nadal, E. Capon-Garcia, C.A. Mendez, and L. Puigjaner. Material transfer operations in batch scheduling. A critical modeling issue. *Industrial & Engineering Chemistry Research*, 47(20):7721–7732, 2008. doi: 10.1021/ie800075u.

H. Lee and C.T. Maravelias. Discrete-time mixed-integer programming models for short-term scheduling in multipurpose environments. *Computers & Chemical Engineering*, 2017. doi: 10.1016/j.compchemeng.2017.06.013.

P.M. Castro, A.P. Barbosa-po, H.A. Matos, and A.Q. Novais. Simple continuous-time formulation for short-term scheduling of batch and continuous processes. *Industrial & Engineering Chemistry Research*, 43(1):105–118, 2004. doi: 10.1021/ie0302995.

N.H. Lappas and C.E. Gounaris. Robust optimization for decision-making under endogenous uncertainty. *Computers & Chemical Engineering*, 111:252–266, 2018b. doi: 10.1016/j.compchemeng.2018.01.006.

Appendix A Model Nomenclature

Indices

n	Event points
i	Processing tasks
j	Processing units
k	Orders
s	States

Sets

$\mathcal{N} \equiv \{1, 2, \ldots, N\}$	Event points
$\mathcal{N}_n^- \equiv \{n - \delta n, \ldots, n - 1\}$	Event points in the immediate past of event point n
$\mathcal{N}_n^+ \equiv \{n + 1, \ldots, n + \delta n\}$	Event points in the immediate future of event point n
$\mathcal{I}$	Processing tasks
$\mathcal{J}$	Processing units
$\mathcal{K}$	Orders
$\mathcal{S}$	States
$\mathcal{I}_j \subseteq \mathcal{I}$	Processing tasks compatible with unit j
$\mathcal{I}_{k,s}^p \subseteq \mathcal{I}$	Processing tasks producing state s for order k
$\mathcal{I}_{k,s}^c \subseteq \mathcal{I}$	Processing tasks consuming state s for order k
$\mathcal{K}_i \subseteq \mathcal{K}$	Orders that are to be processed by task i
$\mathcal{S}_{i,k}^p \subseteq \mathcal{S}$	States storing material produced by task i for order k
$\mathcal{S}_{i,k}^c \subseteq \mathcal{S}$	States storing material consumed by task i for order k
$\mathcal{S}_k \equiv \bigcup_{i \in \mathcal{I}} \left(\mathcal{S}_{i,k}^p \cup \mathcal{S}_{i,k}^c \right)$	States relevant to order k ("sub-states (k, s)")

Variables

z	Objective value (epigraph variable)
T_n	Time of event point n
$SO_{k,s,n}$	Level of state s of order k ("sub-state (k, s)") at event point n
$BO_{i,k,n,n'}$	Batch size of task i for order k that starts at event point n and finishes by event point n'
$G_{j,n}$	*[pseudo-binary]*: 1, if unit j is utilized at event point n; 0, otherwise
$W_{i,n,n'}$	*[binary]*: 1, if task i starts exactly at event point n and finishes by (but no later than) event point n'; 0, otherwise

Parameters

N	Total number of event points
δn	Maximum allowed event-point span for all processing tasks
α_i	Fixed processing time for task i
β_i	Batch size-dependent processing time for task i
$\rho^p_{i,k,s,n}$	Production yield of task i towards sub-state (k, s) at event point n
$\rho^c_{i,k,s,n}$	Consumption yield of task i from sub-state (k, s) at event point n
B_i^{max}	Maximum batch size (total across all orders) for task i
B_i^{min}	Minimum batch size (total across all orders) for task i
$SO_{k,s,0}$	Initial storage level for sub-state (k, s)
$D_{k,s}$	Demand for sub-state (k, s) (only for Min Makespan)
$P_{k,s}$	Price of sub-state (k, s) (only for Max Profit)
H	Horizon (only for Max Profit)

Appendix B Benchmark Instances

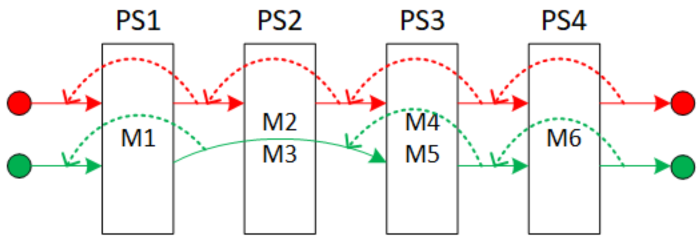


Figure B1: Operations figure for benchmark P1

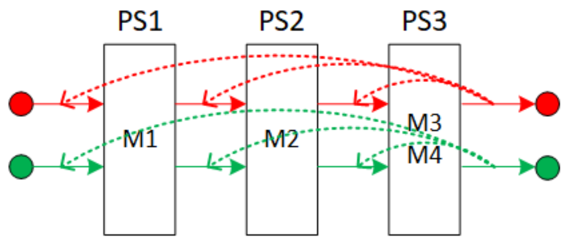


Figure B2: Operations figure for benchmark P2

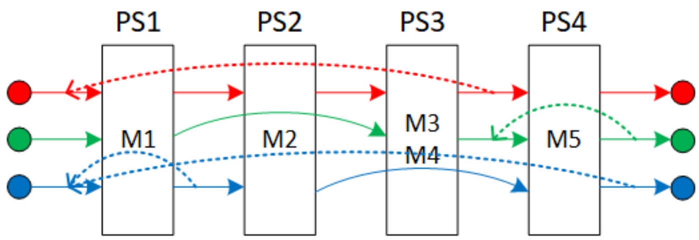


Figure B3: Operations figure for benchmark P3

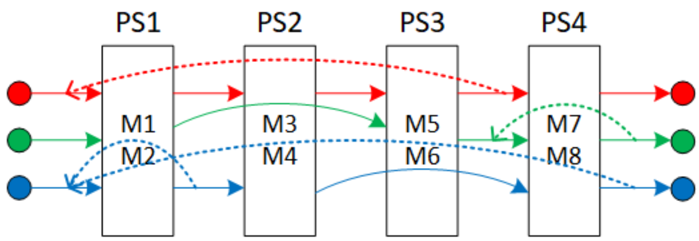


Figure B4: Operations figure for benchmark P4

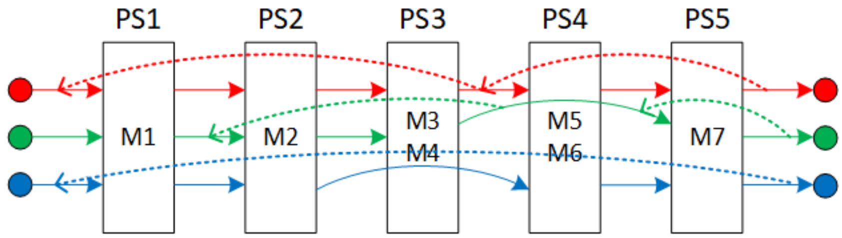


Figure B5: Operations figure for benchmark P5

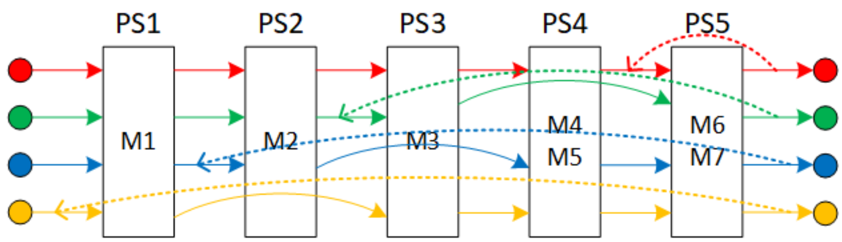


Figure B6: Operations figure for benchmark P6

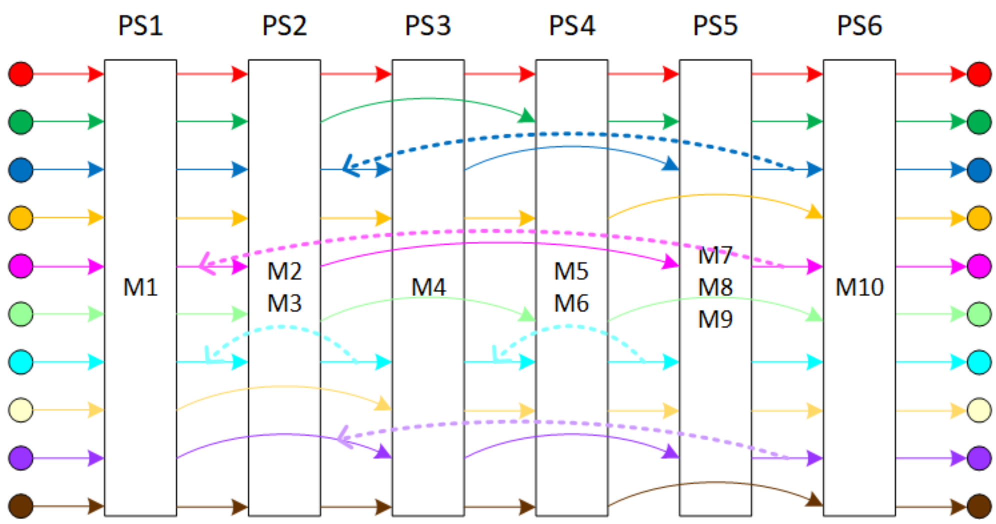


Figure B7: Operations figure for benchmark P7

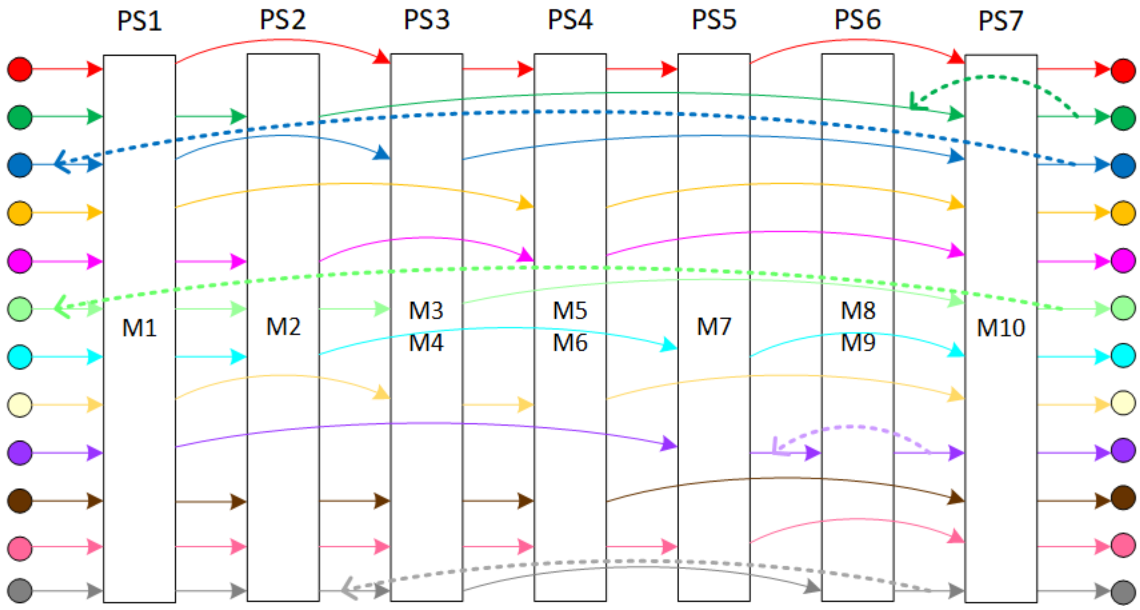


Figure B8: Operations figure for benchmark P8

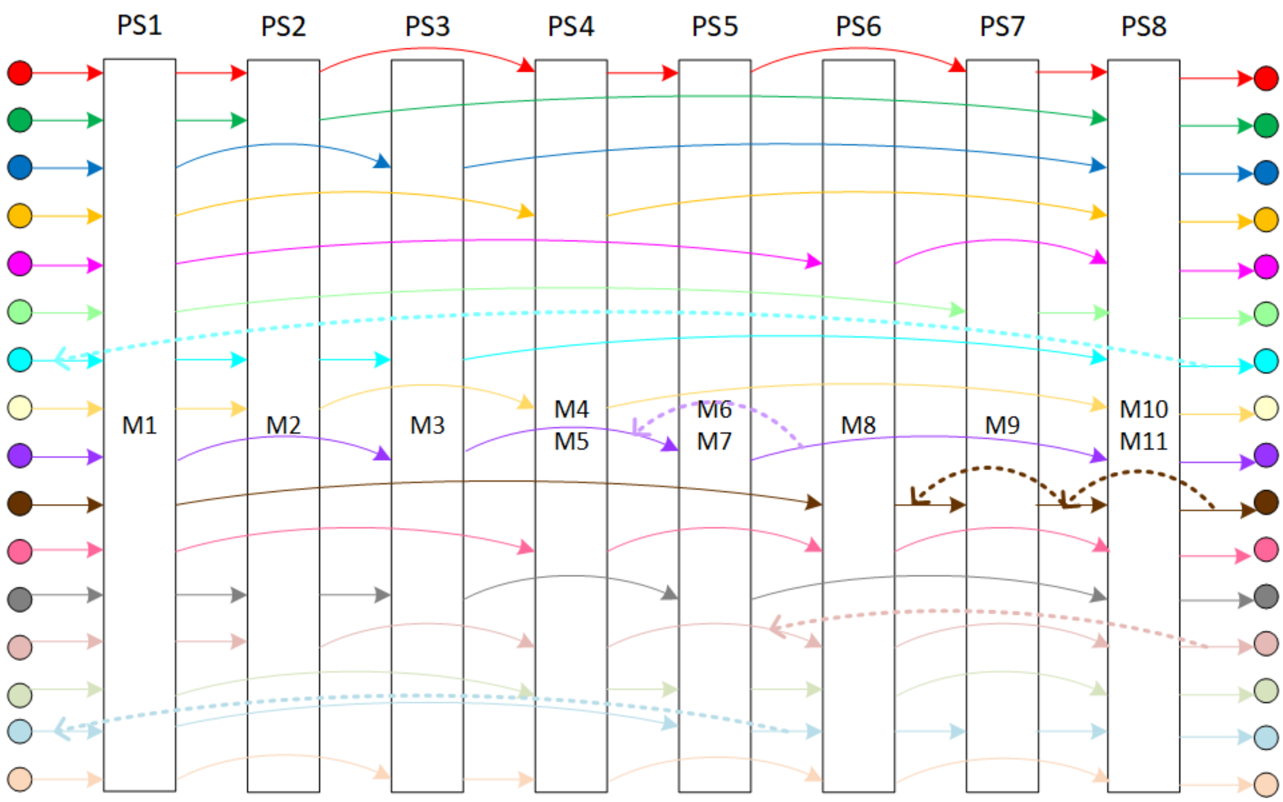


Figure B9: Operations figure for benchmark P9

Appendix C Input Data

Data related to specific machines, such as maximum batch sizes and fixed processing times, are shown below. Note that the minimum batch sizes, B_i^{min}, and batch size-dependent processing times, βi, are zero in all cases.

		B_i^{max}	α_i
P1	M1	140	0.83
	M2	70	0.50
	M3	70	0.50
	M4	50	1.00
	M5	50	1.00
	M6	120	3.08
P2	M1	150	1.83
	M2	100	2.00
	M3	80	1.33
	M4	80	1.33
P3	M1	150	0.83
	M2	80	0.58
	M3	50	0.50
	M4	50	0.50
	M5	100	1.25
P4	M1	15	0.33
	M2	15	0.33
	M3	20	0.50
	M4	20	0.50
	M5	30	0.75
	M6	30	0.75
	M7	20	0.58
	M8	20	0.58

		B_i^{max}	α_i
P5	M1	100	2.00
	M2	140	0.50
	M3	120	1.33
	M4	120	1.33
	M5	110	1.50
	M6	110	1.50
	M7	230	1.00
P6	M1	160	0.66
	M2	110	1.33
	M3	100	1.50
	M4	60	0.66
	M5	60	0.66
	M6	50	0.83
	M7	50	0.83
P7	M1	100	1.50
	M2	50	0.16
	M3	50	0.16
	M4	150	1.33
	M5	60	0.83
	M6	60	0.83
	M7	50	0.75
	M8	50	0.75
	M9	50	0.75
	M10	120	0.66

		B_i^{max}	α_i
P8	M1	200	2.00
	M2	250	1.50
	M3	75	0.75
	M4	75	0.75
	M5	90	1.00
	M6	90	1.00
	M7	200	0.83
	M8	70	0.66
	M9	70	0.66
	M10	100	0.50
P9	M1	300	2.50
	M2	270	2.00
	M3	300	0.58
	M4	130	1.25
	M5	130	1.25
	M6	105	0.75
	M7	105	0.75
	M8	210	1.08
	M9	230	1.50
	M10	95	1.00
	M11	95	1.00

Data related to specific orders, such as initial storage levels for each order's initial sub-state (initial material inputs) as well as unit prices and demands for the final sub-state (final material outputs), are shown below. Note that the orders are referenced in the same sequence as shown in the operations figures of Appendix B. Every sub-state other than those mentioned above have an initial level of zero.

		$SO_{k,\text{inp},0}$	$P_{k,\text{out}}$
P1	O1	100	8.0
	O2	120	7.0
P2	O1	200	7.0
	O2	200	7.0
P3	O1	200	7.0
	O2	200	7.0
	O3	200	8.0
P4	O1	50	7.0
	O2	50	7.0
	O3	50	8.0
P5	O1	150	8.0
	O2	150	8.0
	O3	150	9.0
P6	O1	150	8.0
	O2	150	8.0
	O3	150	8.0
	O4	150	9.0

		$SO_{k,\text{inp},0}$	$P_{k,\text{out}}$
P7	O1	50	9.0
	O2	50	9.0
	O3	50	9.0
	O4	50	8.0
	O5	50	8.0
	O6	50	10.0
	O7	50	9.0
	O8	50	8.0
	O9	50	10.0
	O10	50	9.0
P8	O1	70	7.0
	O2	70	7.0
	O3	70	7.0
	O4	70	8.0
	O5	70	8.0
	O6	70	8.0
	O7	70	8.0
	O8	70	8.0
	O9	70	9.0
	O10	70	9.0
	O11	70	9.0
	O12	70	9.0

		$SO_{k,\text{inp},0}$	$P_{k,\text{out}}$
P9	O1	80	7.0
	O2	80	7.0
	O3	80	7.0
	O4	80	7.0
	O5	80	7.0
	O6	80	8.0
	O7	80	8.0
	O8	80	8.0
	O9	80	8.0
	O10	80	8.0
	O11	80	9.0
	O12	80	9.0
	O13	80	9.0
	O14	80	9.0
	O15	80	9.0
	O16	80	10.0

Finally, for the objective of maximization of profit, the horizon H is set to 8 hours in all instances. Furthermore, since the objective of makespan minimization was not considered in the original benchmark problems, we imposed a demand for each order's final sub-state, $D_{k,\text{out}}$, as being equal to 70% of the initial storage level of the corresponding order's initial sub-state, divided by the number of processing steps required for this order to reach the final sub-state. The number of processing steps for each order can be inferred from Appendix B.

Appendix D Detailed Computational Results

Problem	Obj. Type	ξ	ϕ	# Ev. Points	# Uncert. Param.	# Bin. Var.	# Cont. Var.	# Constraints	Root Node Rlx.	Rel. Obj. Val.	Obj. Val.	Opt. Gap	# Nodes	CPU Time (s)
P1	Max Profit	0.00	0.00	6	0	72	288	343	1,640.0	1,640.0	1,640.0	0.0%	23	< 0.1
P1	Max Profit	0.10	0.00	6	60	72	105,380	57,883	1,640.0	1,640.0	1,640.0	0.0%	0	< 0.1
P1	Max Profit	0.10	0.25	6	60	72	105,380	57,883	1,501.5	1,342.2	1,342.2	0.0%	251	3.1
P1	Max Profit	0.10	0.50	6	60	72	105,380	57,883	1,371.8	1,196.5	1,196.5	0.0%	232	2.7
P1	Max Profit	0.10	0.75	6	60	72	105,380	57,883	1,250.5	1,151.8	1,151.8	0.0%	117	1.7
P1	Max Profit	0.10	1.00	6	60	72	105,380	57,883	1,137.2	1,137.2	1,137.2	0.0%	85	1.3
P1	Max Profit	0.20	0.00	6	60	72	105,380	57,883	1,640.0	1,640.0	1,640.0	0.0%	0	< 0.1
P1	Max Profit	0.20	0.25	6	60	72	105,380	57,883	1,371.8	1,120.1	1,120.1	0.0%	290	3.3
P1	Max Profit	0.20	0.50	6	60	72	105,380	57,883	1,137.2	852.5	852.5	0.0%	728	6.5
P1	Max Profit	0.20	0.75	6	60	72	105,380	57,883	933.5	778.2	778.2	0.0%	264	3.0
P1	Max Profit	0.20	1.00	6	60	72	105,380	57,883	757.8	757.8	757.8	0.0%	38	0.7
P1	Max Profit	0.30	0.00	6	60	72	105,380	57,883	1,640.0	1,640.0	1,640.0	0.0%	0	< 0.1
P1	Max Profit	0.30	0.25	6	60	72	105,380	57,883	1,250.5	964.6	964.6	0.0%	224	2.4
P1	Max Profit	0.30	0.50	6	60	72	105,380	57,883	933.5	621.3	621.3	0.0%	1,760	11.4
P1	Max Profit	0.30	0.75	6	60	72	105,380	57,883	679.6	546.4	546.4	0.0%	417	4.3
P1	Max Profit	0.30	1.00	6	60	72	105,380	57,883	480.2	480.2	480.2	0.0%	0	1.2
P2	Max Profit	0.00	0.00	7	0	60	267	289	2,232.4	1,281.7	1,281.7	0.0%	247	< 0.1
P2	Max Profit	0.10	0.00	7	84	60	60,263	75,581	2,232.4	1,260.0	1,260.0	0.0%	3,217	3.0
P2	Max Profit	0.10	0.25	7	84	60	60,263	75,581	2,224.9	1,256.5	1,256.5	0.0%	2,996	2.0
P2	Max Profit	0.10	0.50	7	84	60	60,263	75,581	2,217.4	1,253.0	1,253.0	0.0%	2,620	2.5
P2	Max Profit	0.10	0.75	7	84	60	60,263	75,581	2,210.0	1,249.5	1,249.5	0.0%	2,504	2.3
P2	Max Profit	0.10	1.00	7	84	60	60,263	75,581	2,202.5	1,246.0	1,246.0	0.0%	1,258	1.6
P2	Max Profit	0.20	0.00	7	84	60	60,263	75,581	2,232.4	1,260.0	1,260.0	0.0%	3,263	3.0
P2	Max Profit	0.20	0.25	7	84	60	60,263	75,581	2,217.4	1,253.0	1,253.0	0.0%	2,708	2.1
P2	Max Profit	0.20	0.50	7	84	60	60,263	75,581	2,202.5	1,246.0	1,246.0	0.0%	1,311	1.5
P2	Max Profit	0.20	0.75	7	84	60	60,263	75,581	2,187.6	1,239.0	1,239.0	0.0%	897	1.1
P2	Max Profit	0.20	1.00	7	84	60	60,263	75,581	2,172.7	1,232.0	1,232.0	0.0%	666	1.0
P2	Max Profit	0.30	0.00	7	84	60	60,263	75,581	2,232.4	1,260.0	1,260.0	0.0%	3,047	2.7
P2	Max Profit	0.30	0.25	7	84	60	60,263	75,581	2,210.0	1,249.5	1,249.5	0.0%	3,481	2.6
P2	Max Profit	0.30	0.50	7	84	60	60,263	75,581	2,187.6	1,239.0	1,239.0	0.0%	967	1.4
P2	Max Profit	0.30	0.75	7	84	60	60,263	75,581	2,165.2	1,228.5	1,228.5	0.0%	756	1.0
P2	Max Profit	0.30	1.00	7	84	60	60,263	75,581	2,143.0	1,218.0	1,218.0	0.0%	851	1.0
P3	Max Profit	0.00	0.00	8	0	90	453	466	3,700.0	3,700.0	3,700.0	0.0%	0	< 0.1
P3	Max Profit	0.10	0.00	8	40	90	88,366	67,586	3,700.0	3,700.0	3,700.0	0.0%	0	< 0.1
P3	Max Profit	0.10	0.25	8	40	90	88,366	67,586	3,615.9	3,613.5	3,613.5	0.0%	521	9.0
P3	Max Profit	0.10	0.50	8	40	90	88,366	67,586	3,564.1	3,558.7	3,558.7	0.0%	242	4.9
P3	Max Profit	0.10	0.75	8	40	90	88,366	67,586	3,511.0	3,505.0	3,505.0	0.0%	35	2.3
P3	Max Profit	0.10	1.00	8	40	90	88,366	67,586	3,456.0	3,456.0	3,456.0	0.0%	0	< 0.1
P3	Max Profit	0.20	0.00	8	40	90	88,366	67,586	3,700.0	3,700.0	3,700.0	0.0%	0	< 0.1
P3	Max Profit	0.20	0.25	8	40	90	88,366	67,586	3,531.2	3,522.8	3,522.8	0.0%	420	7.7
P3	Max Profit	0.20	0.50	8	40	90	88,366	67,586	3,421.3	3,399.5	3,399.5	0.0%	280	7.7
P3	Max Profit	0.20	0.75	8	40	90	88,366	67,586	3,306.2	3,286.1	3,286.1	0.0%	158	5.5
P3	Max Profit	0.20	1.00	8	40	90	88,366	67,586	3,184.0	3,184.0	3,184.0	0.0%	0	0.5
P3	Max Profit	0.30	0.00	8	40	90	88,366	67,586	3,700.0	3,700.0	3,700.0	0.0%	0	< 0.1
P3	Max Profit	0.30	0.25	8	40	90	88,366	67,586	3,445.7	3,424.8	3,424.8	0.0%	668	12.8
P3	Max Profit	0.30	0.50	8	40	90	88,366	67,586	3,271.5	3,210.7	3,210.7	0.0%	230	7.4
P3	Max Profit	0.30	0.75	8	40	90	88,366	67,586	3,071.5	3,027.0	3,027.0	0.0%	144	5.4
P3	Max Profit	0.30	1.00	8	40	90	88,366	67,586	2,786.0	2,786.0	2,786.0	0.0%	0	0.5
P4	Max Profit	0.00	0.00	8	0	144	672	673	1,100.0	1,100.0	1,100.0	0.0%	0	< 0.1
P4	Max Profit	0.10	0.00	8	64	144	194,660	107,921	1,100.0	1,100.0	1,100.0	0.0%	0	< 0.1
P4	Max Profit	0.10	0.25	8	64	144	194,660	107,921	1,064.0	1,059.5	1,059.5	0.0%	3,604	162.3
P4	Max Profit	0.10	0.50	8	64	144	194,660	107,921	1,028.4	1,024.7	1,024.7	0.0%	1,993	166.7
P4	Max Profit	0.10	0.75	8	64	144	194,660	107,921	993.2	991.3	991.3	0.0%	1,824	354.1
P4	Max Profit	0.10	1.00	8	64	144	194,660	107,921	958.5	958.5	958.5	0.0%	0	< 0.1
P4	Max Profit	0.20	0.00	8	64	144	194,660	107,921	1,100.0	1,100.0	1,100.0	0.0%	0	< 0.1
P4	Max Profit	0.20	0.25	8	64	144	194,660	107,921	1,028.4	1,019.7	1,019.7	0.0%	2,203	266.4
P4	Max Profit	0.20	0.50	8	64	144	194,660	107,921	958.5	954.0	954.0	0.0%	1,634	230.3
P4	Max Profit	0.20	0.75	8	64	144	194,660	107,921	890.4	887.8	887.8	0.0%	1,465	190.7
P4	Max Profit	0.20	1.00	8	64	144	194,660	107,921	824.0	824.0	824.0	0.0%	0	< 0.1
P4	Max Profit	0.30	0.00	8	64	144	194,660	107,921	1,100.0	1,100.0	1,100.0	0.0%	0	< 0.1
P4	Max Profit	0.30	0.25	8	64	144	194,660	107,921	993.2	980.8	980.8	0.0%	3,155	299.9
P4	Max Profit	0.30	0.50	8	64	144	194,660	107,921	890.4	882.5	882.5	0.0%	2,799	335.9
P4	Max Profit	0.30	0.75	8	64	144	194,660	107,921	791.5	785.6	785.6	0.0%	4,780	417.2
P4	Max Profit	0.30	1.00	8	64	144	194,660	107,921	696.5	696.5	696.5	0.0%	0	< 0.1
P5	Max Profit	0.00	0.00	6	0	84	431	438	1,700.0	1,700.0	1,700.0	0.0%	0	< 0.1
P5	Max Profit	0.10	0.00	6	42	84	73,067	68,988	1,700.0	1,700.0	1,700.0	0.0%	17	0.3
P5	Max Profit	0.10	0.25	6	42	84	73,067	68,988	1,616.7	1,591.2	1,591.2	0.0%	118	1.0
P5	Max Profit	0.10	0.50	6	42	84	73,067	68,988	1,541.5	1,489.2	1,489.2	0.0%	368	3.4
P5	Max Profit	0.10	0.75	6	42	84	73,067	68,988	1,483.1	1,449.0	1,449.0	0.0%	180	1.3
P5	Max Profit	0.10	1.00	6	42	84	73,067	68,988	1,449.0	1,449.0	1,449.0	0.0%	12	0.3
P5	Max Profit	0.20	0.00	6	42	84	73,067	68,988	1,700.0	1,700.0	1,700.0	0.0%	31	0.5
P5	Max Profit	0.20	0.25	6	42	84	73,067	68,988	1,541.5	1,489.2	1,489.2	0.0%	187	1.4
P5	Max Profit	0.20	0.50	6	42	84	73,067	68,988	1,424.9	1,288.0	1,288.0	0.0%	482	3.7
P5	Max Profit	0.20	0.75	6	42	84	73,067	68,988	1,326.8	1,255.2	1,255.2	0.0%	318	2.7
P5	Max Profit	0.20	1.00	6	42	84	73,067	68,988	1,248.0	1,248.0	1,248.0	0.0%	0	0.1
P5	Max Profit	0.30	0.00	6	42	84	73,067	68,988	1,700.0	1,700.0	1,700.0	0.0%	15	0.4
P5	Max Profit	0.30	0.25	6	42	84	73,067	68,988	1,482.1	1,387.6	1,387.6	0.0%	396	2.8
P5	Max Profit	0.30	0.50	6	42	84	73,067	68,988	1,318.4	1,113.7	1,113.7	0.0%	355	3.0
P5	Max Profit	0.30	0.75	6	42	84	73,067	68,988	1,178.7	1,077.2	1,077.2	0.0%	255	2.2
P5	Max Profit	0.30	1.00	6	42	84	73,067	68,988	1,060.5	1,060.5	1,060.5	0.0%	10	0.2

Problem	Obj. Type	ξ	ϕ	# Ev. Points	# Uncert. Param.	# Bin. Var.	# Cont. Var.	# Constraints	Root Node Rlx.	Rel. Obj. Val.	Obj. Val.	Opt. Gap	# Nodes	CPU Time (s)
P6	Max Profit	0.00	0.00	10	0	168	1,027	878	4,950.0	4,000.0	4,000.0	0.0%	1,653	0.8
P6	Max Profit	0.10	0.00	10	80	168	215,189	271,198	4,950.0	4,000.0	4,000.0	0.0%	4,388	528.3
P6	Max Profit	0.10	0.25	10	80	168	215,189	271,198	4,788.3	3,812.0	3,812.0	0.0%	13,493	874.1
P6	Max Profit	0.10	0.50	10	80	168	215,189	271,198	4,660.3	3,674.7	3,674.7	0.0%	9,286	1,118.5
P6	Max Profit	0.10	0.75	10	80	168	215,189	271,198	4,548.3	3,600.0	3,600.0	0.0%	8,516	727.6
P6	Max Profit	0.10	1.00	10	80	168	215,189	271,198	4,455.0	3,600.0	3,600.0	0.0%	4,430	232.7
P6	Max Profit	0.20	0.00	10	80	168	215,189	271,198	4,950.0	4,000.0	4,000.0	0.0%	4,831	361.5
P6	Max Profit	0.20	0.25	10	80	168	215,189	271,198	4,626.7	3,630.0	3,630.0	0.0%	8,239	848.1
P6	Max Profit	0.20	0.50	10	80	168	215,189	271,198	4,370.7	3,364.6	3,364.6	0.0%	9,089	793.7
P6	Max Profit	0.20	0.75	10	80	168	215,189	271,198	4,146.7	3,210.7	3,210.7	0.0%	17,518	1,327.0
P6	Max Profit	0.20	1.00	10	80	168	215,189	271,198	3,960.0	3,200.0	3,200.0	0.0%	4,900	490.3
P6	Max Profit	0.30	0.00	10	80	168	215,189	271,198	4,950.0	4,000.0	4,000.0	0.0%	4,044	465.2
P6	Max Profit	0.30	0.25	10	80	168	215,189	271,198	4,465.0	3,451.0	3,451.0	0.0%	9,349	1,067.7
P6	Max Profit	0.30	0.50	10	80	168	215,189	271,198	4,081.0	3,068.2	3,068.2	0.0%	10,726	645.5
P6	Max Profit	0.30	0.75	10	80	168	215,189	271,198	3,745.0	2,841.0	2,841.0	0.0%	13,314	818.5
P6	Max Profit	0.30	1.00	10	80	168	215,189	271,198	3,465.0	2,800.0	2,800.0	0.0%	3,537	198.6
P7	Max Profit	0.00	0.00	8	0	180	2,142	1,251	3,157.5	2,690.0	2,690.0	0.0%	105	< 0.1
P7	Max Profit	0.10	0.00	8	104	180	1,646,630	802,859	3,157.5	2,690.0	2,690.0	0.0%	508	492.3
P7	Max Profit	0.10	0.25	8	104	180	1,646,630	802,859	3,151.6	2,687.5	2,687.5	0.0%	5,280	2,013.8
P7	Max Profit	0.10	0.50	8	104	180	1,646,630	802,859	3,145.6	2,686.7	2,686.7	0.0%	1,721	695.3
P7	Max Profit	0.10	0.75	8	104	180	1,646,630	802,859	3,139.7	2,686.7	2,686.7	0.0%	1,946	950.3
P7	Max Profit	0.10	1.00	8	104	180	1,646,630	802,859	3,133.8	2,686.7	2,686.7	0.0%	94	32.0
P7	Max Profit	0.20	0.00	8	104	180	1,646,630	802,859	3,157.5	2,690.0	2,690.0	0.0%	829	860.2
P7	Max Profit	0.20	0.25	8	104	180	1,646,630	802,859	3,145.6	2,685.1	2,685.1	0.0%	5,017	1,571.4
P7	Max Profit	0.20	0.50	8	104	180	1,646,630	802,859	3,133.8	2,683.6	2,683.6	0.0%	2,211	978.1
P7	Max Profit	0.20	0.75	8	104	180	1,646,630	802,859	3,121.9	2,683.6	2,683.6	0.0%	1,285	351.4
P7	Max Profit	0.20	1.00	8	104	180	1,646,630	802,859	3,110.0	2,683.6	2,683.6	0.0%	75	37.6
P7	Max Profit	0.30	0.00	8	104	180	1,646,630	802,859	3,157.5	2,690.0	2,690.0	0.0%	639	820.4
P7	Max Profit	0.30	0.25	8	104	180	1,646,630	802,859	3,139.7	2,682.9	2,682.9	0.0%	4,047	1,401.8
P7	Max Profit	0.30	0.50	8	104	180	1,646,630	802,859	3,121.9	2,680.9	2,680.9	0.0%	1,960	516.2
P7	Max Profit	0.30	0.75	8	104	180	1,646,630	802,859	3,101.3	2,680.9	2,680.9	0.0%	935	319.0
P7	Max Profit	0.30	1.00	8	104	180	1,646,630	802,859	3,063.1	2,680.9	2,680.9	0.0%	316	69.0
P8	Max Profit	0.00	0.00	9	0	210	2,180	1,665	4,980.0	4,295.0	4,295.0	0.0%	3,646	1.7
P8	Max Profit	0.10	0.00	9	63	210	601,724	579,528	4,980.0	4,295.0	4,295.0	0.0%	3,955	1,460.3
P8	Max Profit	0.10	0.25	9	63	210	601,724	579,528	4,957.3	4,190.5	4,190.5	0.0%	7,494	2,700.7
P8	Max Profit	0.10	0.50	9	63	210	601,724	579,528	4,955.5	4,163.2	4,163.2	0.0%	4,587	2,111.1
P8	Max Profit	0.10	0.75	9	63	210	601,724	579,528	4,953.8	4,163.2	4,163.2	0.0%	8,045	2,925.9
P8	Max Profit	0.10	1.00	9	63	210	601,724	579,528	4,952.0	4,163.2	4,163.2	0.0%	2,387	427.8
P8	Max Profit	0.20	0.00	9	63	210	601,724	579,528	4,980.0	4,295.0	4,295.0	0.0%	3,168	1,611.6
P8	Max Profit	0.20	0.25	9	63	210	601,724	579,528	4,934.5	4,098.8	4,098.8	0.0%	6,880	2,800.9
P8	Max Profit	0.20	0.50	9	63	210	601,724	579,528	4,931.0	4,042.4	4,042.4	0.0%	4,669	2,767.5
P8	Max Profit	0.20	0.75	9	63	210	601,724	579,528	4,927.5	4,042.4	4,042.4	0.0%	6,859	2,458.4
P8	Max Profit	0.20	1.00	9	63	210	601,724	579,528	4,924.0	4,042.4	4,042.4	0.0%	4,281	434.4
P8	Max Profit	0.30	0.00	9	63	210	601,724	579,528	4,980.0	4,295.0	4,295.0	0.0%	3,926	1,486.9
P8	Max Profit	0.30	0.25	9	63	210	601,724	579,528	4,911.8	4,039.2	4,039.2	0.0%	9,714	3,936.6
P8	Max Profit	0.30	0.50	9	63	210	601,724	579,528	4,906.5	3,945.2	3,945.2	0.0%	9,816	3,678.4
P8	Max Profit	0.30	0.75	9	63	210	601,724	579,528	4,901.3	3,945.2	3,945.2	0.0%	5,906	2,532.6
P8	Max Profit	0.30	1.00	9	63	210	601,724	579,528	4,896.0	3,945.2	3,945.2	0.0%	3,772	406.5
P9	Max Profit	0.00	0.00	8	0	198	2,647	1,912	6,752.9	4,340.0	4,340.0	0.0%	8,454	3.8
P9	Max Profit	0.10	0.00	8	88	198	1,492,337	980,576	6,752.9	4,340.0	4,340.0	0.0%	8,572	7,710.5
P9	Max Profit	0.10	0.25	8	88	198	1,492,337	980,576	6,694.4	4,315.3	4,315.3	0.0%	15,180	9,355.4
P9	Max Profit	0.10	0.50	8	88	198	1,492,337	980,576	6,688.9	4,308.0	4,308.0	0.0%	15,373	6,803.0
P9	Max Profit	0.10	0.75	8	88	198	1,492,337	980,576	6,685.4	4,308.0	4,308.0	0.0%	12,204	5,970.5
P9	Max Profit	0.10	1.00	8	88	198	1,492,337	980,576	6,682.0	4,308.0	4,308.0	0.0%	12,986	3,345.2
P9	Max Profit	0.20	0.00	8	88	198	1,492,337	980,576	6,752.9	4,340.0	4,340.0	0.0%	12,623	10,066.9
P9	Max Profit	0.20	0.25	8	88	198	1,492,337	980,576	6,688.9	4,302.0	4,302.0	0.0%	16,011	9,336.5
P9	Max Profit	0.20	0.50	8	88	198	1,492,337	980,576	6,682.0	4,294.0	4,294.0	0.0%	13,981	7,134.0
P9	Max Profit	0.20	0.75	8	88	198	1,492,337	980,576	6,675.1	4,294.0	4,294.0	0.0%	13,252	6,304.6
P9	Max Profit	0.20	1.00	8	88	198	1,492,337	980,576	6,668.3	4,294.0	4,294.0	0.0%	8,233	2,811.9
P9	Max Profit	0.30	0.00	8	88	198	1,492,337	980,576	6,752.9	4,340.0	4,340.0	0.0%	11,676	10,948.2
P9	Max Profit	0.30	0.25	8	88	198	1,492,337	980,576	6,685.4	4,298.0	4,298.0	0.0%	13,286	10,474.5
P9	Max Profit	0.30	0.50	8	88	198	1,492,337	980,576	6,675.1	4,286.0	4,286.0	0.0%	13,183	8,870.5
P9	Max Profit	0.30	0.75	8	88	198	1,492,337	980,576	6,665.2	4,286.0	4,286.0	0.0%	11,932	5,812.3
P9	Max Profit	0.30	1.00	8	88	198	1,492,337	980,576	6,655.5	4,286.0	4,286.0	0.0%	8,157	2,332.8

Problem	Obj. Type	ξ	ϕ	# Ev. Points	# Uncert. Param.	# Bin. Var.	# Cont. Var.	# Constraints	Root Node Rlx.	Rel. Obj. Val.	Obj. Val.	Opt. Gap	# Nodes	CPU Time (s)
P1	Min Makespan	0.00	0.00	6	0	72	288	353	3.1	5.4	5.4	0.0%	84	< 0.1
P1	Min Makespan	0.10	0.00	6	60	72	113,048	61,673	1.3	5.4	5.4	0.0%	106	1.5
P1	Min Makespan	0.10	0.25	6	60	72	113,048	61,673	1.3	5.4	5.4	0.0%	70	1.8
P1	Min Makespan	0.10	0.50	6	60	72	113,048	61,673	1.3	5.4	5.4	0.0%	159	2.5
P1	Min Makespan	0.10	0.75	6	60	72	113,048	61,673	1.4	5.4	5.4	0.0%	72	1.5
P1	Min Makespan	0.10	1.00	6	60	72	113,048	61,673	1.4	5.4	5.4	0.0%	124	1.3
P1	Min Makespan	0.20	0.00	6	60	72	113,048	61,673	1.3	5.4	5.4	0.0%	127	1.6
P1	Min Makespan	0.20	0.25	6	60	72	113,048	61,673	1.3	5.4	5.4	0.0%	87	2.0
P1	Min Makespan	0.20	0.50	6	60	72	113,048	61,673	1.4	5.4	5.4	0.0%	70	1.5
P1	Min Makespan	0.20	0.75	6	60	72	113,048	61,673	1.5	5.4	5.4	0.0%	106	2.3
P1	Min Makespan	0.20	1.00	6	60	72	113,048	61,673	1.7	5.4	5.4	0.0%	45	1.3
P1	Min Makespan	0.30	0.00	6	60	72	113,048	61,673	1.3	5.4	5.4	0.0%	199	2.5
P1	Min Makespan	0.30	0.25	6	60	72	113,048	61,673	1.4	5.4	5.4	0.0%	150	2.4
P1	Min Makespan	0.30	0.50	6	60	72	113,048	61,673	1.5	5.8	5.8	0.0%	333	5.9
P1	Min Makespan	0.30	0.75	6	60	72	113,048	61,673	1.7	5.8	5.8	0.0%	248	4.5
P1	Min Makespan	0.30	1.00	6	60	72	113,048	61,673	2.0	5.8	5.8	0.0%	85	1.2
P2	Min Makespan	0.00	0.00	8	0	72	316	349	1.5	5.2	5.2	0.0%	1,118	0.1
P2	Min Makespan	0.10	0.00	8	96	72	82,556	103,389	1.5	5.2	5.2	0.0%	933	1.3
P2	Min Makespan	0.10	0.25	8	96	72	82,556	103,389	1.5	5.2	5.2	0.0%	1,371	2.2
P2	Min Makespan	0.10	0.50	8	96	72	82,556	103,389	1.5	5.2	5.2	0.0%	729	1.6
P2	Min Makespan	0.10	0.75	8	96	72	82,556	103,389	1.5	5.2	5.2	0.0%	2,123	4.2
P2	Min Makespan	0.10	1.00	8	96	72	82,556	103,389	1.5	5.2	5.2	0.0%	411	1.2
P2	Min Makespan	0.20	0.00	8	96	72	82,556	103,389	1.5	5.2	5.2	0.0%	2,373	3.2
P2	Min Makespan	0.20	0.25	8	96	72	82,556	103,389	1.5	5.2	5.2	0.0%	829	1.9
P2	Min Makespan	0.20	0.50	8	96	72	82,556	103,389	1.5	5.2	5.2	0.0%	657	1.7
P2	Min Makespan	0.20	0.75	8	96	72	82,556	103,389	1.6	5.2	5.2	0.0%	1,506	3.3
P2	Min Makespan	0.20	1.00	8	96	72	82,556	103,389	1.6	5.2	5.2	0.0%	682	1.8
P2	Min Makespan	0.30	0.00	8	96	72	82,556	103,389	1.5	5.2	5.2	0.0%	1,204	2.4
P2	Min Makespan	0.30	0.25	8	96	72	82,556	103,389	1.5	5.2	5.2	0.0%	1,364	2.8
P2	Min Makespan	0.30	0.50	8	96	72	82,556	103,389	1.6	5.2	5.2	0.0%	377	1.2
P2	Min Makespan	0.30	0.75	8	96	72	82,556	103,389	1.6	5.2	5.2	0.0%	935	2.3
P2	Min Makespan	0.30	1.00	8	96	72	82,556	103,389	1.6	5.2	5.2	0.0%	475	1.1
P3	Min Makespan	0.00	0.00	8	0	90	453	479	1.6	3.8	3.8	0.0%	501	< 0.1
P3	Min Makespan	0.10	0.00	8	40	90	93,262	70,959	1.6	3.8	3.8	0.0%	606	19.6
P3	Min Makespan	0.10	0.25	8	40	90	93,262	70,959	1.7	3.8	3.8	0.0%	1,069	19.2
P3	Min Makespan	0.10	0.50	8	40	90	93,262	70,959	1.7	3.8	3.8	0.0%	945	33.5
P3	Min Makespan	0.10	0.75	8	40	90	93,262	70,959	1.7	3.8	3.8	0.0%	1,251	39.2
P3	Min Makespan	0.10	1.00	8	40	90	93,262	70,959	1.8	3.8	3.8	0.0%	976	17.5
P3	Min Makespan	0.20	0.00	8	40	90	93,262	70,959	1.6	3.8	3.8	0.0%	598	10.4
P3	Min Makespan	0.20	0.25	8	40	90	93,262	70,959	1.7	3.8	3.8	0.0%	805	25.4
P3	Min Makespan	0.20	0.50	8	40	90	93,262	70,959	1.8	3.8	3.8	0.0%	1,188	43.8
P3	Min Makespan	0.20	0.75	8	40	90	93,262	70,959	1.8	3.8	3.8	0.0%	435	21.3
P3	Min Makespan	0.20	1.00	8	40	90	93,262	70,959	1.9	3.8	3.8	0.0%	427	11.8
P3	Min Makespan	0.30	0.00	8	40	90	93,262	70,959	1.6	3.8	3.8	0.0%	388	11.4
P3	Min Makespan	0.30	0.25	8	40	90	93,262	70,959	1.7	3.9	3.9	0.0%	652	18.1
P3	Min Makespan	0.30	0.50	8	40	90	93,262	70,959	1.8	3.9	3.9	0.0%	714	26.5
P3	Min Makespan	0.30	0.75	8	40	90	93,262	70,959	2.0	3.9	3.9	0.0%	843	28.3
P3	Min Makespan	0.30	1.00	8	40	90	93,262	70,959	2.1	3.9	3.9	0.0%	542	11.2
P4	Min Makespan	0.00	0.00	6	0	96	460	480	0.6	2.2	2.2	0.0%	130	< 0.1
P4	Min Makespan	0.10	0.00	6	48	96	118,468	64,788	0.6	2.2	2.2	0.0%	45	1.4
P4	Min Makespan	0.10	0.25	6	48	96	118,468	64,788	0.6	2.2	2.2	0.0%	44	2.5
P4	Min Makespan	0.10	0.50	6	48	96	118,468	64,788	0.6	2.2	2.2	0.0%	54	2.8
P4	Min Makespan	0.10	0.75	6	48	96	118,468	64,788	0.6	2.2	2.2	0.0%	42	2.5
P4	Min Makespan	0.10	1.00	6	48	96	118,468	64,788	0.7	2.2	2.2	0.0%	52	1.5
P4	Min Makespan	0.20	0.00	6	48	96	118,468	64,788	0.6	2.2	2.2	0.0%	108	2.3
P4	Min Makespan	0.20	0.25	6	48	96	118,468	64,788	0.6	2.2	2.2	0.0%	44	2.0
P4	Min Makespan	0.20	0.50	6	48	96	118,468	64,788	0.7	2.2	2.2	0.0%	81	2.1
P4	Min Makespan	0.20	0.75	6	48	96	118,468	64,788	0.7	2.2	2.2	0.0%	291	5.6
P4	Min Makespan	0.20	1.00	6	48	96	118,468	64,788	0.7	2.2	2.2	0.0%	389	3.7
P4	Min Makespan	0.30	0.00	6	48	96	118,468	64,788	0.6	2.2	2.2	0.0%	51	1.5
P4	Min Makespan	0.30	0.25	6	48	96	118,468	64,788	0.6	2.2	2.2	0.0%	243	5.0
P4	Min Makespan	0.30	0.50	6	48	96	118,468	64,788	0.7	2.2	2.2	0.0%	223	3.8
P4	Min Makespan	0.30	0.75	6	48	96	118,468	64,788	0.7	2.2	2.2	0.0%	414	9.3
P4	Min Makespan	0.30	1.00	6	48	96	118,468	64,788	0.8	2.2	2.2	0.0%	544	5.7
P5	Min Makespan	0.00	0.00	7	0	105	528	550	1.0	6.3	6.3	0.0%	404	< 0.1
P5	Min Makespan	0.10	0.00	7	49	105	105,427	99,390	1.0	6.3	6.3	0.0%	435	7.8
P5	Min Makespan	0.10	0.25	7	49	105	105,427	99,390	1.0	6.3	6.3	0.0%	100	4.0
P5	Min Makespan	0.10	0.50	7	49	105	105,427	99,390	1.1	6.3	6.3	0.0%	264	5.0
P5	Min Makespan	0.10	0.75	7	49	105	105,427	99,390	1.1	6.3	6.3	0.0%	112	5.8
P5	Min Makespan	0.10	1.00	7	49	105	105,427	99,390	1.2	6.3	6.3	0.0%	375	4.5
P5	Min Makespan	0.20	0.00	7	49	105	105,427	99,390	1.0	6.3	6.3	0.0%	484	7.0
P5	Min Makespan	0.20	0.25	7	49	105	105,427	99,390	1.1	6.3	6.3	0.0%	312	7.7
P5	Min Makespan	0.20	0.50	7	49	105	105,427	99,390	1.2	6.3	6.3	0.0%	120	6.1
P5	Min Makespan	0.20	0.75	7	49	105	105,427	99,390	1.3	6.3	6.3	0.0%	193	5.9
P5	Min Makespan	0.20	1.00	7	49	105	105,427	99,390	1.4	6.3	6.3	0.0%	380	6.1
P5	Min Makespan	0.30	0.00	7	49	105	105,427	99,390	1.0	6.3	6.3	0.0%	321	4.6
P5	Min Makespan	0.30	0.25	7	49	105	105,427	99,390	1.1	6.3	6.3	0.0%	693	7.2
P5	Min Makespan	0.30	0.50	7	49	105	105,427	99,390	1.3	6.3	6.3	0.0%	292	6.5
P5	Min Makespan	0.30	0.75	7	49	105	105,427	99,390	1.5	6.3	6.3	0.0%	154	4.7
P5	Min Makespan	0.30	1.00	7	49	105	105,427	99,390	1.8	6.3	6.3	0.0%	117	2.9

Problem	Obj. Type	ξ	ϕ	# Ev. Points	# Uncert. Param.	# Bin. Var.	# Cont. Var.	# Constraints	Root Node Rlx.	Rel. Obj. Val.	Obj. Val.	Opt. Gap	# Nodes	CPU Time (s)
P6	Min Makespan	0.00	0.00	10	0	168	1,027	899	1.1	5.0	5.0	0.0%	17,995	5.9
P6	Min Makespan	0.10	0.00	10	80	168	224,869	282,419	1.1	5.0	5.0	0.0%	30,820	1,575.4
P6	Min Makespan	0.10	0.25	10	80	168	224,869	282,419	1.1	5.0	5.0	0.0%	27,540	1,977.7
P6	Min Makespan	0.10	0.50	10	80	168	224,869	282,419	1.2	5.0	5.0	0.0%	33,698	1,879.5
P6	Min Makespan	0.10	0.75	10	80	168	224,869	282,419	1.2	5.0	5.0	0.0%	30,618	2,509.1
P6	Min Makespan	0.10	1.00	10	80	168	224,869	282,419	1.2	5.0	5.0	0.0%	28,024	1,374.0
P6	Min Makespan	0.20	0.00	10	80	168	224,869	282,419	1.1	5.0	5.0	0.0%	64,881	2,740.5
P6	Min Makespan	0.20	0.25	10	80	168	224,869	282,419	1.2	5.0	5.0	0.0%	21,780	1,542.5
P6	Min Makespan	0.20	0.50	10	80	168	224,869	282,419	1.2	5.0	5.0	0.0%	22,194	1,837.4
P6	Min Makespan	0.20	0.75	10	80	168	224,869	282,419	1.3	5.0	5.0	0.0%	33,148	2,394.3
P6	Min Makespan	0.20	1.00	10	80	168	224,869	282,419	1.4	5.0	5.0	0.0%	6,399	260.0
P6	Min Makespan	0.30	0.00	10	80	168	224,869	282,419	1.1	5.0	5.0	0.0%	49,342	2,135.1
P6	Min Makespan	0.30	0.25	10	80	168	224,869	282,419	1.2	5.2	5.2	0.0%	201,752	12,565.7
P6	Min Makespan	0.30	0.50	10	80	168	224,869	282,419	1.3	5.2	5.2	0.0%	54,470	3,332.5
P6	Min Makespan	0.30	0.75	10	80	168	224,869	282,419	1.4	5.2	5.2	0.0%	112,920	5,821.5
P6	Min Makespan	0.30	1.00	10	80	168	224,869	282,419	1.6	5.2	5.2	0.0%	47,898	1,637.8
P7	Min Makespan	0.00	0.00	8	0	180	2,142	1,311	1.1	5.3	5.3	0.0%	233	0.1
P7	Min Makespan	0.10	0.00	8	104	180	1,745,632	845,871	1.1	5.3	5.3	0.0%	658	832.1
P7	Min Makespan	0.10	0.25	8	104	180	1,745,632	845,871	1.1	5.3	5.3	0.0%	827	1,149.5
P7	Min Makespan	0.10	0.50	8	104	180	1,745,632	845,871	1.1	5.3	5.3	0.0%	829	857.8
P7	Min Makespan	0.10	0.75	8	104	180	1,745,632	845,871	1.1	5.3	5.3	0.0%	1,947	1,016.8
P7	Min Makespan	0.10	1.00	8	104	180	1,745,632	845,871	1.2	5.3	5.3	0.0%	3,517	1,076.5
P7	Min Makespan	0.20	0.00	8	104	180	1,745,632	845,871	1.1	5.3	5.3	0.0%	445	705.0
P7	Min Makespan	0.20	0.25	8	104	180	1,745,632	845,871	1.1	5.3	5.3	0.0%	1,875	1,101.7
P7	Min Makespan	0.20	0.50	8	104	180	1,745,632	845,871	1.2	5.3	5.3	0.0%	1,709	937.5
P7	Min Makespan	0.20	0.75	8	104	180	1,745,632	845,871	1.2	5.3	5.3	0.0%	765	975.4
P7	Min Makespan	0.20	1.00	8	104	180	1,745,632	845,871	1.2	5.3	5.3	0.0%	1,125	579.4
P7	Min Makespan	0.30	0.00	8	104	180	1,745,632	845,871	1.1	5.3	5.3	0.0%	527	983.9
P7	Min Makespan	0.30	0.25	8	104	180	1,745,632	845,871	1.1	5.3	5.3	0.0%	944	849.2
P7	Min Makespan	0.30	0.50	8	104	180	1,745,632	845,871	1.2	5.3	5.3	0.0%	1,596	1,072.4
P7	Min Makespan	0.30	0.75	8	104	180	1,745,632	845,871	1.3	5.3	5.3	0.0%	1,343	1,330.9
P7	Min Makespan	0.30	1.00	8	104	180	1,745,632	845,871	1.3	5.3	5.3	0.0%	593	418.1
P8	Min Makespan	0.00	0.00	9	0	210	2,180	1,727	1.3	5.8	5.8	0.0%	12,462	6.0
P8	Min Makespan	0.10	0.00	9	63	210	632,834	606,491	1.3	5.8	5.8	0.0%	25,348	4,935.8
P8	Min Makespan	0.10	0.25	9	63	210	632,834	606,491	1.3	5.8	5.8	0.0%	12,737	4,171.0
P8	Min Makespan	0.10	0.50	9	63	210	632,834	606,491	1.3	5.8	5.8	0.0%	13,216	5,605.7
P8	Min Makespan	0.10	0.75	9	63	210	632,834	606,491	1.3	5.8	5.8	0.0%	15,062	5,120.3
P8	Min Makespan	0.10	1.00	9	63	210	632,834	606,491	1.4	5.8	5.8	0.0%	17,889	2,492.0
P8	Min Makespan	0.20	0.00	9	63	210	632,834	606,491	1.3	5.8	5.8	0.0%	17,988	4,791.7
P8	Min Makespan	0.20	0.25	9	63	210	632,834	606,491	1.3	5.8	5.8	0.0%	20,727	5,669.7
P8	Min Makespan	0.20	0.50	9	63	210	632,834	606,491	1.4	5.8	5.8	0.0%	18,575	5,740.9
P8	Min Makespan	0.20	0.75	9	63	210	632,834	606,491	1.4	5.8	5.8	0.0%	32,319	5,474.0
P8	Min Makespan	0.20	1.00	9	63	210	632,834	606,491	1.4	5.8	5.8	0.0%	18,075	3,533.2
P8	Min Makespan	0.30	0.00	9	63	210	632,834	606,491	1.3	5.8	5.8	0.0%	17,193	5,028.9
P8	Min Makespan	0.30	0.25	9	63	210	632,834	606,491	1.3	5.8	5.8	0.0%	18,104	4,919.9
P8	Min Makespan	0.30	0.50	9	63	210	632,834	606,491	1.4	5.8	5.8	0.0%	16,753	5,361.8
P8	Min Makespan	0.30	0.75	9	63	210	632,834	606,491	1.5	5.8	5.8	0.0%	15,353	4,659.3
P8	Min Makespan	0.30	1.00	9	63	210	632,834	606,491	1.5	5.8	5.8	0.0%	20,485	2,437.2
P9	Min Makespan	0.00	0.00	7	0	165	2,236	1,731	1.0	6.8	6.8	0.0%	1,706	0.3
P9	Min Makespan	0.10	0.00	7	77	165	1,222,264	794,691	0.5	6.8	6.8	0.0%	759	216.7
P9	Min Makespan	0.10	0.25	7	77	165	1,222,264	794,691	0.5	6.8	6.8	0.0%	865	341.7
P9	Min Makespan	0.10	0.50	7	77	165	1,222,264	794,691	0.6	6.8	6.8	0.0%	1,168	230.3
P9	Min Makespan	0.10	0.75	7	77	165	1,222,264	794,691	0.6	6.8	6.8	0.0%	1,164	310.7
P9	Min Makespan	0.10	1.00	7	77	165	1,222,264	794,691	0.6	6.8	6.8	0.0%	535	155.0
P9	Min Makespan	0.20	0.00	7	77	165	1,222,264	794,691	0.5	6.8	6.8	0.0%	1,175	271.7
P9	Min Makespan	0.20	0.25	7	77	165	1,222,264	794,691	0.6	6.8	6.8	0.0%	1,105	325.7
P9	Min Makespan	0.20	0.50	7	77	165	1,222,264	794,691	0.6	6.8	6.8	0.0%	754	222.5
P9	Min Makespan	0.20	0.75	7	77	165	1,222,264	794,691	0.6	6.8	6.8	0.0%	1,454	435.5
P9	Min Makespan	0.20	1.00	7	77	165	1,222,264	794,691	0.6	6.8	6.8	0.0%	647	192.4
P9	Min Makespan	0.30	0.00	7	77	165	1,222,264	794,691	0.5	6.8	6.8	0.0%	493	174.5
P9	Min Makespan	0.30	0.25	7	77	165	1,222,264	794,691	0.6	6.8	6.8	0.0%	1,328	379.2
P9	Min Makespan	0.30	0.50	7	77	165	1,222,264	794,691	0.6	6.8	6.8	0.0%	1,053	272.4
P9	Min Makespan	0.30	0.75	7	77	165	1,222,264	794,691	0.6	6.8	6.8	0.0%	1,184	388.7
P9	Min Makespan	0.30	1.00	7	77	165	1,222,264	794,691	0.6	6.8	6.8	0.0%	615	134.0