Multi-Stage Adjustable Robust Optimization for Process Scheduling Under Uncertainty

Nikolaos H. Lappas and Chrysanthos E. Gounaris
Dept. of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

DOI 10.1002/aic.15183
Published online March 14, 2016 in Wiley Online Library (wileyonlinelibrary.com)

Variations in parameters such as processing times, yields, and availability of materials and utilities can have a detrimental effect in the optimality and/or feasibility of an otherwise "optimal" production schedule. In this article, we propose a multi-stage adjustable robust optimization approach to alleviate the risk from such operational uncertainties during scheduling decisions. We derive a novel robust counterpart of a deterministic scheduling model, and we show how to obey the observability and non-anticipativity restrictions that are necessary for the resulting solution policy to be implementable in practice. We also develop decision-dependent uncertainty sets to model the endogenous uncertainty that is inherently present in process scheduling applications. A computational study reveals that, given a chosen level of robustness, adjusting decisions to past parameter realizations leads to significant improvements, both in terms of worst-case objective as well as objective in expectation, compared to the traditional robust scheduling approaches. © 2016 American Institute of Chemical Engineers *AIChE J*, 62: 1646–1667, 2016
Keywords: process scheduling, uncertainty, robust optimization

Introduction

Multipurpose, batch-processing facilities can be found in a wide range of industries that produce chemicals,[1,2] metals,[3,4] semiconductors,[5] food,[6–8] and pharmaceuticals,[9,10] to name but a few. The coordination of limited available resources, while attempting to satisfy tight demand profiles, quality controls, and environmental restrictions, provides plant operators with a major challenge, and the ability to readily calculate optimal short-term production schedules can critically impact a facility's competitiveness and long-term economic viability. Scheduling production in batch-processing environments typically includes decisions for the assignment of tasks to processing units, the sequencing and timing of these tasks, and finally the selection of the batch sizes to be processed. Given the stakes, significant work has been performed during the past two decades in terms of modeling and optimizing these decisions, and a number of excellent reviews have been published on the topic.[11–13] To further compound the challenge for the operator, there may inherently exist considerable uncertainty in process parameters. In principle, parameters such as task processing times, changeover durations, process yields, and resource consumption rates, availability of raw materials and utilities, as well as final product demands and prices, can all be uncertain at the time the operator has to commit to a schedule. Ignoring such operational uncertainties and choosing a schedule by taking into account only some nominal, "average" scenario for the realization of process parameters can lead to a significant amount of forgone profit or cause infeasibilities that render the schedule non-implementable. It is thus necessary to employ in this context a risk-averse decision-making framework that explicitly considers the effects of uncertainty and ensures that the schedule remains of value to the operation for a wide range of future outcomes.[14,15]

One approach that has been previously considered with success in process scheduling applications is the robust optimization approach robust optimization, which has also been previously considered with success in the context of process scheduling applications.[16–18] Robust optimization seeks to generate a solution that is immune to uncertainty by ensuring that it remains feasible for all possible realizations of the uncertain parameters from within a set chosen *a priori* by the modeler. Typical uncertainty sets include various types of polyhedra (e.g., boxes,[19] budgets,[20] factor models[21]), ellipsoidal sets,[22] as well as their intersections.[23] The size of the chosen set reflects the modeler's risk tolerance, while the set's shape encodes knowledge that the modeler may have with regards to existing correlations among the realizations of uncertain parameters. Given probability distributions that govern these realizations, the bounds on constraint satisfaction that the above sets offer have been extensively studied and compared.[20,24,25] One of the main advantages of robust optimization is that, in many cases, it enjoys similar tractability properties as the corresponding deterministic setting, which is very attractive from a computational viewpoint.

Correspondence concerning this article should be addressed to C. E. Gounaris at gounaris@cmu.edu.

Lin et al.[26] was the first study to formally apply robust optimization in the process scheduling context. This work, which utilized box uncertainty sets, was later extended by Janak et al.[24] to consider uncertainty sets derived from probabilistic information. A different study was presented by Li and Ierapetritou,[27] who considered box, ellipsoidal and budget uncertainty sets. A comparative study among the robust counterparts for the various types of uncertainty sets with examples from process scheduling instances was presented in a series of papers by Floudas and co-workers.[23,25,28] We collectively refer to the above single-stage approaches as static robust optimization (SRO). SRO has been since applied to the scheduling of crude oil operations[29] and of the continuous casting processes used in steel-making.[30]

The SRO approaches offer a good trade-off between numerical tractability and solution robustness, however they are generally conservative, as they assume that all of the decisions have to be made "here-and-now," before the schedule begins to be implemented. In reality, however, only a subset of the decisions has to be made at the beginning of the scheduling horizon. Many decisions can be "wait-and-see," meaning that they can be delayed until a later point when presumably a subset of the uncertain parameters have revealed their values. In fact, as soon as information about a specific parameter realization becomes available from plant observations, the operator may benefit by appropriately adjusting the subsequent decisions based on the actual realized value. This is true, of course, as long as such adjustment can be implemented expeditiously (e.g., via some sort of simple function evaluation rather than a full reoptimization procedure), which is of interest in contexts such as short-term production scheduling. To handle such multi-stage decision-making settings, Ben-Tal et al.[31] introduced the concept of adjustable robust optimization (ARO), where an optimal policy is derived instead of a single, static solution. The optimal policy constitutes a family of solutions that are parameterized in the uncertain parameter realizations. Typically, an affine relationship is postulated at the interest of numerical tractability, with a number of alternatives having been proposed by a number of researchers.[32–34] Recently, Shi and You[35] applied the Benders-dual-cutting-plane approach[36] and the column-and-constraint-generation approach[37] to solve a two-stage robust process scheduling problem. The two-stage adaptive setup improves on the SRO (i.e., single-stage) solutions, although it is limited to cases where all uncertain information reveals itself before any second-stage decision is taken, which may not be realistic in certain applications. For example, a task's true duration or final yield is usually revealed once the task completes, and since tasks generally begin and conclude at various time points within the horizon, many decisions need to be taken before the totality of realizations can be observed. To that end, a multi-stage framework is required for a fully non-anticipative ARO implementation in process scheduling. It should be mentioned that an alternative option to handle such multi-stage decision-making settings is the stochastic programming approach, which has also been proposed for addressing process scheduling instances.[38–41] Unlike robust optimization that optimizes in view of the worst-case scenario, stochastic programming optimizes in expectation using scenario trees that are generated from historical data. Apart from this difference in objective focus, both alternatives enforce the feasibility of every scenario albeit in a different manner. Stochastic programming achieves feasibility by explicitly modeling scenario-specific recourse actions, whereas ARO implements recourse via adjusting the wait-and-see decisions on past realizations. In general, using a multi-stage ARO framework would be advantageous in cases where detailed statistical information is not available, or in cases where the scenario trees become prohibitively large.

Multi-stage ARO indeed offers a number of benefits over the traditional SRO approaches as well. For one, it can result in robust solutions that are less conservative; that is, solutions that are more profitable in the worst-case while insuring for the same, chosen level of risk (uncertainty set). In addition, since the ARO solutions adjust to the actual uncertainty realizations, they can capitalize on realizations that are more favorable than the worst-case scenario and, since the latter is often selected very conservatively and/or is often very unlikely, ARO solutions are expected to yield a much better objective value on average. Finally, an ARO approach offers an increased flexibility to study problem settings that SRO approaches could not accommodate due to their inherent flexibility limitations. For the process scheduling application, in particular, our proposed ARO approach can handle instances with zero-wait states as well as instances that involve uncertainty in process yields and/or utility consumption rates. Such settings could not be addressed by any of the approaches previously proposed in the literature. It should be highlighted that, although the ARO framework produces multi-stage solutions, it requires only a single optimization run to do so, just like the SRO approach. This feature of the ARO method does not of course preclude its application in a rolling (or receding) horizon approach, where re-optimization is conducted when new information becomes available. In both cases, the robustness of the solution will be guaranteed. In general, an ARO approach is compatible to be fielded in any decision-making setup in which an SRO approach is also applicable.

The novel contributions of our article can be summarized as follows:

- We derive an affinely-ARO model for process scheduling, which encodes multi-stage robust solutions that remain feasible for any uncertainty realization from within some chosen, polyhedral uncertainty set. The resulting optimal solution policies take advantage of the wait-and-see nature of the problem's decision variables, allowing the latter to adapt to the realizations of the uncertain parameters, while ensuring that all applicable non-anticipativity and observability restrictions are obeyed.
- We introduce in this context the use of decision-dependent uncertainty sets, which can address the endogenous uncertainty that is inherent in many process scheduling applications. An additional novel feature in our uncertainty descriptions is that we allow the possibility for independent realization of parameters that are associated with recurring occurrences of the same task.
- We conduct a comprehensive computational study to elucidate the numerical tractability of the proposed approach and to quantify the superiority of the ARO solutions as compared to the traditional SRO solutions. In addition to improving the best known robust solutions for most literature benchmark problem instances we considered, this study also obtains for the first time robust solutions of instances that involve zero-wait states.

The remainder of the article is organized as follows. First, we discuss alternatives for models that have been proposed to address the process scheduling problem deterministically, and we motivate our choice of a model that can serve as the basis for addressing the problem under uncertainty via the ARO approach. We then formalize the description of uncertainty in our context, and we discuss the derivation of the adjustable robust counterpart of the chosen deterministic model. Finally, we present a comprehensive computational study to elucidate the gains that the proposed ARO approach provides in terms of solution quality, its numerical tractability, the impact of various algorithmic choices, as well the trade-off between solution robustness and objective value in the context of scheduling problems.

Deterministic Process Scheduling

In a typical instance of short-term, batch process scheduling, there exist multiple processing units that can perform a variety of tasks, each of which consumes and produces materials (a.k.a., states) according to an applicable production recipe. The processing units are characterized by a range of batch sizes that they can accommodate, as well as the processing times and rate of utilities consumption—both fixed and variable (depending on the batch size)—that they require for each compatible task. The states can be stored in storage units with given capacities and initial levels, while demands and prices can also be defined for each state. The goal is then to determine the optimal (a) allocation, sequencing and timing of the processing tasks, as well as (b) the batch size for each task, such that the demand is met in the most expeditious manner (minimization of makespan), or the plant is utilized in the most profitable manner across a time horizon (maximization of profit).

A testament to the importance of this application is the large number of models that have been proposed in the literature to mathematically describe the problem under deterministic conditions. These models can be broadly classified into two major categories according to the manner in which they account for the timing of tasks to be scheduled. Discrete-time models[42,43] discretize the time into a (typically equally-spaced) time grid and allow tasks to occur only at one of these time points. These models offer simplicity of representation, allowing the modeler to accommodate a multitude of application-specific restrictions, but they typically require a large number of grid points (and, hence, variables and constraints) to capture the optimal solution with sufficient time precision. Conversely, continuous-time models utilize continuous variables to model explicitly the times at which tasks are to be scheduled, allowing for higher time precision and leading to a significant reduction in model size, albeit at the cost of a more complex mathematical representation that may hinder in some cases the ability of the modeler to extend the model toward new settings. The continuous-time models can be further classified into two subcategories, namely single-grid models,[44–51] and multi-grid models,[52–61] depending on whether they employ a common or a unit-specific grid to keep track of task timing. The size and computational tractability of each model may vary across different problems and these aspects have been extensively studied in the deterministic literature.[62,63]

It should be noted that all prior literature that applied SRO to the process scheduling problem utilized continuous-time models. Indeed, having the flexibility to decide the timing of task execution on an as fine as necessary, continuous sense, increases the quality of the robust solutions and avoids unnecessary conservatism by having to execute a task at statically predefined time points. In particular, all SRO approaches to-date have been based on unit-specific event-based models. However, although this can be viewed as a good choice for the case of static robust scheduling, resulting in relevantly compact sizes for the robust counterpart model and to low computational times required to solve the benchmark instances, unit-specific models exhibit some limitations in the scope of adjustable robust scheduling. More specifically, accounting for task timing in a unit-specific manner does not readily accommodate global observability, that is, when one must decide on a variable associated with a unit, one may not readily know the state of task execution in the other units and, hence, not be in position to inform the decision based on that. Therefore, without major modifications to the model formulation—that would essentially "globalize" the event points-, unit-specific event-based models do not allow the modeler to harness the full flexibility and promise of an adjustable robust framework. Conversely, in global event-based models, the unique definition of the event points across all of the processing units directly allows the adjustment of a variable against any available information about realization of uncertainty, irrespectively of whether this information has to cross unit boundaries.

To that end, we have chosen to utilize the global event-based model by Castro et al.[47] so as to showcase our proposed ARO framework. Although, in principle, any global event-based model can be used as the basis for an ARO framework, resulting into equivalent ARO counterpart models that provide the same solutions, our specific choice is further motivated by this model's inherent suitability to also provide traditional SRO solutions readily. This is true because the model by Castro et al.[47] consists of fewer equality constraints that involve potentially uncertain parameters, which are inherently incompatible with the paradigm of SRO (whereby all decisions have to be taken here-and-now). Although such limitations are alleviated by our proposed ARO framework, utilizing a model that also admits readily the SRO framework constitutes a better basis for the comparative study that we present later in our numerical studies.

Deterministic formulation

Below we present the global event-based, continuous-time, deterministic process scheduling model that was first introduced by Castro et al.[47]. Note that the formulation presented below may differ slightly from that in the original publication due to notational changes and small modeling modifications that we have performed for convenience of presentation. However, it should be highlighted that it is mathematically equivalent and identical in spirit with that found in Castro et al.[47]. The notation used in our formulation is listed at the end of this manuscript. Note that the formulation assumes that each task is compatible with at most one equipment unit; such "task-splitting," as commonly referred to in the process scheduling literature, can be achieved by duplicating appropriately those tasks that are compatible with more than one equipment unit during an input-data preprocessing step.

$$\min_{\substack{W,T,B,\\U,S,G,z}} \quad z \tag{1}$$

$$\text{s.t.} \quad z \geq \sum_{s\in\mathcal{S}} P_s\left(S_{s0}-S_{sN}\right) \tag{2a}$$

$$z \geq T_N - T_1 \tag{2b}$$

$$T_{n'} - T_n \geq \sum_{i\in\mathcal{I}_j} \left(\alpha_i W_{inn'} + \beta_i B_{inn'}\right) \qquad \forall\, j\in\mathcal{J},\ \forall\, n\in\mathcal{N},\ \forall\, n'\in\mathcal{N}_n^+ \tag{3}$$

$$W_{inn'} B_i^{min} \leq B_{inn'} \leq W_{inn'} B_i^{max} \qquad \forall\, j\in\mathcal{J},\ \forall\, n\in\mathcal{N},\ \forall\, n'\in\mathcal{N}_n^+ \tag{4}$$

$$G_{jn} = G_{j(n-1)} + \sum_{i\in\mathcal{I}_j}\left[\sum_{n'\in\mathcal{N}_n^+} W_{inn'} - \sum_{n'\in\mathcal{N}_n^-} W_{in'n}\right] \qquad \forall\, j\in\mathcal{J},\ \forall\, n\in\mathcal{N}:\{n>1\} \tag{5}$$

$$S_{sn} = S_{s(n-1)} + \sum_{i\in\mathcal{I}_s^p} \rho_{is} \sum_{n'\in\mathcal{N}_n^-} B_{in'n} - \sum_{i\in\mathcal{I}_s^c} \rho_{is} \sum_{n'\in\mathcal{N}_n^+} B_{inn'} \qquad \forall\, s\in\mathcal{S},\ \forall\, n\in\mathcal{N} \tag{6}$$

$$U_{un} = U_{u(n-1)} + \sum_{i\in\mathcal{I}_u}\left[\sum_{n'\in\mathcal{N}_n^+}\left(\gamma_{iu} W_{inn'} + \delta_{iu} B_{inn'}\right) - \sum_{n'\in\mathcal{N}_n^-}\left(\gamma_{iu} W_{in'n} + \delta_{iu} B_{in'n}\right)\right] \qquad \forall\, u\in\mathcal{U},\ \forall\, n\in\mathcal{N}:\{n>1\} \tag{7}$$

$$T_{n'} - T_n \leq \overline{M}\left[1 - \sum_{i\in\mathcal{I}_j\cap\mathcal{I}^{zw}} W_{inn'}\right] + \sum_{i\in\mathcal{I}_j}\left(\alpha_i W_{inn'} + \beta_i B_{inn'}\right) \qquad \forall\, j\in\mathcal{J},\ \forall\, n\in\mathcal{N},\ \forall\, n'\in\mathcal{N}_n^+ \tag{8}$$

$$\sum_{i\in\mathcal{I}_j} \sum_{\substack{n'\in\mathcal{N}:\\ \{n'\geq n\}}} \sum_{n''\in\mathcal{N}_{n'}^+} \left(\alpha_i W_{in'n''} + \beta_i B_{in'n''}\right) \leq T_N - T_n \qquad \forall\, j\in\mathcal{J},\ \forall\, n\in\mathcal{N} \tag{9}$$

$$T_N = H \tag{10a}$$

$$S_{sN} \geq D_s \qquad \forall\, s\in\mathcal{S} \tag{10b}$$

$$0 \leq G_{jn} \leq 1 \qquad \forall\, j\in\mathcal{J},\ \forall\, n\in\mathcal{N} \tag{11}$$

$$0 \leq S_{sn} \leq S_s^{max} \qquad \forall\, s\in\mathcal{S},\ \forall\, n\in\mathcal{N} \tag{12}$$

$$0 \leq U_{un} \leq U_u^{max} \qquad \forall\, u\in\mathcal{U},\ \forall\, n\in\mathcal{N} \tag{13}$$

$$W_{inn'} = 0 \qquad \forall\, i\in\mathcal{I},\ \forall\, n\in\mathcal{N},\ \forall\, n'\in\mathcal{N}\setminus\mathcal{N}_n^+ \tag{14}$$

$$B_{inn'} = 0 \qquad \forall\, i\in\mathcal{I},\ \forall\, n\in\mathcal{N},\ \forall\, n'\in\mathcal{N}\setminus\mathcal{N}_n^+ \tag{15}$$

$$T_1 = 0 \tag{16}$$

$$G_{jN} = 0 \qquad \forall\, j\in\mathcal{J} \tag{17}$$

$$W_{inn'} \in \{0,1\} \qquad \forall\, i\in\mathcal{I},\ \forall\, n\in\mathcal{N},\ \forall\, n'\in\mathcal{N} \tag{18}$$

Equation 1 is the objective to be minimized, where for convenience we have introduced an auxiliary, unconstrained, continuous variable z to capture its value. Equation 2a, which is to be used for the case of profit maximization, then sets this variable to be equal to the difference between the monetary value of all states in the beginning and the end of the scheduling horizon (i.e., cost of raw materials used minus revenue of final products produced). For the case of makespan minimization, Eq. 2b is to be used instead, setting the value of variable z to be equal to the difference between the final event point at which a task shall finish and the earliest event point at which a task shall begin (i.e., the total makespan). Equations 3 enforce the timing of the occurring tasks by securing that the time difference between any consecutive pair of event points is no smaller than the required duration of any tasks that start and finish at this pair of event points. Equations 4 represent constraints that are enforcing upper and lower bounds on batch sizes when a task is scheduled. These constraints are appropriately relaxed when a task is not selected. Equations 5 track the utilization of the processing units, Eq. 6 constitute material balances, while Eq. 7 constitute utility balances in each specific event point. Equations 8 are used whenever zero-wait states apply, whereby the equipment unit has to discharge the material immediately after the end of its processing.* Equations 9 constitute tightening constraints, which are not

*The coefficient $\bar{M}$ serves here as an appropriately large "big-M" value. For the case of profit maximization, it suffices to use $\bar{M}=H$, the problem's time horizon.

necessary for completeness of the model but are typically included so as to improve its linear programming (LP) relaxation and overall solution time. Equation 10a sets the time horizon for the schedule (only for the case of profit maximization), while Eq. 10b ensure that the product demands are met by the end of the schedule (only for the case of makespan minimization). Equations 11 constitute explicit bounds for variables G_{jn} and, combined with Eq. 5, enforce that no more than one task can be executed in a unit at any given moment in time. Bounds on state amounts (storage capacities) and on utility consumptions (utility availability) are enforced by Eqs. 12 and 13, respectively. Equations 14 and 15 disallow tasks whose start time is past their end time, tasks that are instantaneous, as well as tasks that last more than δn event points.† Equation 16 fixes the first event point at the reference time of zero. Equation 17 dictates that all units need to be released at the end of the schedule (equivalently, every task that started must be finished within the scheduling horizon). Finally, Eq. 18 express the integrality restrictions for the model's binary variables. This formulation is of mixed-integer linear programming (MILP) form and can be readily addressed with standard MILP optimization solvers.

Process Scheduling Under Uncertainty

In the context of optimizing process scheduling instances, a multitude of parameters can be uncertain at the moment the operator has to make decisions and commit to a plan about how to operate the plant. Leaving aside a number of highly invariant parameters, such as material storage capacities, as well parameters that constitute explicit choices of the modeler, such as the length of the horizon, H, and the maximum task span, δn, the remaining parameters could exhibit a significant level of uncertainty against which the operator may want to insure. In general, task processing times, consumption and availability of utilities, and process yields can all be uncertain. Furthermore, in cases where the time horizon is sufficiently long, material prices, and product demands can be as well. Uncertainty in these parameters, if not accounted for, may give rise to infeasibilities during the execution of the schedule. For example, in the case where uncertainty in processing time parameters exists, their direct involvement in the timing constraints (Eq. 3) can lead to task overlaps, violating in this way the unit availability restrictions (Eq. 5). At the same time, the interaction with the mass balance constraints (Eq. 6) can also lead to violation of the constraints for demand satisfaction (Eq. 10b) or storage tank capacities (Eq. 12). Similar arguments can be made for uncertainty in all other parameters. In this section, we discuss the development of an ARO optimization framework that will allow us to optimize the process scheduling instance of interest in view of uncertainty in model parameters.

Parameter time-dependence

It is noteworthy that, in the deterministic model presented above as well as in all deterministic models that have been presented in the scheduling literature to-date, the various model parameters do not depend on time; that is, parameters are never indexed over the set of event points $n \in \mathcal{N}$. This practice implies that these parameters are expected to remain constant throughout the horizon and that identical values for these parameters apply when related tasks occur more than once in the optimal schedule. Although this may be a reasonable—and, if one so desires, easy to circumvent—assumption when one addresses the problem from a deterministic viewpoint, it does not do justice to the breadth of settings that may be encountered in real-life instances, whereby repeated parameter realizations may vary due to time-dependent factors that affect them. Interestingly, this issue has not been addressed by existing literature applying SRO in the context of process scheduling optimization. Despite the fact that these studies indeed take into account uncertainty in model parameters, they all make an implicit assumption that parameter realizations are identical whenever they occur more than once during the horizon, an assumption that can be quite optimistic in certain cases. Furthermore, this assumption gives rise to a couple of significant numerical implications. For one, the modeler only has to insure against the few parameters originally present in the deterministic model. This leads to relatively small robust counterpart formulations, which of course appears desirable, but in essence provides a false perspective regarding the true numerical tractability of an uncertainty-aware framework in this context. Conversely, the obtained solutions would not guarantee robustness against very reasonable uncertainty realizations in which the parameters realized to different values during each recurring realization (e.g., a task lasted 30 min when it took place in the morning, while the same task lasted 32 min when it was repeated in the afternoon).

To move past this limitation of existing methods, we propose here a more realistic setup where each parameter of interest is amended with index n, which is to be viewed as the realization of the original parameter that realized at event point n. This simple model extension readily allows for a unique realization depending on the exact timing of realization. During the quantitative description of uncertainty, the modeler can assign the same nominal value and bounds to each one of these augmented parameters, while temporal correlations among the parameters stemming from the same original parameter may be accounted for, as appropriate. In principle, this notational extension can be applied to the entirety of the model's uncertain parameters. Although doing so constitutes a better representation of reality, it should be highlighted that it also results in instances with a much larger number of uncertain parameters and, hence, into more complex and computationally-intensive robust counterpart models.

Uncertainty set

As discussed earlier, our ARO framework shall support any generic polyhedral description of uncertainty. Such descriptions offer a good trade-off between realism of representation and computational tractability, as has also been demonstrated in previous SRO approaches for process scheduling. In any case, it should be noted that the polyhedral description is not very limiting in practice, as it can accommodate any known affine correlation among problem parameters. Notable examples of such correlations are aggregate forecasts (possibly expressed in various hierarchical levels) as well as factor models. The latter have been used successfully in the past in the context of financial portfolios[21] and distribution operations,[64] and they constitute a plausible uncertainty description for cases when the uncertainty dimension (i.e., the number of uncertain parameters) is large. In addition, note that more involved uncertainty descriptions (e.g., ellipsoidal sets

†Parameter δn is imposed by the modeler as an explicit upper bound on the event-point span of every task. In this article, we choose δn depending on the defined number of event points, N, and according to the following rule of thumb: if $N \leq 5$, then δn=2; if $6 \leq N \leq 8$, then δn=3; if $N \geq 9$, then δn=4.

stemming from normal distributions) can also be handled via efficient polyhedral approximation methods.[65]

We stress that our proposed ARO framework can handle uncertainty in all model parameters, including left-hand-side, right-hand-side, and objective function coefficients. The only exception lies with parameters that directly multiply in the deterministic model those continuous variables on which we have decided to adjust (the possible implications of this exception are minor, but will be addressed later). For ease of exposition, however, in the rest of this article we will consider uncertainty only in the processing-time parameters α_i, which we shall expand toward time-dependent (as per the above discussion) parameters α_{in}. Here, parameter α_{in} shall represent the processing time of the task i that finishes in the time interval $(T_{n-1}, T_n]$ or, equivalently, the processing time of the task that is associated with binary variables $W_{in'n}$, where $n\prime \in \mathcal{N}_n^-$ (note that only one such binary variable may be active at any solution).

We consider the uncertainty set of admissible parameter realizations as the polyhedron that results from the intersection of a set of general affine correlations involving those realizations with a hyperbox that limits the realization of each uncertain parameter between some applicable lower and upper bound. More formally, the uncertainty set can be expressed with Eq. 19.

$$\mathcal{A}=\left\{\alpha \in \mathbb{R}_+^{|\mathcal{I}||\mathcal{N}|} : \begin{array}{ll} \sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}} h_{fin}\alpha_{in} \leq g_f & \forall\, f\in\mathcal{F} \\ \alpha_{in}^{lb} \leq \alpha_{in} \leq \alpha_{in}^{ub} & \forall\, i\in\mathcal{I},\ \forall\, n\in\mathcal{N} \end{array}\right\}, \tag{19}$$

where $\mathcal{F}=\{1,2,\cdots,F\}$ is the set of general affine correlations, h_{fin} and g_f are the coefficients defining the fth correlation of this set, while α_{in}^{lb} and α_{in}^{ub} represent the lower and upper bounds, respectively, for each uncertain parameter realization.

It should be noted that, in process scheduling models, not all parameters play a role in the context of all possible solutions. For example, consider the parameter α_{in} for some i and n. If the equality $\sum_{n'\in\mathcal{N}_n^-} W_{in'n}=0$ holds at the optimal solution, that is, if the optimizer does not pick a solution in which a task of type i concludes in the time interval $(T_{n-1}, T_n]$, the parameter in question vanishes from the model and its realization would never be revealed to us. In other words, had we known that the above equality would hold at the optimal solution, we would not even have to define the associated parameter in the model, let alone consider it as one of our uncertain parameters. Handling uncertainty in such parameters, which in the field of stochastic programming has been referred to as *endogenous* uncertainty,[66–69] requires some careful treatment in the context of a robust optimization framework where, unlike when one addresses the problem deterministically, a non-materialized parameter may still affect the optimal solution implicitly via its referencing in the considered uncertainty set. In fact, given the basic principle of RO to immunize the solution against any realization of the uncertainty within the chosen uncertainty set, imposing a correlation involving such a parameter may expand the admissible set of realizations of the truly materialized parameters and, in turn, require a more conservative solution to ensure robustness against this expanded set of realizations. Furthermore, the very referencing of a non-materialized parameter in a correlation provided by the modeler may not be motivated in practice, as a possible realization of such a parameter may not retain any physical meaning in this case.

To provide a framework that alleviates these unwanted implications, we generalize the description of the uncertainty set into a set that can adapt around the actual subset of uncertain parameters that do materialize in a solution. More specifically, we extend the static uncertainty set of Eq. 19, which is invariant of the decisions taken, into the decision-dependent uncertainty set of Eq. 20. We remark that the more general, decision-dependent form of a polyhedral uncertainty set is considered here without loss of generality, and that the ARO framework we present in this article applies readily with non-decision-dependent sets as well. Therefore, as and when appropriate, a modeler may choose to replace the variable expressions with constant terms, reverting all or part of the uncertainty set into the special case of Eq. 19.

$$\mathcal{A}(W)=\left\{\alpha \in \mathbb{R}_+^{|\mathcal{I}||\mathcal{N}|} : \begin{array}{ll} \sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}} h_{fin}\left(\sum_{n'\in\mathcal{N}_n^-} W_{in'n}\right)\alpha_{in} \leq \sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}} g_{fin}\left(\sum_{n'\in\mathcal{N}_n^-} W_{in'n}\right) & \forall\, f\in\mathcal{F} \\ \alpha_{in}^{lb} \leq \alpha_{in} \leq \alpha_{in}^{ub} & \forall\, i\in\mathcal{I},\ \forall\, n\in\mathcal{N} \end{array}\right\}, \tag{20}$$

where each original coefficient g_f has been split into a set of coefficients g_{fin}, for all $i\in\mathcal{I}$ and $n\in\mathcal{N}$, according to the contribution of each correlated uncertain parameter in the correlation's right-hand-side, and where the sums $\sum_{n\prime\in\mathcal{N}_n^-} W_{in'n}$ constitute the binary terms that indicate whether corresponding parameters α_{in} are materialized or not in the optimal solution.

To better illustrate the concept of a decision-dependent uncertainty set, let us consider the case of three uncertain parameters, $\{\alpha_1,\alpha_2,\alpha_3\}$, with nominal values $\{\alpha_1^0,\alpha_2^0,\alpha_3^0\}$, given lower and upper bounds at $\pm 20\%$ of their nominal values, as well as a single decision-dependent correlation of the form $W_1\alpha_1+W_2\alpha_2+W_3\alpha_3 \leq 110\%\left(W_1\alpha_1^0+W_2\alpha_2^0+W_3\alpha_3^0\right)$. Here, it is implied that the materialization of a parameter α_k is indicated by some corresponding binary variable W_k. This affine correlation, which can be interpreted as "those tasks that take place would not cumulatively exceed 10% of their nominal processing times," constitutes a realistic correlation that the modeler may want to postulate. Given this setting, in a solution where $\{W_1,W_2,W_3\}=\{1,1,1\}$, all three parameters materialize and the uncertainty set considered will be as depicted in Figure 1a. However, in a solution where only parameters α_1 and α_2 materialize due to $\{W_1,W_2,W_3\}=\{1,1,0\}$, the uncertainty set will adapt to the one depicted in Figure 1c. In this case, the uncertainty set is not affected by the realization of the parameter α_3, effectively

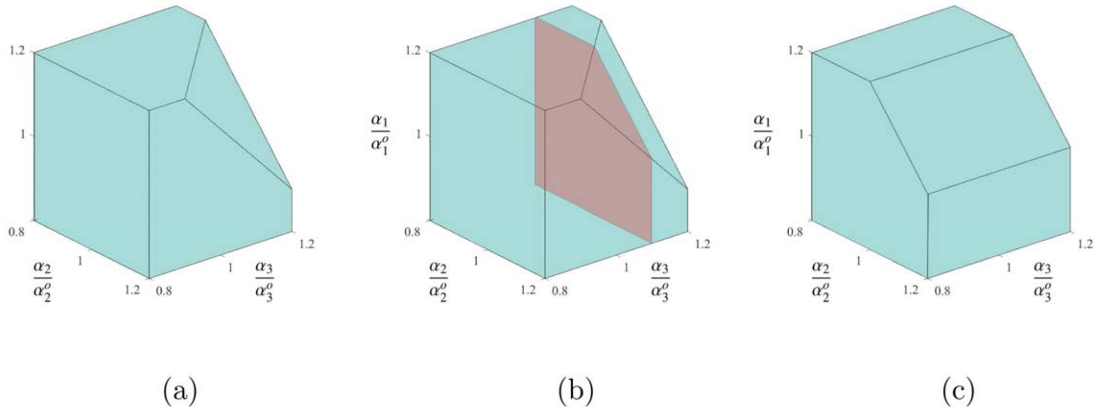


Figure 1. Illustration of a decision-dependent uncertainty set: (a) uncertainty set when all three uncertain parameters materialize; (b) realizations that matter when only two uncertain parameters materialize; and (c) uncertainty set when only two uncertain parameters materialize.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

immunizing against a subset of the original set of admissible realizations (see magenta-colored polygon in Figure 1b).

Adjustment of decisions

The core idea behind ARO is to adjust all, or some subset of, the continuous decision variables based on the actual realizations of the uncertain parameters, as soon as the latter are observed in the plant. To that end, a very significant choice has to be made by the modeler with respect to which of the problem's original continuous variables to select as adjustable. As already mentioned, the proposed ARO framework facilitates the adjustment of any continuous variables as long as those are not directly multiplied in the deterministic model by an uncertain parameter. For the process scheduling model used in this study, this translates to a hierarchy of options. In all cases, the timing variables, T_n, can be selected as adjustable variables, since they only participate in the deterministic model with constant coefficients. The same holds true for utility consumption variables, U_{un}, which can be adjusted in all cases, while state inventory variables, S_{sn}, can be adjusted as long as the material prices, P_s (relevant only for profit maximization), are not considered as uncertain. Indeed, adjusting variables T_n, U_{un}, and S_{sn} is especially important when there exists uncertainty in processing times, utility consumption rates, and production yields, respectively. Furthermore, as long as the parameters β_i, δ_{iu}, ρ_{is}, and P_s are not considered uncertain, one may further choose to adjust the batch-size variables, $B_{inn'}$, as well. Note that, when one chooses to adjust variables $B_{inn'}$, one should also adjust variables S_{sn} and U_{un}, since these are effectively auxiliary, "state" variables that directly depend on the batch sizes. Finally, note that uncertainty in parameters that only multiply binary variables, such as parameters α_i and γ_{iu}, or uncertainty in parameters that only participate in the model as right-hand-sides, such as parameters D_s, can be accommodated always, irrespectively of the choice of adjustable variables. In any case, as it will demonstrated later in our computational study, adjusting only on timing variables T_n (which allows us to consider uncertainty in all types of parameters of the process scheduling problem) provides a favorable trade-off between optimality and computational tractability.

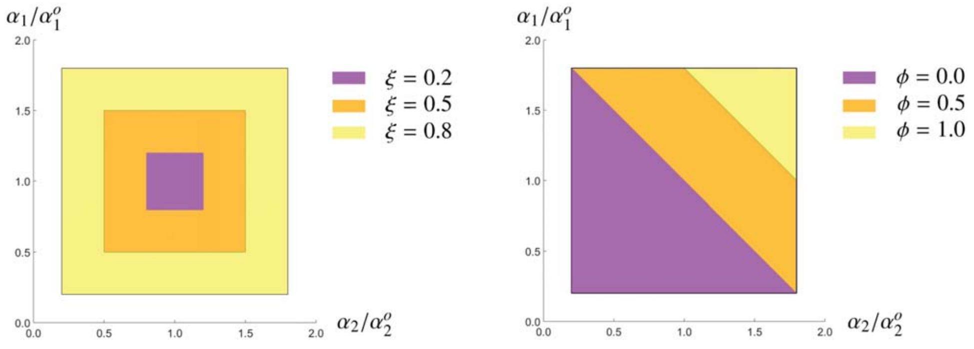
Affine Decision Rules. To enable the adjustment of a decision according to parameter realizations, the functional relationship between the corresponding decision variable and the applicable parameter realizations must be postulated by the modeler. Although the "best" relationship to use in each case is typically unbeknown to the modeler, the use of an affine relationship (a.k.a., an "affine decision rule") is typically used, and we will follow this practice in this article. As discussed by Bertsimas et al.[17], such a choice offers the advantage of tractability of the robust counterpart model, which can thus retain its original MILP structure, while still offering acceptable performance in most applications. For even better adjustment quality, piecewise-affine[34] or polynomial relationships[70] may also be used, albeit at a cost on computational tractability.

For ease of exposition, we focus on a timing variable, T_n, and present below how this variable can be adjusted on the realizations of our uncertain parameters, namely the task processing times, $\alpha_{in'}$, using affine decision rules. The adjustment of whichever other continuous variable has been selected to this purpose can be achieved in a similar fashion. Equation 21 presents the applicable affine relationship. In this equation, $[T_n]_0$ and $[T_n]_{in'}$ are new continuous variables, which we refer to as *adjustment variables*. They are introduced so as to encode, respectively, the intercept and the slopes against each uncertain parameter realization $\alpha_{in'}$ (a full gamut of slopes for each $i \in \mathcal{I}$ and $n' \in \mathcal{N}$). Note that each occurrence in the deterministic model of the term T_n is to be replaced by this affine expression of the adjustment variables.

$$T_n \leftarrow [T_n]_0 + \sum_{i \in \mathcal{I}} \sum_{n' \in \mathcal{N}} [T_n]_{in'} \alpha_{in'} \tag{21}$$

Restrictions on adjustment variables

Bounds. The domain of the new variables, $[T_n]_0$ and $[T_n]_{in'}$, is generally considered as unbounded. However, when one declares these variables in the context of a numerical solver, some explicit lower and upper bounds will have to be imposed. The domain of the intercept variable $[T_n]_0$ needs to be at least as inclusive as the original domain of variable T_n, while the domain of each slope variable $[T_n]_{in'}$ needs to include the value of zero; that is, the lower bounds for the slope variables should be non-positive, while the upper bounds should be non-negative. The bounds for the slopes $[T_n]_{in'}$, in particular, can be viewed as the "magnitude" of adjustment, whereby setting these bounds (sufficiently close) to $\pm\infty$ corresponds to full adjustment, while setting them to 0 (i.e., fixing these



(a) (b)

Figure 2. Effect of (a) magnitude ξ, for constant ϕ=1.0, and of (b) correlation strength ϕ, for constant ξ=0.8, on the shape and size of the uncertainty set.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

variables to 0) corresponds to no adjustment. Evidently, the ARO framework recovers the SRO framework when all slope variables $[T_n]_{in'}$ are fixed to zero, at which point the intercept variable $[T_n]_0$ represents a mere notational change for the original variable T_n. It should be highlighted that the robustness of the ARO solution is not affected by the exact choice of bounds for the new continuous variables. A more restrictive choice only affects the framework's capacity to admit solutions that are considerably less conservative than their corresponding SRO counterparts.

Non-Anticipativity. Certain restrictions on the domain of the slope variables $[T_n]_{in'}$ need to be imposed so as to ensure that the resulting ARO solution respects the non-anticipativity principle and, thus, can be implemented in practice. The non-anticipativity principle simply expresses the fact that one may not utilize the value of some uncertain parameter realization that has not yet occurred so as to adjust a decision that one has to make at the present moment. In mathematical modeling terms, disallowing the adjustment of a decision on future realizations of uncertain parameters can be achieved by fixing to zero all slope variables multiplying realizations of uncertain parameters that correspond to event points that succeed the event point of the applicable decision. More specifically, for the continuous decision T_n, which is to be taken at the nth event point, we need to fix to zero all variables $[T_n]_{in'}$, such that $n' > n$; that is, we need to apply Eq. 22 in our model.

$$[T_n]_{in'}=0 \quad \forall\ i \in \mathcal{I},\ \forall\ n' \in \mathcal{N} : n' > n \tag{22}$$

Observability. Equations 22 suffice to ensure that no adjustment of decision variables on future realizations takes place and that adjustment can take place only on past realizations. However, further restrictions on the domain of the slopes $[T]_{in'}$ are necessary so as to ensure that adjustment of decision variables on past realizations is allowed only when those parameters have indeed materialized and, thus, can be observed and recorded by the plant operator. This can be achieved by fixing the corresponding slope variables on a conditional basis, according to Eq. 23.[‡] Note that the latter constitute implication constraints. Such constraints need to be typically reformulated via standard "big-M" techniques before they are incorporated in the overall MILP model. However, we note that many modern MILP solvers can nowadays accommodate these constraints directly in their native, implication form, which they treat via use of SOS1 techniques.

$$\sum_{n'' \in \mathcal{N}_{n'}^-} W_{in''n'}=0 \ \Rightarrow\ [T_n]_{in'}=0 \quad \forall\ i \in \mathcal{I},\ \forall\ n' \in \mathcal{N} : n' \le n \tag{23}$$

Adjustable robust model

In this section, we describe the derivation of an adjustable robust model that is a counterpart of the deterministic model we presented earlier. Note that, just like its deterministic predecessor, the adjustable robust model is also of MILP type and can thus be addressed directly via an MILP optimization solver. To facilitate the description, we partition the set of original constraints of the deterministic model into three subsets, which we refer to as constraint sets $\mathcal{M}_I$, $\mathcal{M}_E$, and $\mathcal{M}_C$. The set $\mathcal{M}_I$ shall contain all inequality constraints that make reference to at least one uncertain parameter or at least one variable that has been selected for adjustment. Similarly, the set $\mathcal{M}_E$ shall contain all equality constraints that make reference to at least one uncertain parameter or at least one adjustable variable. Finally, the set $\mathcal{M}_C$ shall constrain the remaining constraints, namely those inequalities and equalities that do not make any reference to either uncertain parameters or adjustable variables. For example, in our expository case of considering uncertainty only in parameters α_{in} and having selected for adjustment all variables T_n,[§] and for the case of the profit maximization objective, the applicable partition of the model will be as follows: $\mathcal{M}_I=\{(3),(8),\ (9)\}$, $\mathcal{M}_E=\{\}$, $\mathcal{M}_C=\{(2a),(4)\text{–}(7),(10a),(11)-(18)\}$. For comparison, in the case where all continuous variables T_n, $B_{inn'}$, S_{sn}, and U_{un} are allowed to be adjusted,[¶] then the partition will be as follows: $\mathcal{M}_I=\{(2a),(3),(4),(8),(9),(12),(13)\}$, $\mathcal{M}_E=\{(6),(7)\}$, $\mathcal{M}_C=\{(5),(10a),(11),(14)\text{–}(18)\}$.

Given a choice of adjustable variables and a choice of a decision-dependent uncertainty set, $\mathcal{A}(W)$ (as in Eq. 20), we discuss below how the deterministic constraints must be reformulated so as to derive the ARO counterpart model. For ease of exposition, we will present the applicable reformulations in the context of one representative constraint from each of the

[‡]Without loss of generality, it is assumed here that the value of an uncertain parameter that realizes at the n^{th} event point (e.g., α_{in}) is available to the operator immediately and, thus, can be used to inform a decision taken at the very same event point (e.g., T_n).

[§]Variables T_0 and T_N can be exempted from adjustment, as they will attain fixed values dictated by the model.

[¶]Variables T_0, T_N, and $B_{inn'}$, $\forall\ i \in \mathcal{I},\ \forall\ n \in \mathcal{N},\ \forall\ n' \in \backslash \mathcal{N}\mathcal{N}_n^+$, can be exempted.

sets $\mathcal{M}_I$ and $\mathcal{M}_E$. The reformulation of the remaining constraints in these sets is to be performed in a similar fashion. With regards to the set $\mathcal{M}_C$, we remark that no reformulation is necessary for these constraints. Instead, these constraints should be directly carried over to the ARO model. Finally, we note that the adjustable robust model must also include all applicable adjustment restrictions of the type shown in Eqs. 22 and 23, for each and every variable that has been selected for adjustment.

Reformulation of Constraints $\mathcal{M}_I$. Consider a timing constraint (Eq. 3) for some $j \in \mathcal{J}$, $n \in \mathcal{N}$, and some $n' \in \mathcal{N}_n^+$, and assume that all variables T_n, $T_{n'}$, and $B_{inn'}$ (for all $i \in \mathcal{I}_j$) are selected for adjustment. The adjustable robust counterpart of this constraint is shown in Eq. 24, where variables $s^{jnn'}$, $p^{jnn'}$, $q^{jnn'}$, and $r^{jnn'}$ are auxiliary, continuous, nonnegative variables. We remark that these variables are associated with the specific constraint reformulated here, and that they do not appear anywhere else in the ARO model. The implication constraints are to be handled by the MILP solver via SOS1 indications. The intermediate derivation steps are presented in detail in Appendix A.

$$
\begin{aligned}
&\sum_{i' \in \mathcal{I}} \sum_{n'' \in \mathcal{N}} \left(\sum_{f \in \mathcal{F}} g_{fi'n''} s_{fi'n''}^{jnn'} - \alpha_{i'n''}^{lb} q_{i'n''}^{jnn'} + \alpha_{i'n''}^{ub} r_{i'n''}^{jnn'} \right) \leq [T_{n'}]_0 - [T_n]_0 - \sum_{i \in \mathcal{I}_j} \beta_i [B_{inn'}]_0 \\
&\left.\begin{aligned}
&\sum_{f \in \mathcal{F}} h_{fi'n''} s_{fi'n''}^{jnn'} - q_{i'n''}^{jnn'} + r_{i'n''}^{jnn'} \geq \\
&1_{\{i' \in \mathcal{I}_j\}} 1_{\{n''=n\}} W_{i'n''n'} - [T_{n'}]_{i'n''} + [T_n]_{i'n''} + \sum_{i \in \mathcal{I}_j} \beta_i [B_{inn'}]_{i'n''} \\
&\left.\begin{aligned}
&\sum_{n''' \in \mathcal{N}_{n''}^-} W_{i'n'''n''} = 1 \Rightarrow s_{fi'n''}^{jnn'} \geq p_f^{jnn'} \\
&\sum_{n''' \in \mathcal{N}_{n''}^-} W_{i'n'''n''} = 0 \Rightarrow s_{fi'n''}^{jnn'} \leq 0 \\
&s_{fi'n''}^{jnn'} \leq p_f^{jnn'}
\end{aligned}\right\} \quad \forall\, f \in \mathcal{F}
\end{aligned}\right\} \quad \forall\ i' \in \mathcal{I},\ \forall\ n'' \in \mathcal{N}
\end{aligned}
\tag{24}
$$

Reformulation of Constraints $\mathcal{M}_E$. Consider a material balance constraint (Eq. 6) for some $s \in \mathcal{S}$ and some $n \in \mathcal{N}$, and assume that both variables S_{sn} and $S_{s(n-1)}$ as well as all batch-size variables that appear in that constraint are selected for adjustment. The adjustable robust counterpart of this constraint is shown in Eq. 25. The intermediate derivation steps are presented in detail in Appendix A.

$$
\begin{aligned}
&[S_{sn}]_0 = \left[S_{s(n-1)}\right]_0 + \sum_{i \in \mathcal{I}_s^p} \rho_{is} \sum_{n' \in \mathcal{N}_n^-} [B_{in'n}]_0 - \sum_{i \in \mathcal{I}_s^c} \rho_{is} \sum_{n' \in \mathcal{N}_n^+} [B_{inn'}]_0 \\
&[S_{sn}]_{i'n''} = \left[S_{s(n-1)}\right]_{i'n''} + \sum_{i \in \mathcal{I}_s^p} \rho_{is} \sum_{n' \in \mathcal{N}_n^-} [B_{in'n}]_{i'n''} \\
&- \sum_{i \in \mathcal{I}_s^c} \rho_{is} \sum_{n' \in \mathcal{N}_n^+} [B_{inn'}]_{i'n''} \quad \forall\ i' \in \mathcal{I},\ \forall\ n'' \in \mathcal{N}
\end{aligned}
\tag{25}
$$

The above set of constraints ensures that the original equality will hold for any realization of the uncertain parameters admitted by the uncertainty set. We remark that such "direct robustification" of an equality constraint constitutes a novelty in the process scheduling literature. To-date, the standard literature practice for process scheduling via robust optimization has been to simply reformulate the original deterministic model so as eliminate all equality constraints that reference uncertain parameters. In addition to being incompatible with many existing scheduling models, this literature practice also exhibits limitations in cases where equalities are explicitly necessary, such as in cases involving zero-wait states. In contrast, the ARO reformulation approach presented in this article can in principle be applied with any existing deterministic model as the basis and, as we demonstrate later in our computational studies, can indeed handle instances that involve zero-wait states.

Computational Results

In this section, we test the performance of the proposed framework on a set of benchmark instances that have been widely used in the process scheduling literature. The applicable data for these problem instances have been compiled in Appendix B of this article, for convenience. The models were implemented and solved using the C++ API of the commercial MILP solver Gurobi 6.0.[71] All computational experiments reported in this article were conducted on a single-thread (limited via appropriate solver option) of an Intel Xeon E5-2680 (2.80 GHz) processor with 4 GB of available RAM. To ensure a fair comparison among different settings, the optimal solution in each case was provided to the solver as an initial incumbent, and the solver's internal heuristic efforts to locate a better upper bound were disabled. By doing so, the reported results truly reflect the computational effort required to prove the optimality of the final solution, and they better indicate the tractability of the considered models. The optimality gap tolerance was set to zero, ensuring that all runs resulted to a guaranteed optimal solution. Apart from the above, all default options of the solver were used.

Uncertainty set specification

As already discussed, in the current work we consider uncertainty in the task processing time parameters, α_{in}, and we allow these parameters to vary in a decision-dependent polyhedron. More specifically, we regard the original α_i-values (the ones provided by the benchmark data and listed in Appendix B) as nominal values α_i^0, which are to apply against every parameter α_{in} (for all $n \in \mathcal{N}$). We then consider the maximum allowable magnitude of each parameter's deviation from its nominal value to be a percentage of the latter (e.g., each processing time shall realize within $\pm 30\%$ of its nominal value).

In addition, we also consider one general affine correlation per each equipment unit to postulate an upper limit on the cumulative processing time deviation among all tasks to be performed on this unit. To that end, the uncertainty set we consider here is similar to those previously utilized in the SRO literature with the generalizations of having split the original processing times α_i into α_{in} and, hence, of having to consider a decision-dependent set. The resulting uncertainty set for our computational experiments, which constitutes a specific instantiation of the polyhedral decision-dependent set shown in Eq. 20, is

$$\mathcal{A}(W)=\left\{\alpha_{in}\in\mathbb{R}_+^{|\mathcal{I}||\mathcal{N}|}:\begin{array}{ll}\displaystyle\sum_{i\in\mathcal{I}_j}\sum_{n\in\mathcal{N}}\left(\sum_{n'\in\mathcal{N}_n^-}W_{in'n}\right)\alpha_{in}\le\sum_{i\in\mathcal{I}_j}\sum_{n\in\mathcal{N}}(1+\xi\phi)\alpha_i^0\left(\sum_{n'\in\mathcal{N}_n^-}W_{in'n}\right) & \forall\ j\in\mathcal{J}\\ (1-\xi)\alpha_i^0\le\alpha_{in}\le(1+\xi)\alpha_i^0 & \forall\ i\in\mathcal{I},\ \forall\ n\in\mathcal{N}\end{array}\right\} \tag{26}$$

where the quantities $\xi\in[0,1]$ and $\phi\in[0,1]$ are used to parameterize the uncertainty set (see Figure 2).

Impact of adjustable variables selection

As previously alluded to, there often exist multiple choices with regards to which variables are to be considered as adjustable, and this selection has important implications on the performance of an ARO framework. The more variables selected for adjustment, the better the optimal robust solutions but, at the same time, the larger the complexity of the resulting models and the more difficult for an MILP solver to identify those solutions. To explore the impact that this selection has on the trade-off between quality of solutions and computational tractability, we choose three different options for variable adjustment and compare the resulting solutions. More specifically, we consider the option where no variable is selected for adjustment, the option where only the timing variables, T_n, are selected, and the option where both the timing variables as well as the variables related to batch sizes, that is, $B_{inn'}$ and S_{sn}, are selected. The first option represents the special case of SRO, while we refer to the latter two options as ARO. For reference, this study was performed on the standard benchmark example originally introduced by Kondili et al.[72] (see Figure in instance P1 of Appendix B) and considers a moderately-sized uncertainty set where $(\xi,\phi)=(0.3,0.5)$.

Figure 3 presents the optimal schedules that resulted from this experiment. Note that all depicted solutions are feasible in the robust sense because they all guarantee that the schedule will remain feasible for any realization within the chosen uncertainty set. The solutions achieve such robustness by appropriately incorporating "time gaps" between the executions of consecutive tasks. The gaps are necessary to accommodate the more unfavorable scenarios we want to insure against, such as scenarios where all processing times happen

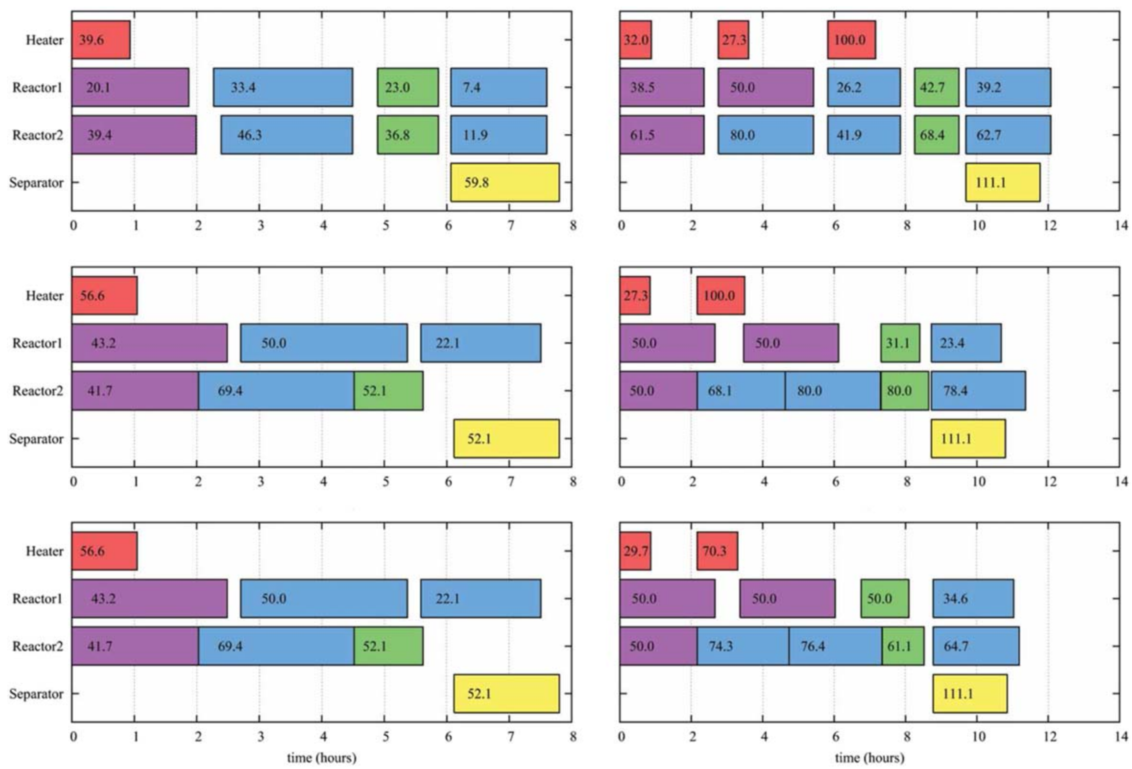


Figure 3. Robust optimal solutions for both objectives, profit maximization (*left*) and makespan minimization (*right*), and for various selections of adjustable variables; *top*: SRO (adjusting no variables); *middle*: ARO (adjusting timing variables only); *bottom*: ARO (adjusting timing and batch-size-related variables).

The numbers in each box represent batch sizes. Solutions depicted under nominal realization conditions. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 1. Impact of Adjustable Variables Selection on the Size and Computational Tractability of the Resulting Models

	Max Profit			Min Makespan		
	SRO	ARO	ARO	SRO	ARO	ARO
Adjustable variables	–	T	T, B, S	–	T	T, B, S
# Event points	7	7	7	8	8	8
CPU time (s)	72	269	32,925	91	324	75,129
# Nodes	16,289	14,924	17,367	44,502	40,453	62,642
# Variables (cont.)	3,812	7,385	43,322	5,188	12,290	64,171
# Variables (bin.)	78	78	78	94	94	94
# Constraints	8,421	12,276	76,373	12,876	18,607	112,177
Worst-case objective	934.1	1,034.7	1,034.7	12.46	12.15	12.08

Table 2. Impact of Number of Event Points on the ARO Model's Size, Computational Tractability, and Optimal Solution

	Max Profit				Min Makespan			
# Event points	5	6	7	8	5	6	7	8
CPU time (s)	1	28	269	2,839	–	5	95	324
# Nodes	135	3,432	14,924	66,845	–	208	3,442	40,453
# Variables (cont.)	2,989	4,951	7,385	10,291	4,848	6,166	8,968	12,290
# Variables (bin.)	46	62	78	94	46	62	78	94
# Constraints	4,948	8,208	12,276	17,152	5,717	9,127	12,120	18,607
Worst-case objective	949.8	968.4	1,034.7	1,034.7	Infeas.	12.47	12.33	12.15

to assume values that are larger than their nominal values. However, although such gaps are present in all depicted schedules, it is important to note that the ARO solutions feature smaller and less frequent gaps than the SRO solutions. To that end, the ARO framework leads to schedules with smaller amount of potentially idle time and, hence, higher equipment utilization. Consequently, the ARO solutions correspond to schedules with better guaranteed, worst-case performance. In particular, for the objective of profit maximization, the minimum guaranteed profits were \$934.1 for the case of SRO and \$1,034.7 for both ARO cases; that is, in this instance the adjustable robust solutions were able to guarantee a profit at least as 10.8% higher than the static robust solution. For the objective of makespan minimization, the maximum guaranteed makespan was reduced from 12.47 h, for the case of SRO, to 12.15 h, for the case of ARO with only adjustment of timing variables, and further down to 12.08 h for the full adjustment case.

It is evident that expanding the set of which variables are to be considered as adjustable results in less conservative solutions, exhibiting a positive impact on the worst-case, robust optimal objective values. However, this comes at a computational tractability cost, since the sizes of the resulting MILP formulations also increase. Table 1 presents some relevant information gathered from the computational experiment described above. We can observe that, although the ARO formulations reference the same amount of binary variables as their SRO counterparts, they reference a larger number of continuous variables and also include a larger number of constraints. As a consequence, the CPU time required to solve the ARO cases where adjustment only occurs on the timing variables required almost four times the amount of CPU time required to solve the SRO cases, while the CPU time required to solve the full adjustment cases was approximately three orders of magnitude larger. To that end, and given also the fact that the objective improvement as one goes from partial to

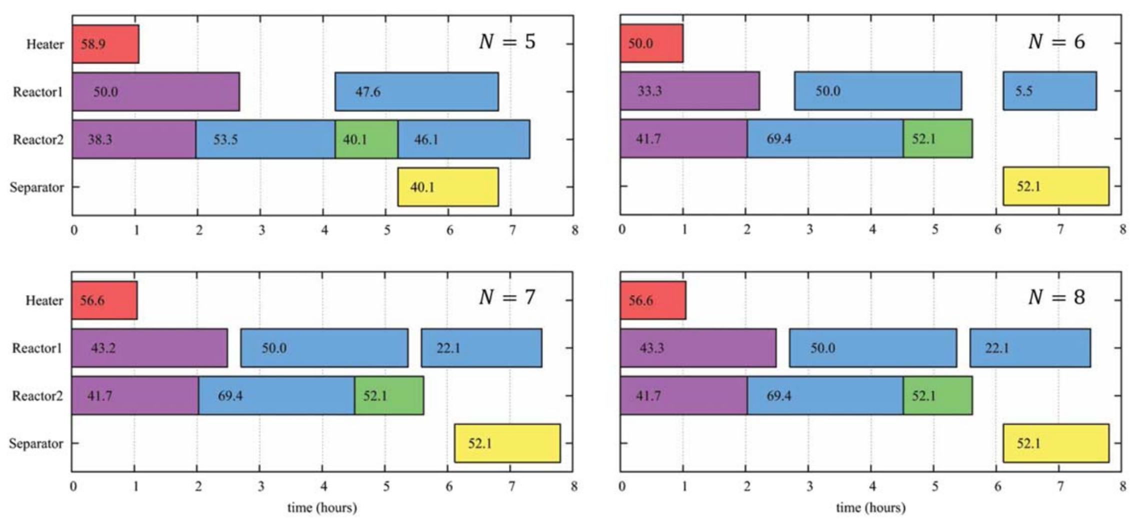


Figure 4. Impact of number of event points, *N*, on the optimal ARO schedules for the case of profit maximization.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 3. Worst-Case Profit (in \$) for the Case of ARO and for Various Levels of ξ and ϕ

		ϕ				
		0.00	0.25	0.50	0.75	1.00
ξ	0.00	1,498.6	1,498.6	1,498.6	1,498.6	1,498.6
	0.10	1,400.8	1,346.5	1,310.3	1,299.3	1,288.5
	0.20	1,285.4	1,212.0	1,136.6	1,100.2	1,078.4
	0.30	1,201.8	1,150.9	1,034.7	924.3	877.5
	0.40	1,159.9	1,069.2	931.2	788.5	715.5
	0.50	1,114.9	983.8	818.7	631.8	592.1

Table 4. Worst-Case Makespan (in Hours) for the Case of ARO and for Various Levels of ξ and ϕ

		ϕ				
		0.00	0.25	0.50	0.75	1.00
ξ	0.00	10.67	10.67	10.67	10.67	10.67
	0.10	10.69	10.87	11.06	11.14	11.27
	0.20	10.93	11.32	11.59	11.72	11.87
	0.30	11.18	11.69	12.15	12.31	12.47
	0.40	11.48	12.19	12.71	12.97	13.07
	0.50	11.87	12.82	13.28	13.48	13.66

Table 5. Worst-Case Profit Recovery Ratio (%) for Various Levels of ξ and ϕ

		ϕ				
		0.00	0.25	0.50	0.75	1.00
ξ	0.00	–	–	–	–	–
	0.10	41	19	0	0	–
	0.20	36	26	7	0	–
	0.30	40	47	35	13	–
	0.40	48	53	47	23	–
	0.50	51	55	48	13	–

Table 6. Worst-Case Makespan Recovery Ratio (%) for Various Levels of ξ and ϕ

		ϕ				
		0.00	0.25	0.50	0.75	1.00
ξ	0.00	–	–	–	–	–
	0.10	97	86	67	63	–
	0.20	78	59	41	36	–
	0.30	72	55	32	26	–
	0.40	66	47	27	13	–
	0.50	60	36	23	19	–

full adjustment was only marginal, we conclude that adjusting only on the timing variables represents a favorable trade off between optimality and computational tractability, and we will adopt this practice in the remainder of our study.

Impact of number of event points

The number of event points, N, defined in a continuous-time scheduling model constitutes an important choice for the modeler, and the implications of this choice have been extensively discussed in the deterministic scheduling literature.[73–75] Choosing too few event points can lead to suboptimal solutions, since the resulting models possess inadequate flexibility to accommodate the diversity of time points at which tasks begin and end in the actual optimal solutions. Conversely, choosing too many event points results in unnecessarily large models that offer no benefit in terms of solution quality. The established method is to determine the number of event points for a specific instance in an iterative procedure where the number of these points is increased until no improvement in the objective value is observed. In most (although not all) cases, this procedure results in the minimum number of event points that admits the instance's true optimal solution.

Similarly to the deterministic case, the number of event points affects the scheduling flexibility admitted by the corresponding SRO and ARO models as well. For the case of ARO, in particular, the number of event points also determines the number of stages considered in this multi-stage decision-making setup, whereby a selection of N event points results in an $(N-1)$-stage adjustable robust model. To investigate the effect that the number of event points has on the optimality of the ARO solutions, we focus on the same motivating example as the one used in the previous section, and we solve the case of ARO (only timing variables adjustable) for values of N ranging between 5 and 8. The results are presented in Table 2, while the corresponding optimal schedules for the case of profit maximization are provided in Figure 4. We observe that, as N increases from 5 to 7, the worst-case optimal profit increases from \$949.8 to \$1,034.7, while no further profit increase is observed when using eight event points instead; that is, the selection of seven event points is the "best" for this instance. Similarly, the worst-case optimal makespan decreases from 12.47 to 12.15 h as N increases from 6 to 8, while the case of five event points is in fact infeasible, being unable to provide the schedule with enough flexibility to ensure robust feasibility. In the remainder of this article, we always present results that correspond to each benchmark instance's "best" number of event points.

The price of robustness

In this section, we investigate the effect that the size and shape of the uncertainty set has on the quality of the optimal solution; that is, we quantify the deterioration of the robust optimal objective value as the uncertainty set is enlarged or, as it is commonly referred to in the robust optimization literature, we quantify the "price of robustness" in this application setting. More specifically, for the same benchmark instance with before as an illustrative example, we report in Tables 3 and 4 the objective values of the robust optimal ARO solutions for a range of values of the uncertainty set parameters ξ and ϕ.

As we move from the top-left to the bottom-right of these tables, we observe that the optimal objective values monotonically deteriorate, since a more conservative solution is required to ensure robust feasibility in the context of a more enlarged uncertainty set. Note that each entry in the first row ($\xi=0.00$) corresponds to the deterministic case, as the uncertainty sets collapse to a singleton point (the nominal realization), irrespectively of the value of ϕ. Furthermore, note that the last column ($\phi=1.00$) corresponds to cases where the uncertainty set is the full hyperbox and, thus, the applicable solutions correspond to the pessimistic scenario where each uncertain parameter, α_{in}, has independently realized to its worst possible value, $(1+\xi)\alpha_i^0$. The values in the right-most column essentially reflect what is commonly referred to, in the robust optimization lingo, as "Soyster solutions".[19] We highlight that similar observations also hold for the case of SRO. However, the SRO solutions would have exhibited a higher

Table 7. Comparison of the SRO and ARO Frameworks Across the Full Set of Literature Benchmark Instances

	Max Profit				Min Makespan			
Instance	SRO	ARO	WCD	RR	SRO	ARO	WCD	RR
P1	934.1	1,034.7	1,219.2	35.3	12.47	12.15	11.46	31.6
P2	1,349.5	1,402.6	1,619.4	19.7	8.36	8.21	7.68	22.0
P3	1,150.0	1,150.0	1,290.2	0.0	16.03	15.87	13.36	6.0
P4	5,280.0	5,280.0	5,619.5	0.0	6.60	6.50	6.24	28.1
P5	–	6.0	8.0	–	–	13.55	12.47	–
P6	191,888	192,568	247,360	1.2	16.58	16.55	14.92	1.8
P7	1316.0	1,404.0	1,568.0	34.9	29.39	29.09	25.84	8.4
Avg.				15.2				16.3

Values reported in either $ or hours, depending on the applicable objective. Recovery ratio values reported in %.

Table 8. Expected Makespan Values (in Hours) Across the Full Set of Literature Benchmark Instances, and Associated Expected Recovery Ratios (in %)

Instance	SRO	ARO	EDO	ERR
P1	12.47	11.64	10.66	45.9
P2	8.36	7.99	7.06	28.5
P3	16.03	15.58	13.36	16.9
P4	6.60	6.34	6.00	43.3
P5	–	12.66	11.00	–
P6	16.58	16.28	14.38	13.6
P7	29.39	26.53	23.50	48.6
Avg.				32.8

Expectations computed across a uniform sampling of the uncertainty set.

level of objective value deterioration off the nominal case, as compared to the ARO solutions reported here.

To quantify this "reduction of conservatism" of the ARO framework, we define the *recovery ratio* to be the fraction of the optimal objective value that was recovered from the static robust case as compared with the *worst-case deterministic solution*.** More specifically, we define this ratio as

$$\mathrm{RR}=\frac{\zeta^*_{\mathrm{SRO}}-\zeta^*_{\mathrm{ARO}}}{\zeta^*_{\mathrm{SRO}}-\zeta^*_{\mathrm{WCD}}},$$

where ζ^*_{SRO} and ζ^*_{ARO} are the optimal objective values for the SRO and ARO frameworks, respectively, and ζ^*_{WCD} is the worst-case deterministic solution.

Tables 5 and 6 show that the ARO approach is able to considerably reduce the over-conservatism of the SRO approach, whereby the recovery ratio reaches approximately 35% for the moderate case of $(\xi,\phi)=(0.3,0.5)$. The tables further reveal that the recovery ratios are higher for lower values of ϕ, that is, for cases where the correlations among the uncertain parameters are stronger, which showcases the ARO framework's ability to capitalize on the postulated correlations and reduce the conservatism of the resulting solutions. We must also highlight that, in general, the worst-case deterministic solution does not correspond to an implementable schedule inasmuch it relies on future realizations that are unobserved at the time an operator needs to commit to a given decision. To that end, the reported recovery ratios are only underestimating how much of the recoverable conservatism gap has in fact been recovered by moving from the traditional SRO framework to the proposed ARO framework.

Application to a suite of literature benchmarks

In this last part of our computational studies section, we apply the ARO framework on the seven benchmark instances that are listed in the Appendix. We shall again use a moderately-sized uncertainty set with $(\xi,\phi)=(0.3,0.5)$. In Table 7 we present the objective values for the robust optimal solutions of the ARO approach. For comparison, the table also presents the corresponding SRO and worst-case deterministic solutions, as well as the corresponding recovery ratios. The average recovery ratio for all the benchmark instances we considered was approximately 16%. Of particular note is instance P5, which involves zero-wait states. Since such states require immediate transfer after their production by some task, and since the duration of a task is uncertain in our setting, the timing of this material transfer cannot be decided here-and-now, in a static robust sense. In other words, the SRO approach is inherently inadequate to address instances that feature zero-wait states. In contrast, our proposed ARO framework can address such an instance as it can suitably adjust the material transfer timing on the actual realization of the preceding task's duration. Note that, although instance P5 did not contribute to the recovery ratio averages reported in Table 7, the recovery ratio in this case can be conceptually regarded as 100%, as the ARO framework essentially restores the infeasibility exhibited by the SRO approach.

Furthermore, as previously discussed, one of the advantages of the proposed ARO framework over the traditional SRO approach is its ability to adjust and take advantage of a realization that is better than the worst-case realization. In fact, the worst-case scenario admitted by an uncertainty set is typically not very likely to occur, which in turn leads to the conclusion that a static robust solution may be overly conservative in real-life. To elucidate this, we report in Table 8 the *expected* makespan†† for all benchmark instances. These solutions are also compared against the corresponding expected deterministic optimal solutions, which constitute a valid lower bound for a non-anticipative framework's performance. A recovery ratio in expectation, ERR, is also defined and reported. Across the seven instances we considered, the ARO framework manages to recover, on average, 33% of the makespan required by the SRO framework to ensure feasibility.

**The worst-case deterministic solution is a non-robust, fully-anticipative (i.e., assuming perfect information is available *a priori*) solution that constitutes a valid bound on what a non-anticipative framework, such as the SRO or ARO, can obtain. We approximated this solution by sampling the core of the uncertainty set uniformly and very finely (10,000 samples in each case), solving the corresponding deterministic optimization instance for each sample, and recording the solution with the worst objective value. The core of a decision-dependent uncertainty set is defined as the set of realizations that remain applicable for every possible decision W.

††The expected makespan values were obtained by assuming that the uncertainty realizations draw from a uniform distribution across the applicable uncertainty set, $\mathcal{A}(W^*)$, where W^* is the optimal solution in each case. Each expected value reflects 10,000 sample realizations.

Conclusions

In this article we developed for the first time in the open literature a multi-stage, ARO framework for process scheduling applications under uncertainty. We demonstrated that our ARO approach exhibits superior performance compared to the single-stage, SRO approaches that were previously proposed, resulting into solutions that are more profitable in the worst-case while insuring against the same level of risk. The percentage of SRO's risk premium that the ARO avoids having to pay varies across instances, but this can be significant, and especially so for moderate uncertainty levels. Furthermore, since our approach effectively adapts to the observed realizations of uncertainty, it has the capacity to perform even better in expectation. This is in stark contrast with all previously proposed approaches, which compute only a single, "one-fits-all" solution that is a purely "here-and-now" solution. An ancillary benefit of the ARO approach's flexibility is that it can address instances that involve zero-wait states, which are inherently unaddressable by SRO approaches. Moreover, we proposed for the first time the use of decision-dependent uncertainty sets in this context. Such sets, which generalize the non-decision-dependent sets typically used in robust optimization, provide the modeler with the means to effectively model endogenous uncertainty that is inherent in this problem. We also suggested that it is more realistic for parameter realizations to be decoupled across repeated executions of a particular task. To that end, it should be highlighted that our computational studies, which were performed on a suite of well-known literature benchmark instances, effectively considered a significantly increased (sometimes 10-fold) number of uncertain parameters compared to previous studies utilizing these benchmarks.

Acknowledgments

The authors 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. We would also like to thank Dr. Wolfram Wiesemann of Imperial College London Business School for a number of insightful discussions on the topic.

Notation

Indices

n = event points
i = tasks
j = units
s = states
u = utilities

Sets

$\mathcal{N}$ = event points in time horizon $\equiv \{1, 2, \cdots, N\}$
$\mathcal{N}_n^-$ = event points in the immediate past of event point n $\equiv \{n' \in \mathcal{N} : n - \delta n \leq n' \leq n-1\}$
$\mathcal{N}_n^+$ = event points in the immediate future of event point n $\equiv \{n' \in \mathcal{N} : n+1 \leq n' \leq n+\delta n\}$
$\mathcal{I}$ = tasks
$\mathcal{I}_j$ = tasks that can be performed in unit j
$\mathcal{I}_s^p$ = tasks that produce state s
$\mathcal{I}_s^c$ = tasks that consume state s
$\mathcal{I}_u$ = tasks that consume utility u
$\mathcal{I}^{zw}$ = tasks that produce a "zero-wait" state
$\mathcal{J}$ = units
$\mathcal{S}$ = states
$\mathcal{U}$ = utilities

Binary variables

$W_{inn'}$ = task i started at event point n and finished by event point n'

Continuous variables

z = objective value
T_n = time of event point n
$B_{inn'}$ = batch size for the task i that started at event point n and finished by event point n'
U_{un} = consumption rate of utility u at event point n
S_{sn} = amount of state s at event point n
G_{jn} = unit j is utilized at event point n (indication)

Parameters

α_i = processing time of task i (fixed)
β_i = processing time of task i (per unit batch)
γ_{iu} = utility u consumption by task i (fixed)
δ_{iu} = utility u consumption by task i (per unit batch)
ρ_{is} = proportion of state s in the total production/consumption by task i
$B_i^{\min}$ = minimum capacity for task i in unit j
$B_i^{\max}$ = maximum capacity for task i in unit j
$U_u^{\max}$ = maximum availability of utility u
S_{s0} = initial available amount of state s
$S_s^{\max}$ = maximum storage capacity for state s
D_s = demand of state s at the end of horizon
P_s = price of state s
H = horizon
$\bar{M}$ = upper bound for makespan
δn = maximum event-point span of any task

Literature Cited

1. Kallrath J. Mixed integer optimization in the chemical process industry. *Chem Eng Res Des.* 2000;78(6):809–822.
2. Wassick J. Enterprise-wide optimization in an integrated chemical complex. *Comp Chem Eng.* 2009;33(12):1950–1963.
3. Harjunkoski I, Grossmann I. A decomposition approach for the scheduling of a steel plant production. *Comp Chem Eng.* 2001; 25(11–12):1647–1660.
4. Prasad P, Maravelias C, Kelly J. Optimization of aluminum smelter casthouse operations. *Ind Eng Chem Res.* 2006;45(22):7603–7617.
5. Aguirre AM, Méndez C, Castro P. A novel optimization method to automated wet-etch station scheduling in semiconductor manufacturing systems. *Comp Chem Eng.* 2011;35(12):2960–2972.
6. Smaïli F, Angadi D, Hatch C, Herbert O, Vassiliadis V, Wilson D. Optimization of scheduling of cleaning in heat exchanger networks subject to fouling. *Food Bioprod Proc.* 1999;77(2):159–164.
7. Kopanos G, Puigjaner L, Georgiadis M. Optimal production scheduling and lot-sizing in dairy plants: the yogurt production line. *Ind Eng Chem Res.* 2010;49(2):701–718.
8. Kopanos G, Puigjaner L, Georgiadis M. Efficient mathematical frameworks for detailed production scheduling in food processing industries. *Comp Chem Eng.* 2012;42:206–216.
9. Papageorgiou L, Rotstein G, Shah N. Strategic supply chain optimization for the pharmaceutical industries. *Ind Eng Chem Res.* 2001; 40(1):275–286.
10. Kabra S, Shaik M, Rathore A. Multi-period scheduling of a multi-stage multi-product bio-pharmaceutical process. *Comp Chem Eng.* 2013;57:95–103.
11. Floudas C, Lin X. Mixed integer linear programming in process scheduling: modeling, algorithms, and applications. *Annal Operat Res.* 2005;139(1):131–162.
12. Méndez C, Cerdá J, Grossmann I, Harjunkoski I, Fahl M. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comp Chem Eng.* 2006;30(6–7):913–946.
13. Harjunkoski I, Maravelias C, Bongers P, Castro P, Engell S, Grossmann I, Hooker J, Méndez C, Sand G, Wassick J. Scope for industrial applications of production scheduling models and solution methods. *Comp Chem Eng.* 2014;62:161–193.
14. Aytug H, Lawley M, McKay K, Mohan S, Uzsoy R. Executing production schedules in the face of uncertainties: a review and some future directions. *European J Operat Res.* 2005;161(1):86–110.
15. Li Z, Ierapetritou M. Process scheduling under uncertainty: review and challenges. *Comp Chem Eng.* 2008;32(4–5):715–727.
16. Ben-Tal A, El Ghaoui L, Nemirovski A. *Robust Optimization.* Princeton, NJ: Princeton University Press, 2009.

17. Bertsimas D, Brown D, Caramanis C. Theory and applications of robust optimization. *SIAM Rev*. 2011;53(3):464–501.
18. Gorissen B, Yankolu, den Hertog D. A practical guide to robust optimization. *Omega*. 2015;53:124–137.
19. Soyster A. Technical noteconvex programming with set-inclusive constraints and applications to inexact linear programming. *Operat Res*. 1973;21(5):1154–1157.
20. Bertsimas D, Sim M. The price of robustness. *Operat Res*. 2004; 52(1):35–53.
21. Ceria S, Stubbs R. Incorporating estimation errors into portfolio selection: robust portfolio construction. *J Asset Manag*. 2006;7(2): 109–127.
22. Ben-Tal A, Nemirovski A. Robust solutions of uncertain linear programs. *Operat Res Lett*. 1999;25(1):1–13.
23. Li Z, Ding R, Floudas C. A comparative theoretical and computational study on robust counterpart optimization: I. Robust linear optimization and robust mixed integer linear optimization. *Ind Eng Chem Res*. 2011;50(18):10567–10603.
24. Janak S, Lin X, Floudas C. A new robust optimization approach for scheduling under uncertainty: II. *Uncertainty with known probability distribution*. *Comp Chem Eng*. 2007;31(3):171–195.
25. Li Z, Tang Q, Floudas C. A comparative theoretical and computational study on robust counterpart optimization: II. Probabilistic guarantees on constraint satisfaction. *Ind Eng Chem Res*. 2012; 51(19):6769–6788.
26. Lin X, Janak S, Floudas C. A new robust optimization approach for scheduling under uncertainty: I. Bounded uncertainty. *Comp Chem Eng*. 2004;28(6–7):1069–1085.
27. Li Z, Ierapetritou M. Robust optimization for process scheduling under uncertainty. *Ind Eng Chem Res*. 2008;47(12):4148–4157.
28. Li Z, Floudas C. A comparative theoretical and computational study on robust counterpart optimization: III. Improving the quality of robust solutions. *Ind Eng Chem Res*. 2014;53(33):13112–13124.
29. Li J, Misener R, Floudas C. Scheduling of crude oil operations under demand uncertainty: A robust optimization framework coupled with global optimization. *AIChE Journal*. 2012;58(8):2373–2396.
30. Ye Y, Li J, Li Z, Tang Q, Xiao X, Floudas C. Robust optimization and stochastic programming approaches for medium-term production scheduling of a large-scale steelmaking continuous casting process under demand uncertainty. *Comp Chem Eng*. 2014;66:165–185.
31. Ben-Tal A, Goryashko A, Guslitzer E, Nemirovski A. Adjustable robust solutions of uncertain linear programs. *Math Prog*. 2004; 99(2):351–376.
32. Chen X, Zhang Y. Uncertain linear programs: extended affinely adjustable robust counterparts. *Operat Res*. 2009;57(6):1469–1482.
33. Kuhn D, Wiesemann W, Georghiou A. Primal and dual linear decision rules in stochastic and robust optimization. *Math Prog*. 2011; 130(1):177–209.
34. Bertsimas D, Georghiou A. Design of near optimal decision rules in multistage adaptive mixed-integer optimization. *Operat Res*. 2015; 63(3):610–627.
35. Shi H, You F. A computational framework and solution algorithms for two-stage adaptive robust scheduling of batch manufacturing processes under uncertainty. *AIChE J*. 2015.
36. Thiele A, Terry T, M. E. Robust linear optimization with recourse. www.optimization-online.org, 2010.
37. Zeng B, Zhao L. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operat Res Lett*. 2013;41(5):457–461. ISSN 0167-6377. doi:10.1016/j.orl.2013.05.003.
38. Ierapetritou M, Pistikopoulos E. Batch plant design and operations under uncertainty. *Ind Eng Chem Res*. 1996;35(3):772–787.
39. Grossmann I, Balasubramanian J. Approximation to multistage stochastic optimization in multiperiod batch plant scheduling under demand uncertainty. *Ind Eng Chem Res*. 2004;43(14):3695–3713.
40. Bonfill A, Bagajewicz M, Espuña A, Puigjaner L. Risk management in the scheduling of batch plants under uncertain market demand. *Ind Eng Chem Res*. 2004;43(3):741–750.
41. Bonfill A, Espuña A, Puigjaner L. Addressing robustness in scheduling batch processes with uncertain operation times. *Ind Eng Chem Res*. 2005;44(5):1524–1534.
42. Maravelias C. On the combinatorial structure of discrete-time MIP formulations for chemical production scheduling. *Comp Chem Eng*. 2012;38:204–212.
43. Velez S, Maravelias C. Multiple and nonuniform time grids in discrete-time MIP models for chemical production scheduling. *Comp Chem Eng*. 2013;53:70–85.
44. Castro P, Barbosa-po A, Matos H. An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. *Ind Eng Chem Res*. 2001;40(9):2059–2068.
45. Maravelias C, Grossmann I. New general continuous-time state—task network formulation for short-term scheduling of multipurpose batch plants. *Ind Eng Chem Res*. 2003;42(13):3056–3074.
46. Gupta S, Karimi I. An improved MILP formulation for scheduling multiproduct, multistage batch plants. *Ind Eng Chem Res*. 2003; 42(11):2365–2380.
47. Castro P, Barbosa-po A, Matos H, Novais A. Simple continuous-time formulation for short-term scheduling of batch and continuous processes. *Ind Eng Chem Res*. 2004;43(1):105–118.
48. Sundaramoorthy A, Karimi I. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. *Chem Eng Sci*. 2005;60(10):2679–2702.
49. Liu Y, Karimi I. Scheduling multistage batch plants with parallel units and no interstage storage. *Comp Chem Eng*. 2008;32(4–5):671–693.
50. Giménez D, Henning G, Maravelias C. A novel network-based continuous-time representation for process scheduling: Part I. Main concepts and mathematical formulation. *Comp Chem Eng*. 2009;33(9): 1511–1528.
51. Giménez D, Henning G, Maravelias C. A novel network-based continuous-time representation for process scheduling: Part II. General framework. *Comp Chem Eng*. 2009;33(10):1644–1660.
52. Ierapetritou M, Floudas C. Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. *Ind Eng Chem Res*. 1998;37(11):4341–4359.
53. Giannelos N, Georgiadis M. A simple new continuous-time formulation for short-term scheduling of multipurpose batch processes. *Ind Eng Chem Res*. 2002;41(9):2178–2184.
54. Janak S, Lin X, Floudas C. Enhanced continuous-time unit-specific event-based formulation for short-term scheduling of multipurpose batch processes: resource constraints and mixed storage policies. *Ind Eng Chem Res*. 2004;43(10):2516–2533.
55. Shaik M, Floudas C. Improved unit-specific event-based continuous-time model for short-term scheduling of continuous processes: rigorous treatment of storage requirements. *Ind Eng Chem Res*. 2007; 46(6):1764–1779.
56. Shaik M, Floudas C. Unit-specific event-based continuous-time approach for short-term scheduling of batch plants using RTN framework. *Comp Chem Eng*. 2008;32(1–2):260–274.
57. Shaik M, Floudas C. Novel Unified Modeling Approach for Short-Term Scheduling. *Ind Eng Chem Res*. 2009;48(6):2947–2964.
58. Susarla N, Li J, Karimi I. A novel approach to scheduling multipurpose batch plants using unit-slots. *AIChE J*. 2009;56(7):1859–1879.
59. Seid R, Majozi T. A robust mathematical formulation for multipurpose batch plants. *Chem Eng Sci*. 2012;68(1):36–53.
60. Vooradi R, Shaik M. Improved three-index unit-specific event-based model for short-term scheduling of batch plants. *Comp Chem Eng*. 2012;43(19):148–172.
61. Vooradi R, Shaik M. Rigorous unit-specific event-based model for short-term scheduling of batch plants using conditional sequencing and unit-wait times. *Ind Eng Chem Res*. 2013;52(36):12950–12972.
62. Mishra B, Mayer E, Raisch J, Kienle A. Short-term scheduling of batch processes. A comparative study of different approaches. *Ind Eng Chem Res*. 2005;44(11):4022.
63. Shaik M, Janak S, Floudas C. Continuous-time models for short-term scheduling of multipurpose batch plants: a comparative study. *Ind Eng Chem Res*. 2006;45(18):6190–6209.
64. Gounaris C, Wiesemann W, Floudas C. The robust capacitated vehicle routing problem under demand uncertainty. *Operat Res*. 2013; 61(3):677–693.
65. Bärmann A, Heidt A, Martin A. Polyhedral approximation of ellipsoidal uncertainty sets via extended formulations. www.optimization-online.org, 2014.
66. Goel V, Grossmann I. A class of stochastic programs with decision dependent uncertainty. *Math Prog*. 2006;108(2–3):355–394.
67. Solak S, Clarke J, Johnson E, Barnes E. Optimization of R&D project portfolios under endogenous uncertainty. *European J Operat Res*. 2010;207(1):420–433.
68. Gupta V, Grossmann I. Solution strategies for multistage stochastic programming with endogenous uncertainties. *Comp Chem Eng*. 2011;35(11):2235–2247.
69. Gupta V, Grossmann I. A new decomposition algorithm for multistage stochastic programs with endogenous uncertainties. *Comp Chem Eng*. 2014;62:62–79.

70. Chen X, Sim M, Sun P, Zhang J. A linear decision-based approximation approach to stochastic programming. *Operat Res.* 2008; 56(2):344–357.
71. Gurobi Optimization, Inc. Gurobi Optimizer Reference Manual. 2015.
72. Kondili E, Pantelides C, Sargent R. A general algorithm for short-term scheduling of batch operationsI. MILP formulation. *Comp Chem Eng.* 1993;17(2):211–227.
73. Li J, Floudas C. Optimal event point determination for short-term scheduling of multipurpose batch plants via unit-specific event-based continuous-time approaches. *Ind Eng Chem Res.* 2010;49(16):7446–7469.
74. Li J, Susarla N, Karimi I, Shaik M, Floudas C. An analysis of some unit-specific event-based models for the short-term scheduling of noncontinuous processes. *Ind Eng Chem Res.* 2010;49(2):633–647.
75. Seid R, Majozi T. A novel technique for prediction of time points for scheduling of multipurpose batch plants. *Chem Eng Sci.* 2012; 68(1):54–71.
76. Karimi I, McDonald C. Planning and scheduling of parallel semicontinuous processes. 2. Short-term scheduling. *Ind Eng Chem Res.* 1997;36(7):2701–2714.
77. Pantelides C. Unified frameworks for the optimal process planning and scheduling. In *Proceedings of the 2nd Conference on the Foundations of Computer Aided Operations*, p. 253. Cache Publications, 1994.

Appendix A: Detailed Constraint Reformulation Process

In this appendix we present in detail the process that must be followed so as to reformulate the constraints of the deterministic model into their adjustable robust counterparts. The complete ARO model results when all inequality constraints in the set $\mathcal{M}_I$ have been reformulated as per the process presented in Section "Reformulation of a Representative Inequality from Set $\mathcal{M}_I$" and all equality constraints in the set $\mathcal{M}_E$ have been reformulated as per the process presented in Section "Reformulation of a Representative Equality from Set $\mathcal{M}_E$". No reformulation is required for the constraints in the set $\mathcal{M}_C$, which must simply be appended to the robust counterpart in the same form as they appear in the deterministic model. The reader is directed to the main discussion for the definitions of sets $\mathcal{M}_I$, $\mathcal{M}_E$, and $\mathcal{M}_C$.

For ease of exposition, we will demonstrate the reformulation steps for two representative sets of constraints. More specifically, in Section "Reformulation of a Representative Inequality from Set $\mathcal{M}_I$" we will focus on the reformulation of a timing inequality (Eq. 3), while in Section "Reformulation of a Representative Equality from Set $\mathcal{M}_E$" we will focus on the reformulation of a material balance equality (Eq. 6). The remaining constraints in the sets $\mathcal{M}_I$ and $\mathcal{M}_E$ can be reformulated similarly. In both cases, we will consider the most general case where every continuous variable that participates in these constraints is to be adjusted. If a continuous variable of interest x is not to be adjusted, the robust counterpart can be derived by simply ignoring the slope adjustment variables, that is, by replacing the slope variables $[x]_{i'n''}$, for all $i' \in \mathcal{I}$ and $n'' \in \mathcal{N}$, in the reformulated constraints with the value of 0. We remark that the reformulation procedures presented below result in an ARO model that retains the MILP structure originally present in the deterministic model.

Reformulation of a representative inequality from set $\mathcal{M}_I$

Let a representative timing constraint; that is, a constraint shown in Eq. 3 for some specific $j \in \mathcal{J}$, $n \in \mathcal{N}$, and some $n' \in \mathcal{N}_n^+$. After application of parameter time-splitting, $\alpha_i \leftarrow \alpha_{in}$, this constraint must hold for every possible realization $\alpha \in \mathcal{A}(W)$. Equation A1 applies.

$$T_{n'} - T_n \geq \sum_{i \in \mathcal{I}_j} (\alpha_{in} W_{inn'} + \beta_i B_{inn'}) \quad \forall\ \alpha \in \mathcal{A}(W) \tag{A1}$$

After introducing the adjustment variables and replacing the continuous variables with their adjusted expressions (affine decision rules), the above constraint is reformulated into Eq. A2. Note how we have gathered into the right-hand-side all terms that do not reference the realization of an uncertain parameter α.

$$\sum_{i \in \mathcal{I}_j} \alpha_{in} W_{inn'} + \sum_{i' \in \mathcal{I}} \sum_{n'' \in \mathcal{N}} \left(-[T_{n'}]_{i'n''} + [T_n]_{i'n''} + \sum_{i \in \mathcal{I}_j} \beta_i [B_{inn'}]_{i'n''} \right) \alpha_{i'n''} \leq [T_{n'}]_0 - [T_n]_0 - \sum_{i \in \mathcal{I}_j} \beta_i [B_{inn'}]_0 \ \forall\ \alpha \in \mathcal{A}(W) \tag{A2}$$

The semi-infinite robustness restriction ($\forall\ \alpha \in \mathcal{A}(W)$) can now be equivalently replaced with an inner optimization problem, leading to the constraint

$$\theta^* \leq [T_{n'}]_0 - [T_n]_0 - \sum_{i \in \mathcal{I}_j} \beta_i [B_{inn'}]_0, \tag{A3}$$

where θ^* is the optimal objective value of the following maximization problem:

$$\begin{aligned}
\theta^* = \max_{\alpha \in \mathbb{R}_+^{|\mathcal{I}||\mathcal{N}|}} \ & \sum_{i \in \mathcal{I}_j} \alpha_{in} W_{inn'} + \sum_{i' \in \mathcal{I}} \sum_{n'' \in \mathcal{N}} \left(-[T_{n'}]_{i'n''} + [T_n]_{i'n''} + \sum_{i \in \mathcal{I}_j} \beta_i [B_{inn'}]_{i'n''} \right) \alpha_{i'n''} \\
\text{s.t.} \ & \sum_{i' \in \mathcal{I}} \sum_{n'' \in \mathcal{N}} h_{fi'n''} \left(\sum_{n''' \in \mathcal{N}_{n''}^-} W_{i'n'''n''} \right) \alpha_{i'n''} \leq \sum_{i' \in \mathcal{I}} \sum_{n'' \in \mathcal{N}} g_{fi'n''} \left(\sum_{n''' \in \mathcal{N}_{n''}^-} W_{i'n'''n''} \right) \quad \forall\ f \in \mathcal{F} \\
& -\alpha_{i'n''} \leq -\alpha_{i'n''}^{lb} \quad \forall\ i' \in \mathcal{I},\ \forall\ n'' \in \mathcal{N} \\
& +\alpha_{i'n''} \leq +\alpha_{i'n''}^{ub} \quad \forall\ i' \in \mathcal{I},\ \forall\ n'' \in \mathcal{N}.
\end{aligned}$$

However, due to LP duality, θ^* can be also computed via a minimization problem. To that end, we introduce three new sets of continuous variables, namely $p^{jnn'} \in \mathbb{R}_+^{|\mathcal{F}|}$, $q^{jnn'} \in \mathbb{R}_+^{|\mathcal{I}||\mathcal{N}|}$ and $r^{jnn'} \in \mathbb{R}_+^{|\mathcal{I}||\mathcal{N}|}$, which serve as the dual variables that are associated with each of the maximization problem's constraints. More specifically, variables $p^{jnn'}$, $q^{jnn'}$, and $r^{jnn'}$ correspond to the constraints representing the uncertainty set's generic facets, the lower bound facets and the upper bound facets, respectively. Note how these variables are specific to the individual constraint of interest, that is, they are associated with the specific timing constraint of given $j \in \mathcal{J}$, $n \in \mathcal{N}$, and $n' \in \mathcal{N}_n^+$. This implies that a separate set of dual variables is required for each original deterministic constraint in the set $\mathcal{M}_I$. Given these new variables, θ^* can be computed as the optimal objective value of the following minimization problem:

$$\theta^* = \min_{\substack{p^{jnn'} \in \mathbb{R}_+^{|\mathcal{F}|}, \\ q^{jnn'} \in \mathbb{R}_+^{|\mathcal{I}||\mathcal{N}|}, \\ r^{jnn'} \in \mathbb{R}_+^{|\mathcal{I}||\mathcal{N}|}}} \sum_{i' \in \mathcal{I}} \sum_{n'' \in \mathcal{N}} \left(\sum_{f \in \mathcal{F}} g_{fi'n''} \Big(\sum_{n''' \in \mathcal{N}_{n''}^-} W_{i'n'''n''} \Big) p_f^{jnn'} - \alpha_{i'n''}^{lb} q_{i'n''}^{jnn'} + \alpha_{i'n''}^{ub} r_{i'n''}^{jnn'} \right)$$

$$\text{s.t.} \left. \begin{array}{c} \displaystyle\sum_{f \in \mathcal{F}} h_{fi'n''} \left(\sum_{n''' \in \mathcal{N}_{n''}^-} W_{i'n'''n''} \right) p_f^{jnn'} - q_{i'n''}^{jnn'} + r_{i'n''}^{jnn'} \geq \\ \mathbf{1}_{\{i' \in \mathcal{I}_j\}} \mathbf{1}_{\{n''=n\}} W_{i'n''n'} - [T_{n'}]_{i'n''} + [T_n]_{i'n''} + \displaystyle\sum_{i \in \mathcal{I}_j} \beta_i [B_{inn'}]_{i'n''} \end{array} \right\} \forall\ i' \in \mathcal{I},\ \forall\ n'' \in \mathcal{N},$$

where 1_X is the indicator function (i.e., equals 1 if condition X holds, and equals 0 otherwise).

We now observe that, for constraint (Eq. A3) to hold, it suffices for the inner minimization problem to have some feasible solution that evaluates to an objective value that is no greater than the constraint's right-hand-side. Consequently, we can drop the minimization operator and simplify into the set of Eq. A4.

$$\begin{aligned} &\sum_{i' \in \mathcal{I}} \sum_{n'' \in \mathcal{N}} \left(\sum_{f \in \mathcal{F}} g_{fi'n''} \Big(\sum_{n''' \in \mathcal{N}_{n''}^-} W_{i'n'''n''} \Big) p_f^{jnn'} - \alpha_{i'n''}^{lb} q_{i'n''}^{jnn'} + \alpha_{i'n''}^{ub} r_{i'n''}^{jnn'} \right) \leq [T_{n'}]_0 - [T_n]_0 - \sum_{i \in \mathcal{I}_j} \beta_i [B_{inn'}]_0 \\ &\left. \begin{array}{c} \displaystyle\sum_{f \in \mathcal{F}} h_{fi'n''} \Big(\sum_{n''' \in \mathcal{N}_{n''}^-} W_{i'n'''n''} \Big) p_f^{jnn'} - q_{i'n''}^{jnn'} + r_{i'n''}^{jnn'} \geq \\ \mathbf{1}_{\{i' \in \mathcal{I}_j\}} \mathbf{1}_{\{n''=n\}} W_{i'n''n'} - [T_{n'}]_{i'n''} + [T_n]_{i'n''} + \displaystyle\sum_{i \in \mathcal{I}_j} \beta_i [B_{inn'}]_{i'n''} \end{array} \right\} \quad \forall\ i' \in \mathcal{I},\ \forall\ n'' \in \mathcal{N} \end{aligned} \tag{A4}$$

Because of the uncertainty set's decision-dependency, products of binary variables w and dual variables p appear in the above set of constraints. We thus implement an exact linearization of these bilinear terms via the standard McCormick technique. More specifically, we introduce auxiliary variables $s_{fi'n''}^{jnn'}$ to capture the terms $(\sum_{n''' \in \mathcal{N}_{n''}^-} W_{i'n'''n''}) p_f^{jnn'}$, where it is important to highlight that the sum is itself a binary expression. Note that an auxiliary variable $s_{fi'n''}^{jnn'}$ needs to be introduced only when $h_{fi'n''} \neq 0$ or $g_{fi'n''} \neq 0$. Typically this only holds in a small number of cases, since in practice a decision-dependent set $\mathcal{A}(W)$, as in Eq. 20, would reference only a small number of uncertain parameters, $\alpha_{i'n''}$, in each of its generic facets, $f \in \mathcal{F}$. After linearization of the bilinear terms, we result in the set of Eq. A5, which constitute the adjustable robust counterpart of the original deterministic constraint from Eq. 3. The implication constraints stemming from the McCormick linearization can be implemented in a modern MILP solver either as SOS1 indications or as "big-M" constraints. The former is recommended in this case, as the fact that the p variables are duals of a primal LP of interest makes it cumbersome to verify the validity of any chosen big-M value.

$$\begin{aligned} &\sum_{i' \in \mathcal{I}} \sum_{n'' \in \mathcal{N}} \left(\sum_{f \in \mathcal{F}} g_{fi'n''} s_{fi'n''}^{jnn'} - \alpha_{i'n''}^{lb} q_{i'n''}^{jnn'} + \alpha_{i'n''}^{ub} r_{i'n''}^{jnn'} \right) \leq [T_{n'}]_0 - [T_n]_0 - \sum_{i \in \mathcal{I}_j} \beta_i [B_{inn'}]_0 \\ &\left. \begin{array}{c} \displaystyle\sum_{f \in \mathcal{F}} h_{fi'n''} s_{fi'n''}^{jnn'} - q_{i'n''}^{jnn'} + r_{i'n''}^{jnn'} \geq \\ \mathbf{1}_{\{i' \in \mathcal{I}_j\}} \mathbf{1}_{\{n''=n\}} W_{i'n''n'} - [T_{n'}]_{i'n''} + [T_n]_{i'n''} + \displaystyle\sum_{i \in \mathcal{I}_j} \beta_i [B_{inn'}]_{i'n''} \\ \left. \begin{array}{c} \displaystyle\sum_{n''' \in \mathcal{N}_{n''}^-} W_{i'n'''n''} = 1 \Rightarrow s_{fi'n''}^{jnn'} \geq p_f^{jnn'} \\ \displaystyle\sum_{n''' \in \mathcal{N}_{n''}^-} W_{i'n'''n''} = 0 \Rightarrow s_{fi'n''}^{jnn'} \leq 0 \\ s_{fi'n''}^{jnn'} \leq p_f^{jnn'} \end{array} \right\} \quad \forall\ f \in \mathcal{F} \end{array} \right\} \quad \forall\ i' \in \mathcal{I},\ \forall\ n'' \in \mathcal{N} \end{aligned} \tag{A5}$$

Reformulation of a representative equality from set $\mathcal{M}_E$

Let a representative material balance equality; that is, a constraint shown in Eq. 6 for some specific $s\in\mathcal{S}$ and some $n\in\mathcal{N}$. Since this constraint must hold for every possible realization $\alpha\in\mathcal{A}(W)$, Eq. A6 applies.

$$S_{sn}=S_{s(n-1)}+\sum_{i\in\mathcal{I}_s^p}\rho_{is}\sum_{n'\in\mathcal{N}_n^-}B_{in'n}-\sum_{i\in\mathcal{I}_s^c}\rho_{is}\sum_{n'\in\mathcal{N}_n^+}B_{inn'}\quad\forall\ \alpha\in\mathcal{A}(W) \tag{A6}$$

After application of parameter time-splitting, $\alpha_i\leftarrow\alpha_{in}$, as well as after introducing the adjustment variables and replacing the continuous variables with their adjusted expressions (affine decision rules), the above constraint is reformulated into Eq. A7. Note how we have gathered into the right-hand-side all terms that do not reference the realization of an uncertain parameter α.

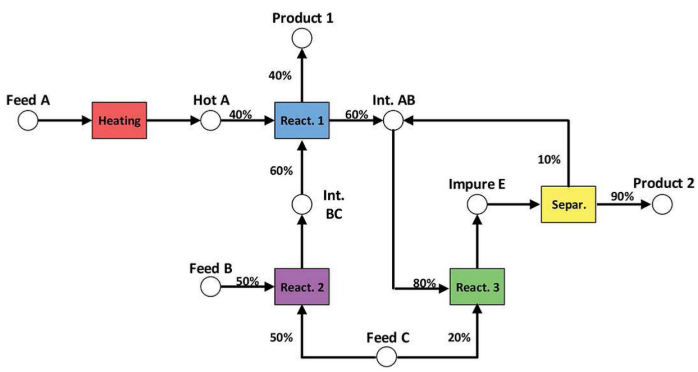
$$\begin{aligned}&\sum_{i'\in\mathcal{I}}\sum_{n''\in\mathcal{N}}\left([S_{sn}]_{i'n''}-\left[S_{s(n-1)}\right]_{i'n''}-\sum_{i\in\mathcal{I}_s^p}\rho_{is}\sum_{n'\in\mathcal{N}_n^-}[B_{in'n}]_{i'n''}+\sum_{i\in\mathcal{I}_s^c}\rho_{is}\sum_{n'\in\mathcal{N}_n^+}[B_{inn'}]_{i'n''}\right)\alpha_{i'n''}=\\&-[S_{sn}]_0+\left[S_{s(n-1)}\right]_0+\sum_{i\in\mathcal{I}_s^p}\rho_{is}\sum_{n'\in\mathcal{N}_n^-}[B_{in'n}]_0-\sum_{i\in\mathcal{I}_s^c}\rho_{is}\sum_{n'\in\mathcal{N}_n^+}[B_{inn'}]_0\quad\forall\ \alpha\in\mathcal{A}(W)\end{aligned} \tag{A7}$$

In order for such an equality, which constitutes an affine expression of uncertain parameters α, to hold for any possible realization admitted by the uncertainty set ($\forall\ \alpha\in\mathcal{A}(W)$), and assuming that the set $\mathcal{A}(W)$ has interior (i.e., includes infinitely many points),ô[‡‡]ô the equality needs to hold coefficient-wise; that is, the aggregate coefficients of each uncertain parameter, $\alpha_{i'n''}$, as well as the aggregate term not multiplied by any uncertain parameter, must be all set to zero. This "coefficient matching" exercise results into Eq. A8, which constitute the adjustable robust counterpart of the original deterministic constraint from Eq. 6.

$$\begin{aligned}&[S_{sn}]_0=\left[S_{s(n-1)}\right]_0+\textstyle\sum_{i\in\mathcal{I}_s^p}\rho_{is}\sum_{n'\in\mathcal{N}_n^-}[B_{in'n}]_0\\&-\textstyle\sum_{i\in\mathcal{I}_s^c}\rho_{is}\sum_{n'\in\mathcal{N}_n^+}[B_{inn'}]_0\\&[S_{sn}]_{i'n''}=\left[S_{s(n-1)}\right]_{i'n''}+\textstyle\sum_{i\in\mathcal{I}_s^p}\rho_{is}\sum_{n'\in\mathcal{N}_n^-}[B_{in'n}]_{i'n''}\\&-\textstyle\sum_{i\in\mathcal{I}_s^c}\rho_{is}\sum_{n'\in\mathcal{N}_n^+}[B_{inn'}]_{i'n''}\quad\forall\ i'\in\mathcal{I},\ \forall\ n''\in\mathcal{N}\end{aligned} \tag{A8}$$

Appendix B: Data for Benchmark Problem Instances

Instance P1 (from Kondili et al.[72])



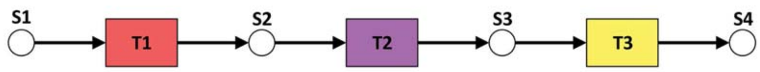
States:	Feed A	Feed B	Feed C	Hot A	Int. BC	Int. AB	Impure E	Product 1	Product 2
Capacity (kg)	1000	1000	1000	100	200	150	200	1000	1000
Initial load (kg)	1000	1000	1000						
Price ($/kg)								10	10
Demand (kg)								100	100
Horizon = 8 hours									

Units:	Heater		Reactor 1		Reactor 2		Separator	
Minimum Load (kg)	20		10		16		40	
Maximum Load (kg)	100		50		80		200	
Tasks	α	β	α	β	α	β	α	β
Heating	0.667	0.007						
Reaction 1			1.334	0.027	1.334	0.017		
Reaction 2			1.334	0.027	1.334	0.017		
Reaction 3			0.667	0.013	0.667	0.008		
Separation							1.334	0.007

α in hours, β in hours/kg

[‡‡]This assumption usually holds. If this is not the case, and the uncertainty set consists of one or more discrete points (scenarios), then one may simply append in the formulation the equality constraint with the parameters assuming each time their corresponding values in each scenario.

Instance P2 (from Karimi and McDonald[76])



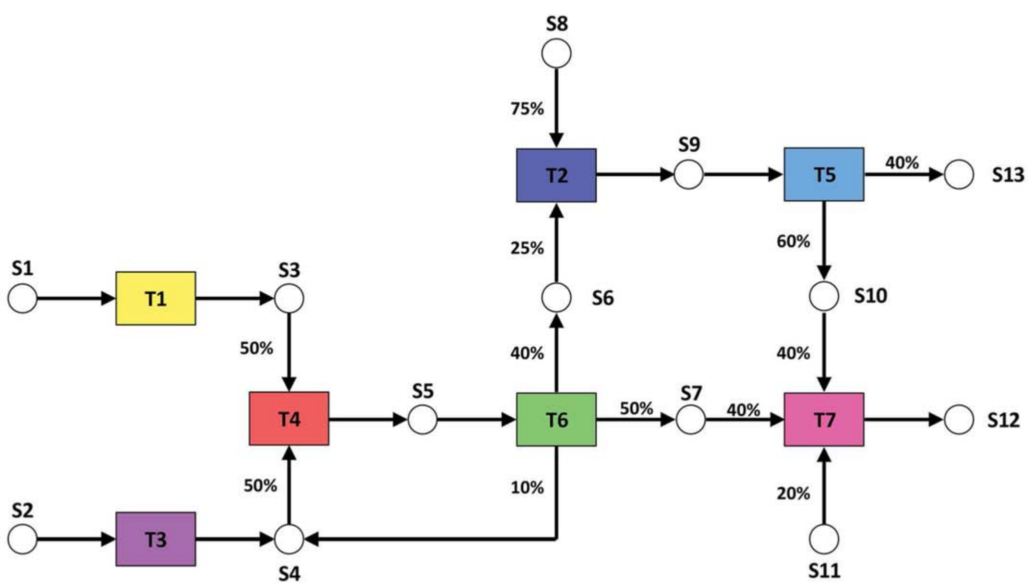
States:	S1	S2	S3	S4
Capacity (kg)	UL	UL	UL	UL
Initial load (kg)	AA			
Price ($/kg)				5
Demand (kg)				300

Horizon = 8 hours

Units:	U1		U2		U3		U4		U5	
Minimum Load (kg)										
Maximum Load (kg)	100		150		200		150		150	
Tasks	α	β	α	β	α	β	α	β	α	β
T1	1.333	0.01333	1.333	0.01333						
T2					1.000	0.00500				
T3							0.667	0.00445	0.667	0.00445

UL: unlimited, AA: available as and when required
α in hours, β in hours/kg

Instance P3 (from Karimi and McDonald[76])



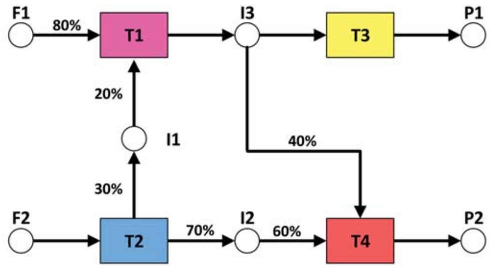
States:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
Capacity (kg)	UL	UL	UL	UL	UL	UL	UL	UL	UL	UL	UL	UL	UL
Initial load (kg)	AA	AA				50	50	AA			AA		
Price ($/kg)												5	5
Demand (kg)												100	200

Horizon = 8 hours

Units:	heater		reactor1		reactor2		separator		mixer1		mixer2	
Minimum Load (kg)									20		20	
Maximum Load (kg)	100		100		150		300		200		200	
Tasks	α	β	α	β	α	β	α	β	α	β	α	β
T1	0.667	0.00667										
T2	1.000	0.01000										
T3			1.333	0.01333	1.333	0.00889						
T4			0.667	0.00667	0.667	0.00445						
T5			1.333	0.01330	1.333	0.00889						
T6							2.000	0.00667				
T7							1.000	0.00000	1.333	0.00667	1.333	0.00667

UL: unlimited, AA: available as and when required
α in hours, β in hours/kg

Instance P4 (from Maravelias and Grossmann[45])

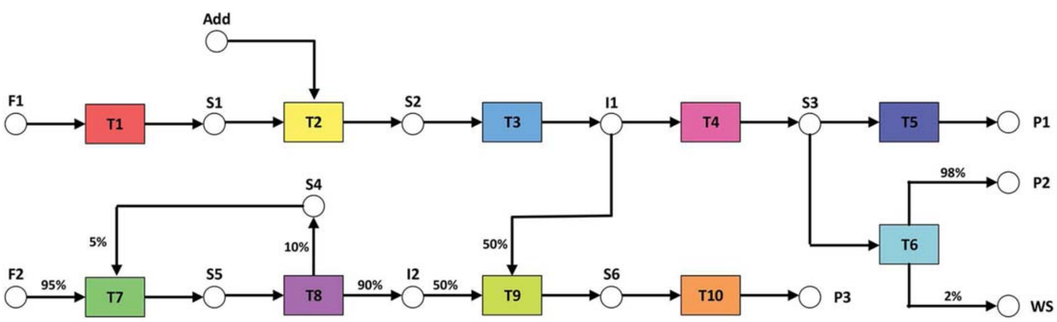


States:	F1	F2	I1	I2	I3	P1	P2
Capacity (kg)	1000	1000	200	100	500	1000	1000
Initial load (kg)	400	400					
Price ($/kg)						30	40
Demand (kg)						10	10
Horizon = 8 hours							

Units:	R1		R2		R3	
Minimum Load (kg)	40		25		40	
Maximum Load (kg)	80		50		80	
Tasks	α	β	α	β	α	β
T1	0.50	0.0250	0.50	0.4000		
T2	0.75	0.0375	0.75	0.0600		
T3					0.25	0.0125
T4					0.50	0.0250
Tasks (Utility Type)	γ	δ	γ	δ	γ	δ
T1(HS)	6	0.25	4	0.25		
T2(CW)	4	0.30	3	0.30		
T3(HS)					8	0.40
T4(CW)					4	0.50

α in hours, β in hours/kg, γ in uu/min, δ in uu/min/kg

Instance P5 (from Maravelias and Grossmann[45])



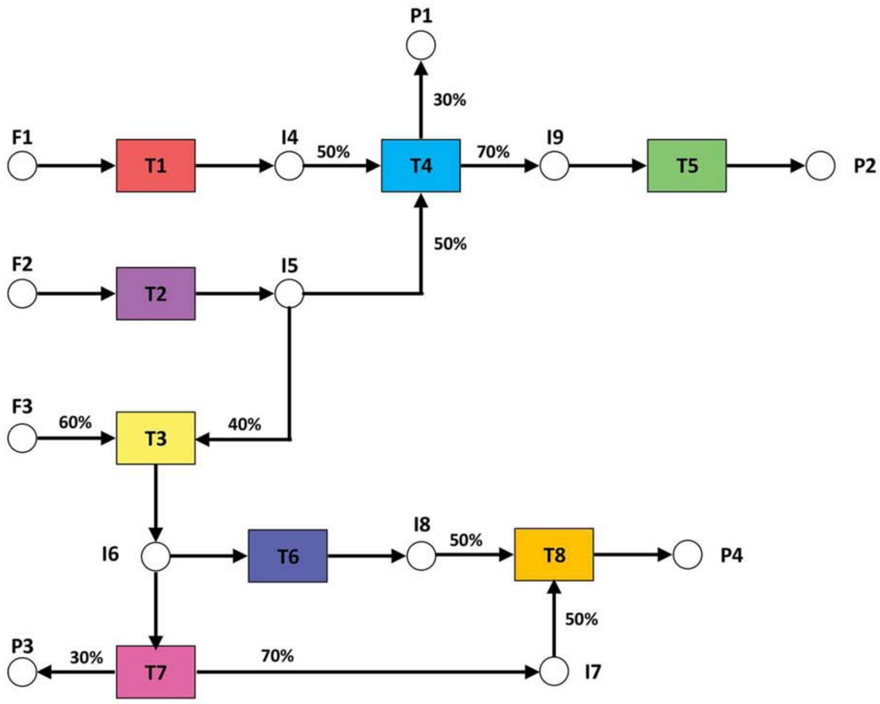
States:	F1	F2	S1	S2	S3	S4	S5	S6	I1	I2	P1	P2	P3
Capacity (kg)	UL	UL	ZW	ZW	15	40	ZW	ZW	UL	UL	UL	UL	UL
Initial load (kg)	100	100				10							
Price ($/kg)											1	1	1
Demand (kg)											1	1	1
Horizon = 8 hours													

Units:	U1		U2		U3		U4		U5		U6	
Minimum Load (kg)												
Maximum Load (kg)	5		8		6		8		3		4	
Tasks	α	β	α	β	α	β	α	β	α	β	α	β
T1	2											
T2			1									
T3					1							
T4	1											
T5							2					
T6							2					
T7									4			
T8											2	
T9									2			
T10											3	
Tasks (Utility Type)	γ	δ	γ	δ	γ	δ	γ	δ	γ	δ	γ	δ
T1(LPS)	3	2										
T2(CW)			4	2								
T3(LPS)					4	3						
T4(HPS)	3	2										
T5(LPS)							8	4				
T6(HPS)							4	3				
T7(CW)									5	4		
T8(LPS)											5	3
T9(CW)									5	3		
T10(CW)											3	3

UL: unlimited, ZW: zero-wait state

α in hours, β in hours/kg, γ in uu/min, δ in uu/min/kg

Instance P6 (from Ierapetritou and Floudas[52])

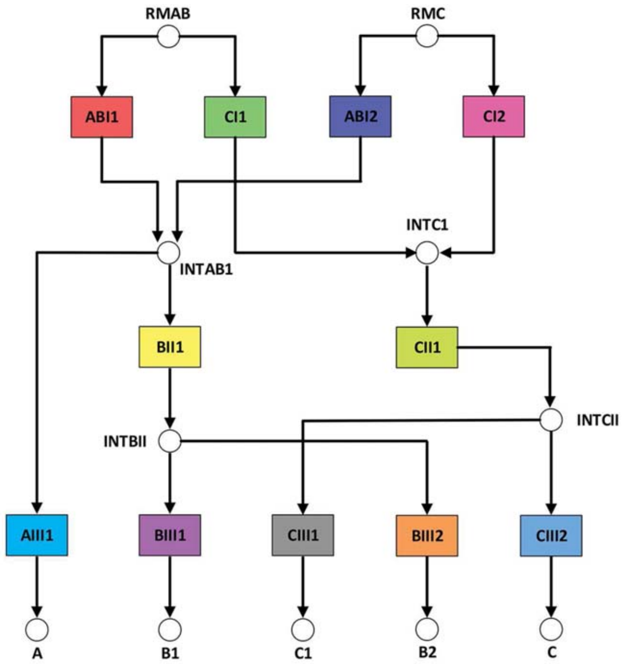


States:	F1	F2	F3	I4	I5	I6	I7	I8	I9	P1	P2	P3	P4
Capacity (kg)	UL	UL	UL	1000	1000	1500	2000	1000	3000	UL	UL	UL	UL
Initial load (kg)	AA	AA	AA										
Price ($/kg)										18	19	20	21
Demand (kg)										6000	8000	2000	8000
Horizon = 8 hours													

Units:	R1		R2		R3		R4		R5		R6	
Minimum Load (kg)												
Maximum Load (kg)	1000		2500		3500		1500		1000		4000	
Tasks	α	β	α	β	α	β	α	β	α	β	α	β
T1	0.667	0.000667										
T2							0.667	0.000667				
T3			0.667	0.000667								
T4					0.667	0.000667						
T5											0.667	0.000667
T6									0.667	0.000667		
T7			0.667	0.000667								
T8											0.667	0.000667

UL: unlimited, AA: available as and when required
α in hours, β in hours/kg

Instance P7 (from Pantelides[77])



States:	RMAB	RMC	INTAB1	INTC1	INTBII	INTCII	A	B1	C1	B2	C2
Capacity (kg)	500	500	20	16	16	16	500	500	500	500	500
Initial load (kg)	500	500									
Price ($/kg)							80	50	70	100	50
Demand (kg)							2	2	2	2	2

Horizon = 11 hours

Units:	UI1		UI2		UII1		UIII1		UIII2	
Minimum Load (kg)	3		4		5		4		5	
Maximum Load (kg)	6		8		10		8		10	
Tasks	α	β	α	β	α	β	α	β	α	β
ABI1	3.250	0.25								
CI1	3.200	0.30								
ABI2			3.500	0.20						
CI2			4.075	0.20						
BII1					3.250	0.15				
CII1					3.125	0.20				
AIII1							5.050	0.25		
BIII1							4.750	0.30		
CIII1							6.000	0.25		
BIII2									4.900	0.30
CIII2									5.750	0.20

α in hours, β in hours/kg

Manuscript received Oct. 31, 2015, and revision received Jan. 17, 2016.