



# Optimization with uncertainties: a scheduling example

Máté Hegyháti<sup>1</sup> · Krisztián Attila Bakon<sup>2</sup> · Tibor Holczinger<sup>2</sup>

Accepted: 25 April 2023  
© The Author(s) 2023

## Abstract

The optimization of manufacturing systems is rarely a deterministic task in practice, as uncertainties of various origins often have significant impact on the feasibility or quality of a production plan. The decision making process is often a complex multi-stage mechanism that anticipates and reacts to numerous parameter changes, both internal and external. Such optimization problems and the corresponding approaches are often classified and categorized based on the features of the addressed uncertainties and the desired objective. The goal of this paper is twofold. First, a systematic approach is presented to enumerate the possible optimization problems for a given set of uncertainties and degree of freedom in the planning. This approach is illustrated via a scheduling example that is inspired by literature case studies and focuses on financial objectives with varying demands and prices in the market. Then, selected problem classes identified by the approach are discussed in detail, illustrating and examining the available techniques to tackle them, citing literature studies applying the same or similar approaches.

**Keywords** Scheduling · Market uncertainties · Stochastic optimization

## List of symbols: Notations of the problem data and proposed models

### Sets

$U$  Set of available processing units  
 $P$  Set of possible products

---

✉ Krisztián Attila Bakon  
bakon.krisztian@pen.uni-pannon.hu

Máté Hegyháti  
hegyhati.mate@uni-sopron.hu

Tibor Holczinger  
holczinger.tibor@pen.uni-pannon.hu

<sup>1</sup> Institute of Informatics and Mathematics, University of Sopron, Bajcsy-Zsilinszky utca 9., Sopron 9400, Hungary

<sup>2</sup> Department of Applied Informatics, University of Pannonia, Zrinyi u. 18., Nagykanizsa 8800, Hungary

$I$	Set of tasks
$S$	Set of scenarios
$I_p \subseteq I$	Set of tasks to produce product $p \in P$
$I_i^- \subset I$	Set of predecessor tasks for task $i \in I$
$U_i \subseteq U$	Set of units capable to carry out task $i \in I$
$C$	Set of batch number configurations feasible within the time horizon

### Parameters

$h$	$[h]$ time horizon
$P(s)$	Probability of scenario $s \in S$
$pt_{i,u}$	$[h]$ processing time of task $i \in I$ in unit $u \in U_i$
$b_p$	$[t]$ (maximal) batch size for product $p \in P$
$lb_p$	$[t]$ lower bound for demand of product $p \in P$
$ub_p$	$[t]$ upper bound for demand of product $p \in P$
$d_p$	$[t]$ market demand for product $p \in P$
$d_{p,s}$	$[t]$ expected demand for product $p \in P$ in scenario $s \in S$
$s_p$	$[cu/t]$ selling price for product $p \in P$
$s_{p,s}$	$[cu/t]$ selling price for product $p \in P$ in scenario $s \in S$
$o_p$	$[cu/t]$ overproduction cost for product $p \in P$
$o_{p,s}$	$[cu/t]$ overproduction cost for product $p \in P$ in scenario $s \in S$
$u_p$	$[cu/t]$ underproduction cost for product $p \in P$
$u_{p,s}$	$[cu/t]$ underproduction cost for product $p \in P$ in scenario $s \in S$
$n_{p,c}$	Number of batches produced from $p \in P$ in configuration $c \in C$

### Variables

$b_p \geq 0$	Variable for batch size for product $p \in P$
$q_p \geq 0$	Variable for produced quantity of product $p \in P$
$q_p^o \geq 0$	Variable for overproduction of product $p \in P$
$q_{p,s}^o \geq 0$	Variable for overproduction of product $p \in P$ in scenario $s \in S$
$q_p^u \geq 0$	Variable for underproduction of product $p \in P$
$q_{p,s}^u \geq 0$	Variable for underproduction of product $p \in P$ in scenario $s \in S$
$x_{c,s}$	Binary variable denotes whether batch configuration $c \in C$ would be selected in scenario $s \in S$
$w_s \geq 0$	Aggregated objective variable for scenario $s \in S$

### Notations for problem class enumeration

$\mathcal{I}$	Set of input parameters
$\mathcal{D}$	Set of decisions
$\mathcal{I}^k$	Set of input parameters become available after stage $k$
$\mathcal{D}^k$	Set of decisions made in stage $k$
$\mathcal{I}^P$	All process data necessary for scheduling
$\mathcal{D}^S$	Set of decisions related to scheduling
$\mathcal{I}^M$	Set of market related parameters
$\mathcal{D}^B$	Set of decisions determining the size of each batch
$v^B$	Set of variables represent or are infected by decisions from $\mathcal{D}^B$

$\bar{v}^B$	Set of variables do not represent or are not infected by decisions from $\mathcal{D}^B$
$c^B$	Set of constraints contain some $v^B$ variables
$\bar{c}^B$	Set of constraints do not contain any $v^B$ variable
$v_k$	Number of variables affected by decision $\mathcal{D}^k$
$s_k$	Number of scenarios which approximate $\mathcal{I}^k$

## 1 Introduction and literature review

Real-life operation and optimization of production systems almost always have to deal with several uncertainties. Scheduling problems arising in the industry are no exception. Many papers in the literature discuss various approaches to tackle these uncertainties in both scheduling and other fields of process optimization.

The goal of this paper is twofold:

- First, a generalized way of looking at the decision making process is presented. Based on this aspect, a systematic approach can be derived to enumerate the possible problem classes regard to the sets of inputs given and decisions to be made.
- This approach is demonstrated with an illustrative scheduling problem class that aims to maximize profit. Some of the generated problem classes feature uncertainties, and the possible techniques to address them are discussed for some classes.

The paper is structured as follows: the rest of the section provides a literature review of batch process scheduling (Sect. 1.1), followed by a brief review of addressing uncertainties in the production industry (Sect. 1.2). The end of this section, Sect. 1.3 discusses advancement on the intersection of the two fields above, scheduling with uncertainties.

Section 2 provides the definition of an illustrative scheduling problem class that will be used later for demonstration. The proposed abstraction to decision making processes is presented in Sect. 3. Section 3.2 details the problem class enumeration procedure based on this abstraction. The problem class from Sect. 2 is used as an illustrative example.

Selected problem classes (referred to as cases) are investigated in detail in Sect. 4. These cases are selected in a way to highlight deterministic, reactive, and preemptive techniques in scheduling. At the end of the paper, the conclusions are drawn.

### 1.1 Batch process scheduling

Scheduling is an often used decision-making process in many and different kinds of manufacturing and services industry, where in a given time period the allocation of resources to different tasks is dealt with the goal to optimize one or more objectives. Resources can take many different forms, tasks may be the operations, activities, stages or executions of different kinds of processes. Objectives can also take several shapes and forms, like minimization of completion time of several tasks. In

industrial environments, the processes take place within a given framework, but due to the different environmental effects and other factors, there is always the possibility of a certain degree of uncertainty.

Multi-purpose batch production plants are characterized by flexibility and variety in production, which can be used to meet customer demands and with the variety of different production paths to reduce production costs. Due to their diversity, they can be categorized in several ways (Méndez et al. 2006). Problem definition consists of three types of information, which are the recipe per product, the assignment of tasks to equipment, and the quantities to be produced from the products. The recipe contains the minimum information needed to manufacture the product, which is a unique description of specific product manufacturing requirements. Depending on user requirements, the following four levels are used in batch processes:

- General recipe identifies the raw materials, their quantity and the required processing tasks.
- Site recipe is a general recipe supplemented with site information, which means site-specific information
- Master recipe contains information about the required equipment, raw materials with the required quantities and the process of product production.
- Control recipe contains additional information about the production units. This recipe is derived from the master recipe.

Of the four types, the master recipe is applied during the batch process scheduling, which can be represented by a graph where nodes denote the production tasks and arcs denote the relationships between them (Sanmartí et al. 2002). Another important parameter of batch scheduling problems is the intermediate storage method. The intermediate storage options and implementation depend on the situation, production policy and the product itself. If the products can be stored without restriction, the Unlimited Intermediate Storage (UIS) policy is applied, whereas if storage is not an option or not possible, it is called Non-intermediate Storage (NIS) policy. In the case where the intermediate storage capacity is finite, it means Finite Intermediate Storage (FIS) or Common Intermediate Storage (CIS) policy. The strictest storage policy is Zero Wait (ZW), when the rule does not allow intermediate storage, even in the processing units. These storage rules may be present in a mix, called Mixed Intermediate Storage (MIS) policy (Hegyháti and Friedler 2010).

## 1.2 Uncertainty in process optimization

As the complexity of the processes increases, so does the likelihood of uncertainty, which requires special attention. There are several solutions for dealing with uncertainty, but to use them first, it is necessary to identify the uncertainty, which provides a preliminary determination and understanding of the hypothetical cause. Each can be classified into separate categories according to certain aspects. The source of

uncertainty can be a dynamic event, which has the following categories (Suresh and Chaudhuri 1993):

- Workpiece related events: random job arrival, non-deterministic processing time
- Machine related events: machine breakdowns, loading limits
- Operation related events: operation delays, unstable production
- Other events: operator absence, defective materials

Another approach is that uncertainty is not simply a lack of knowledge. On the one hand, uncertainty can arise from inadequate information, which can be of the following types: inexactness, unreliability, and border with ignorance. On the other hand, there are situations where information is abundant, yet there is uncertainty (Van Asselt and Rotmans 2002). Expanding and decreasing information can further increase or decrease uncertainty. Increasing the necessary information in complex processes can put already established processes into a new perspective, about which additional knowledge can be used to identify additional uncertainties, which may mean that our knowledge is limited and insufficient, or the process is more complex than it used to be (Sluijs 1997). Uncertainties resulting from this knowledge-based deficiency can also appear in the model in several ways. Several general locations can be identified to establish the occurrence of uncertainty in the outcome (Walker et al. 2003):

- In the problem-making phase of the model, the **context** is defined, which is particularly important in the decision support practice (Dunn 2018).
- The **model uncertainty** can be divided into two parts: the model structure uncertainty and the model technical uncertainty, which is the uncertainty arising from the computer solution (Sluijs 1997).
- It is sometimes worthwhile to classify the **inputs** of the model into controllable and uncontrollable categories, depending on whether it is possible to influence the values of the input variables.
- The method used to calibrate the model parameters is associated with the **uncertainty of the parameters** (Harremoës and Madsen 1999).
- The output of the model may include uncertainty, referenced as **model outcome uncertainty**, which can appear to the interest to the decision maker as cumulative uncertainty (Banks 1993).

Additional uncertainty factors must also be taken into account, but the main factors can be summarized based on the aforementioned (Bakon et al. 2022). The identification of uncertainty is crucial in constructing the appropriate model, as ignoring it can lead to misleading or bad results, the consequences of which are revealed after decisions are made.

### 1.3 Uncertainty in batch process scheduling

To deal with uncertainty, it is necessary to distinguish between static and dynamic scheduling. In the case of a static schedule, the data are known in advance and the schedule is fixed. The origin of the uncertainty can arise from several factors, as mentioned earlier, therefore a dynamic schedule is required to address it as the schedule plan must be modifiable.

- *Proactive approach* To address uncertainty, it is achieved through modeling solutions or by optimizing its performance under different scenarios (Liu et al. 2007; O'Donovan et al. 1999).
- *Reactive scheduling* A preliminary schedule is required to start, which is modified or re-optimized when the uncertainty parameters are realized or an unexpected event occurs (Mihoubi et al. 2021).
- *Stochastic scheduling* In the case of available information about the uncertainty, stochastic variables can be used with probabilistic description when implementing a deterministic model (Ma et al. 2021).
- *Fuzzy programming* In the event when historical data is not available or the probability distribution is unknown, fuzzy programming can handle the uncertainty. Instead of using stochastic variables, a fuzzy programming implementation is used (Balasubramanian and Grossmann 2003; Prade 1979).
- *Robust scheduling* During robust scheduling steps, the impact of disturbances on performance should be minimized by incorporating uncertainties into the model describing the scheduling task. This makes the schedule less sensitive to various distractions. Two fundamental aspects of optimization are solution robustness and quality robustness (Herroelen and Leus 2005).

In the case of a production system, the rework process is important if several aspects support the need to achieve better results. In case of semi-finished and finished products, when the objective is to minimize inventory cost, determining the optimal batch size is of paramount importance (Sarker et al. 2008). To reduce inventory cost if inventory is depleted at certain intervals in the production stage, an approach can be made that provides a solution for equal- and unequal-sized batch deliveries (Glock 2010), which can result in lower total costs. In the case of a two-stage production system, supplementing the solution with a variable that can properly measure the production rate may be a good solution to reduce excess inventory (Glock 2011). There are several other solutions in the literature where the batch sizing problem is solved. Such a solution could be a holistic scheduling algorithm (Bicheno et al. 2001), or in the case of a single-stage system due to defective items (Jamal et al. 2004) and in the case of multi-stage manufacturing systems solving the optimal batch sizes (Sarker et al. 2008). Investigating the implementation of Just-In-Time, where batch-sizing decisions (McKenzie and Jayanthi 2007) have an impact on different demand patterns from an operational and financial point of view adequate results were achieved. Similarly, if the Just-In-Time scheduling problem is approached from a general due date perspective, solution is achieved by treating it as an integrated batch sizing problem (Hazır and Kedad-Sidhoum 2014).

Distributionally robust optimization based models can provide sufficient flexibility for random demands to adjust resource actions (batch sizes) in multi-stage decision-making (Shang and You 2018). Adjusting batch sizes to dynamic events entails frequent adjustments of job sequence, which can be avoided with a buffering mechanism (Qin et al. 2018). In Chemical, Petrochemical and Pharmaceutical Industries, risk mitigation approaches have the ability to explicitly take into account parameter uncertainty within a mathematical framework (Verderame et al. 2010). Demand uncertainty can be addressed with stochastic models, which can allow uncertain product demand correlation with the objective of profit maximization (Petkov and Maranas 1997). In short-term scheduling of multiproduct and multipurpose batch plants, the schedule from a multiperiodic formulation has much bigger robustness compared to the one determined based on the vertexes of the uncertainty range, considered as scenarios (Vin and Ierapetritou 2001). For due date changes, adopting discrete-time simulation method for minimizing responsiveness and tardiness is a viable solution in precast concrete structures (Kim et al. 2020). With prefabricated construction productivity and on-time delivery of precast components, a real case study can be conducted to test the validity of a two-level rescheduling model and achieve significant cost savings (Wang and Hu 2018). The applicability of two-stage adaptive robust optimization approaches are demonstrated on two case studies by introducing uncertain parameters (processing time, order demand) into the single-stage deterministic model and reformulating it into a two-stage batch scheduling model (Shi and You 2016). Polynomial-time efficient algorithms can be proposed for single-machine scheduling problems with serial and parallel batches under uncertain processing times using worst-case scenario analysis (Wu et al. 2023). For a single machine with availability constraints, where the machine suffers from unexpected breakdowns, a genetic algorithm by integrating a run-based preventive maintenance into the production scheduling model can optimize system robustness and stability with experimental results (Lu et al. 2015). Demand uncertainty may be represented by scenario trees in a multi-stage stochastic programming model with lot-sizing (batch size for each product) and overall system costs (production cost, setup cost, inventory cost, backlog cost)(Hu and Hu 2018). The results of a case study with two-stage stochastic programming model for a manufacturing company producing braking equipment in automotive industry has showed that production quantity is more sensitive to the uncertainty than production sequence (Hu and Hu 2016). A combination of a mixed-integer linear programming and an agent-based reactive scheduling method can be used to minimize total cost of inventory holding and production setup, where processing times are uncertain (Chu et al. 2015). Rescheduling is traditionally viewed as an approach if uncertainty is present, but the event-triggered rescheduling has some shortcomings (constantly incoming new information), which can be addressed by approaching the rescheduling as an online problem (Gupta et al. 2016).

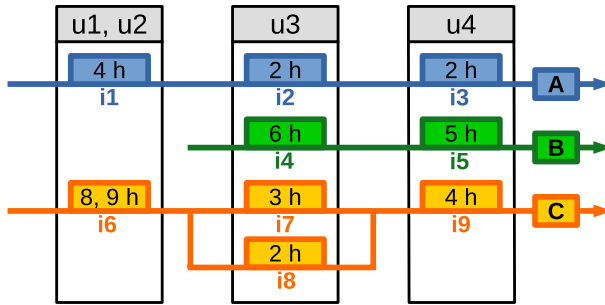


Fig. 1 Example instance of the illustrative problem class

## 2 Illustrative scheduling problem class

The problem class selected for illustration is the mid-term scheduling of a production plant with an economic objective. The goal is to find a production schedule that fits into the predefined time horizon and maximizes the expected profit that derives from the following components:

- Income by selling the produced products
- Penalties if market demand is not met, or exceeded

There are several possible products produced via precedential recipes by the available units. For the sake of simplicity, processing is the only step with time requirements. Cleaning, transportation, etc. are considered to be instantaneous.

The core sets and parameters of the problem can be summarized as:

$U$  set of available processing units

$P$  set of possible products

$b_p$  maximal batch size for  $p \in P[t]$

$d_p$  market demand for  $p \in P[t]$

$s_p$  selling price for  $p \in P[cu/t]$

$o_p$  overproduction penalty for  $p \in P[cu/t]$

$u_p$  underproduction penalty for  $p \in P[cu/t]$

$I_p$  set of tasks to produce  $p \in P$ , and let  $I = \cup_{p \in P} I_p$

$I_i^- \subset I$  set of predecessor tasks for  $i \in I$

$U_i \subseteq U$  set of units capable to carry out  $i \in I$

$pt_{i,u} \geq 0$  processing time of  $i \in I$  in  $u \in U_i[h]$

$h \geq 0$  is the time horizon  $[h]$

To help to understanding the notations, an instance of this problem class is presented. Figure 1 shows the recipe of this instance which contains 3 products ( $P = \{A, B, C\}$ ). Product A has a sequential production recipe with 3 production steps ( $I_A = \{i1, i2, i3\}$ ), product B also has a sequential production recipe with 2 production steps ( $I_B = \{i4, i5\}$ ) and the recipe of product C has 4 tasks



**Table 1** Parameters of products for the example problem instance shown in Fig. 1. The units of measure are  $[t]$ ,  $[t]$ ,  $[cu/t]$ ,  $[cu/t]$  and  $[cu/t]$  for parameters  $b_p$ ,  $d_p$ ,  $s_p$ ,  $o_p$  and  $u_p$  respectively

Product $p$	$b_p$	$d_p$	$s_p$	$o_p$	$u_p$	$I_p$
A	7	29	100	2	3	$\{i1, i2, i3\}$
B	4	20	120	5	3	$\{i4, i5\}$
C	8.5	78	150	1	1	$\{i6, i7, i8, i9\}$

**Table 2** Parameters of tasks for the example problem instance shown in Fig. 1

Task $i$	Task		$pt_{i,u}[h]$			
	$I_i^-$	$U_i$	$u1$	$u2$	$u3$	$u4$
$i1$	$\emptyset$	$\{u1, u2\}$	4	4		
$i2$	$\{i1\}$	$\{u3\}$			2	
$i3$	$\{i2\}$	$\{u4\}$				2
$i4$	$\emptyset$	$\{u3\}$			6	
$i5$	$\{i4\}$	$\{u4\}$				5
$i6$	$\emptyset$	$\{u1, u2\}$	8	9		
$i7$	$\{i6\}$	$\{u3\}$			3	
$i8$	$\{i7\}$	$\{u3\}$			2	
$i9$	$\{i7, i8\}$	$\{u4\}$				4

( $I_C = \{i6, i7, i8, i9\}$ ) and it is not sequential. The recipe contains 4 processing units ( $U = \{u1, u2, u3, u4\}$ ), which can be used for production. Tasks  $i1$  and  $i6$  can be performed by two processing units ( $u1, u2$ ) where the processing time of task  $i1$  is 4 h and the processing time of  $i6$  is 8 or 9 h using  $u1$  or  $u2$ , respectively. Task  $i6$  generates 2 intermediates and task  $i9$  has 2 prerequisite tasks ( $i7, i8$ ).

Figure 1 does not contain all information of the system, like cost parameters. The parameters of the products and tasks are summarized in Tables 1 and 2, respectively, and the time horizon ( $h$ ) is 55 h.

### 3 Proposed abstraction for decision processes

#### 3.1 General concepts

From an abstract point of view, the difference between deterministic and stochastic problems comes down to the timing of the revelation of the values of input parameters and when certain decisions have to be made.

In the deterministic case, the values of all of the input parameters are known in advance, i.e., their values are revealed before any decision is made. In contrast, stochastic problems have some input data, whose value is revealed later than others, and some decisions have to be made between these two timings. The data that reveals themselves after some decisions have been made are often called *uncertain*.

If all of the decisions have to be made before the uncertain data reveal themselves, the approach is non-reactive. Robust and preventive approaches belong to this category, and the differentiation often lies in the selected objective function. However, if some decisions can be made or altered after an uncertain input is revealed, the approach is reactive.

Note, that unexpected events, such as unit shutdowns are within the scope of the description above. Such events could be considered as new, however, they can also be looked at as predefined parameters that reveal their values later. For example, possible shutdowns can be considered as boolean parameters for each task, which either turn out to be 0 or 1 at the time of their execution.

More often than not, if uncertain values are revealed later, there is still some information available on their possible values. These can be based on a predictive simulation, historical data, assessment from experts, etc. The level of detail of this information can vary a lot, and can influence the approach to be used. In this paper we will focus on the following options:

*Estimated value*- Only a single value is prognosed, which may be the mean or median value.

*Interval* - An interval is given for the possible values of the parameter, but no information is available about the probability distribution within that interval.

*Probability distribution* - A continuous or discrete probability distribution function is given for uncertain parameters.

Naturally, not all of the above are applicable for all cases, e.g., an interval prediction for a binary variable is trivial. However, the detailedness of this information can highly influence the possible objectives and approaches that can be developed. For example, without an interval, it is theoretically impossible to develop a robust approach with the worst case as an objective or bound. Similarly, expected values can only be incorporated into the approach if some distributions on the contributing values are available. From the practical point of view, the difficulty to assess this data can vary a lot based on the parameter, and may be unreasonably expensive to do so.

### 3.2 Enumeration procedure on the illustrative problem class

From the theoretical point of view, if there are  $n$  inputs and  $m$  decisions, there are  $(n + m)!$  different orders of them, that could be considered as different problems. From the planners' perspective, some of these orders can be considered the same. A decision can only rely on data that has already been revealed, therefore if two decisions are neighbors in an ordering, swapping them will not alter the information available for either of them. This can reduce the number of cases by magnitudes.

Moreover, in practice, input parameters are often revealed in larger groups, and some decisions also have to be made simultaneously. For illustration, let us split

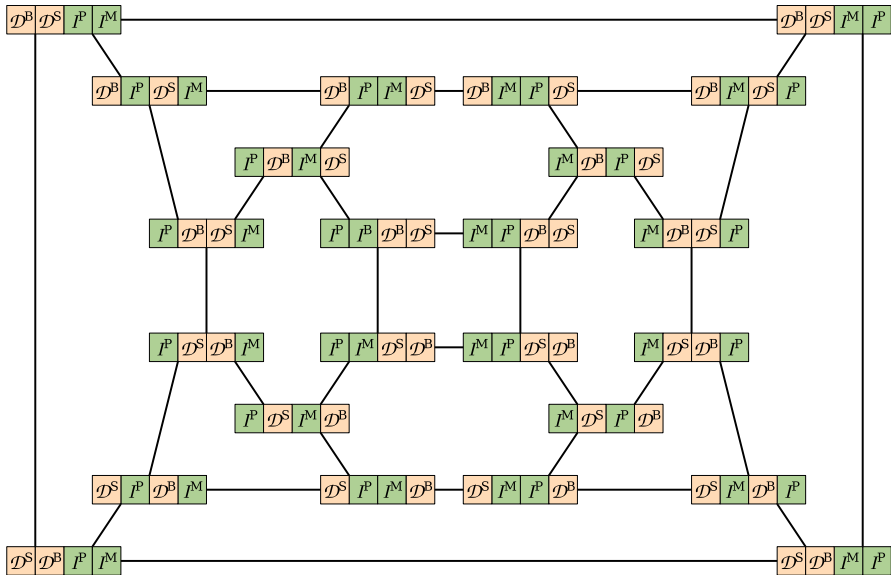


Fig. 2 Different cases for the illustrative example

both the decisions and inputs of the considered illustrative problem class into two such groups:

- $\mathcal{D}^S$ : the set of decisions related to scheduling, such as choosing the number of batches to be produced, and the assignment, sequencing, and timing of each task.
- $\mathcal{D}^B$ : the set of decisions determining the size of each batch.
- $\mathcal{I}^P$ : all the process data including recipes, processing times, task-unit compatibilities, etc. ( $U, P, b_p, I_p, I_i^-, U_i, pt_{i,u}, h$ ).
- $\mathcal{I}^M$ : market related parameters describing demand, sell price, and costs for over- and underproduction ( $s_p, o_p, u_p, d_p$ ).

The idea behind this separation is that while  $\mathcal{I}^M$  may change on a daily basis,  $\mathcal{I}^P$  is more stable. Similarly, a facility often prepares the schedule ( $\mathcal{D}^S$ ) in advance for a week or so, and may only do small adjustments on a daily basis, which in this case is considered to be the sizing of the batches ( $\mathcal{D}^B$ ).

Having these four groups,  $4! = 24$  different orders could have been considered, as illustrated in Fig. 2. Each order represents a different problem class, which will be referred to as cases in the rest of the paper. For a more expressive visualization, decision-items have orange background color, while information-items have green. The connected cases can be transformed into each other by swapping two neighboring items. The connections by swapping non-neighboring items are omitted from the Figure for transparency, as it would mean an additional 36 edges.

As mentioned before, some of these cases can be considered identical, as the decision-items have the same information available for them. These groups of cases are indicated by gray areas in Fig. 3. For example, the following four orders all

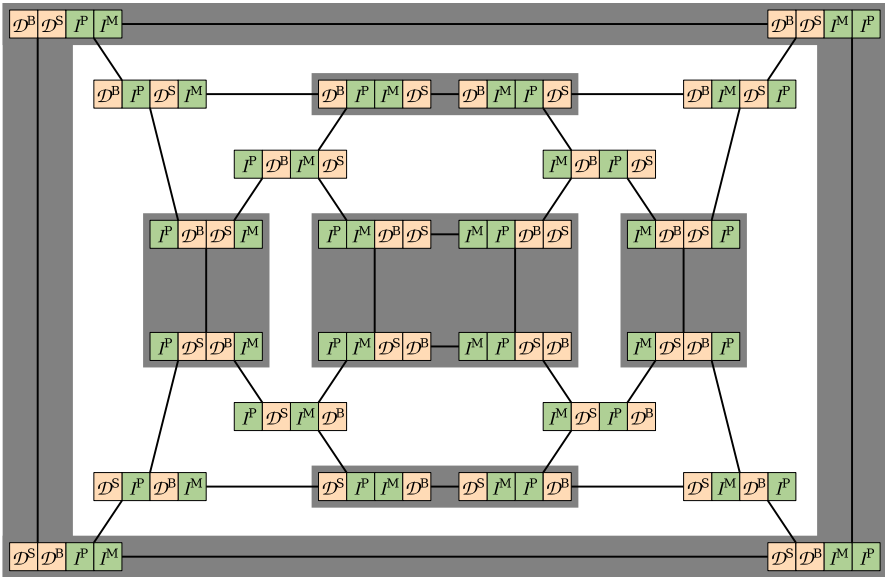


Fig. 3 Equivalent cases for the illustrative example

Table 3 Considered cases for the illustrative problem class from Sect. 2

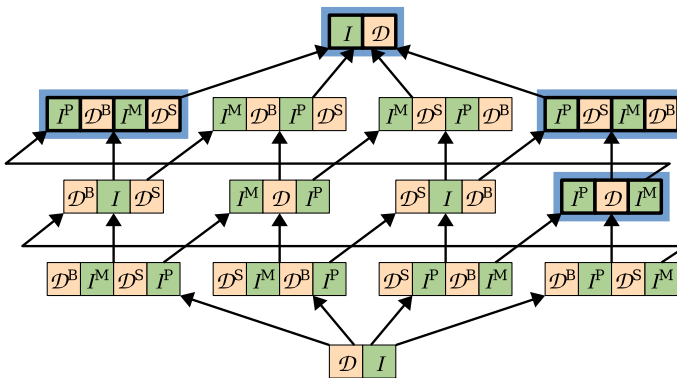
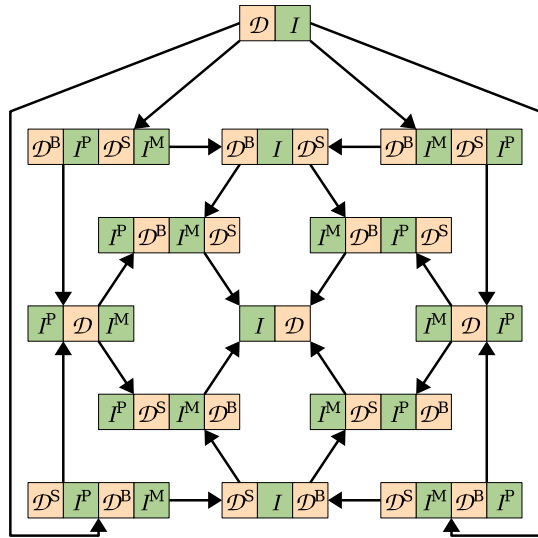
Case	Order	Case	Order
1	$ID$	8	$DI$
2	$I^P D^S I^M D^B$	9	$D^S I^P D^B I^M$
3	$I^P D^B I^M D^S$	10	$D^S I^M D^B I^P$
4	$I^P D I^M$	11	$D^S I D^B$
5	$I^M D^S I^P D^B$	12	$D^B I^P D^S I^M$
6	$I^M D^B I^P D^S$	13	$D^B I^M D^S I^P$
7	$I^M D I^P$	14	$D^B I D^S$

describe the deterministic case:  $I^P I^M D^S D^B$ ,  $I^M I^P D^S D^B$ ,  $I^P I^M D^B D^S$ ,  $I^M I^P D^B D^S$ , which can be simplified just as  $ID$ .

Removing these duplications, 14 cases remain, which are summarized in Table 3. Not all of the cases have equal practical significance:

- Scheduling before any process data are available is meaningless, which eliminates cases 5, 7–11, 13.
- In cases 5–7, 10, and 13, market parameters are known earlier than process parameters. While not completely impossible, these cases will be omitted from discussion as market environment data are usually much more volatile than process data.
- The remaining are cases 12 and 14, where batch sizing data have to be made a-priori. A valid scenario that could be described by these cases is when the sizes

**Fig. 4** Strongness directed network of merged cases for the illustrative example

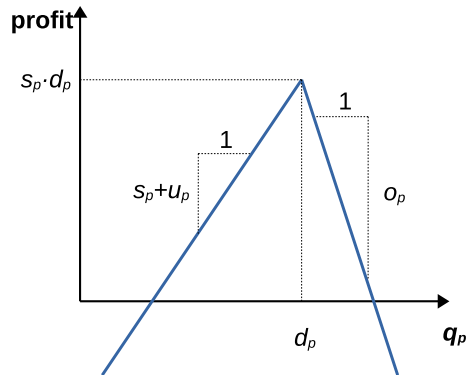


**Fig. 5** Strongness hierarchy of merged cases for the illustrative example

of the units have to be decided in the design phase of a plant, and scheduling decisions have to be made much later without the option of scaling the processes down. From the scheduling point of view, these are cases when the batch sizes are fixed.

Note, that in some cases, the decision maker is in a better position than in others. Clearly, the deterministic case ( $ID$ ) is the most advantageous among all of them, when all of the information is available before any decision has to be made. This relation can be defined formally and forms a partial ordering, with the deterministic case being the maximal element, and case 8 as the minimal. In

**Fig. 6** The profit function of a product



the current context, it is more expressive to talk about stronger and weaker cases instead of larger and smaller ones. Informally, a case is stronger than another, if the decision maker has always at least as much or more information available for all of the decisions than in the other case. (Formally, case  $X$  is stronger than case  $Y$ , if for all  $\mathcal{D}$ -type items, the set of  $\mathcal{I}$ -type items preceding it in case  $Y$  is a subset of those in case  $X$ .) Obviously, identical cases are equally strong and form equivalence classes.

In Fig. 4, this ordering is represented by the direction of the arcs. Transitive arcs are omitted for transparency.

Another layout of the same network in Fig. 5 better indicates the strongness hierarchy, and the selected cases for Sect. 4 are also indicated.

## 4 Evaluation of selected cases

The 4 selected cases will be discussed in detail in this section. The main aspects of the investigation will focus on

- *Complexity* - How many variables or constraints a proposed model may require compared to the baseline.
- *Reusability* - In a practical setting, how much of the effort needs to be repeated if the problem is solved on a daily basis with changed data.
- *Generalizability* - How easy, solvable it is if the profit function is generalized.

### 4.1 $ID$ deterministic case

This is the strongest case that can serve as a baseline. All information is known in advance, thus the model is deterministic.

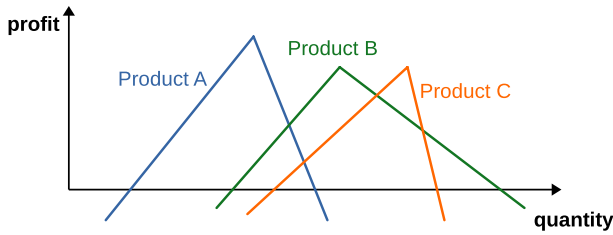


Fig. 7 The profit function of three products

The problem can be formulated as a mixed-integer linear programming problem easily. Depending on the type of model used (precedence-based, discrete, continuous, etc.) the number of continuous and binary variables, as well as the number of constraints may vary a lot.

The profit function according to the base problem defined in Sect. 2 is:

$$\sum_{p \in P} (s_p \cdot \min(q_p, d_p) - u_p \cdot \max(0, d_p - q_p) - o_p \cdot \max(0, q_p - d_p))$$

where  $q_p$  indicates the produced quantity from product  $p \in P$ . This is a piecewise linear function for each product with two pieces: above and below  $d_p$ . For better understanding Fig. 6 shows the profit function of a product and Fig. 7 shows the profit function of 3 products.

This function can be modeled in different ways: a binary variable may be introduced for each product for the selection of the segments, or two non-negative continuous variables can be introduced for under- and overproduction, and the profit is reformulated as:

$$\sum_{p \in P} (s_p \cdot d_p - (u_p + s_p) \cdot q_p^u - o_p \cdot q_p^o)$$

where  $q_p^o$  and  $q_p^u$  are non-negative continuous variables that model  $\max(0, q_p - d_p)$  and  $\max(0, d_p - q_p)$ , respectively.<sup>1</sup> This also means that  $q_p$  must be replaced by  $d_p + q_p^o - q_p^u$  everywhere else in the model.

It is also important to note, that the evaluation of the profit (and thus  $o_p, u_p$  and  $s_p$ ) only appears in the objective function, and they have no influence on the other parts of the model. This has two beneficial consequences.

If the  $T^M$  variables change for the next day, the solution acquired before will still be feasible if the  $q_p^o$  and  $q_p^u$  are shifted with the change in demand. This means that with this simple preprocessing step, a feasible solution for the MILP model is available, providing an initial upper bound. Since this solution is primal feasible, the bound can be improved by running the second phase of the simplex algorithm to

<sup>1</sup> Note, that this formulation is only valid if  $o_p \geq 0$  and  $u_p \geq -s_p$ , which is generally the case, as both  $o_p$  and  $u_p$  are at least non-negative in a real scenario.

achieve higher profit by allowing to change only the mass related continuous variables, while keeping discrete scheduling variables and continuous timing variables fixed.

The other benefit is, that if a more complex profit evaluation is needed to model the real life example, properly it will only appear in the objective function. Thus, for example, if this function is quadratic in nature, the problem can still be formulated as a Mixed Integer Quadratic Programming (MIQP) problem (Bonami et al. 2018; Mencarelli and D'Ambrosio 2019). Modern solvers such as CPLEX or Gurobi can solve MIQP problems with some solver specific restrictions. For example in case of CPLEX the objective function can contain quadratic terms and it must be convex or the objective function can contain only quadratic terms which are products of binary variables (in this case, the objective function is not necessarily convex). Moreover, the constraints must be inequalities and the constraints that contain a quadratic term can be represented as second order cone programs (SOCP) or the quadratic term in a constraint involves only multiplication of binary variables.<sup>2</sup> In the case of multi-objective problems, formulating the problem as MILP or MINLP, Pareto optimal solutions can be achieved with the CPLEX or BARON solver (Capón-García et al. 2011).

## 4.2 $I^P D I^M$ market-sensitive scheduling and batch-sizing

In this case, one has to make the scheduling and sizing decisions before the market data reveals itself. Unlike in the deterministic case, it becomes important what kind of information is available about the the uncertain values.

### 4.2.1 Expected demand

The simplest option is if a value is estimated, and then the goal of the optimization is to maximize profit for that scenario. This situation is equivalent to the deterministic case from the mathematical point of view.

### 4.2.2 Demand interval

A slightly more detailed option is, if there is an interval  $[lb_p, ub_p]$  available for each uncertain variable. Then, for each production plan, a worst-case and a best-case scenario can be evaluated. If an estimated case is also available within the interval, an “expected-case” is also available. These values give rise to several optimization objectives, for example:

<sup>2</sup> <https://www.ibm.com/docs/en/icos/20.1.0?topic=optimization-solving-mixed-integer-programming-problems-quadratic-terms> Accessed 24 Oct 2021.



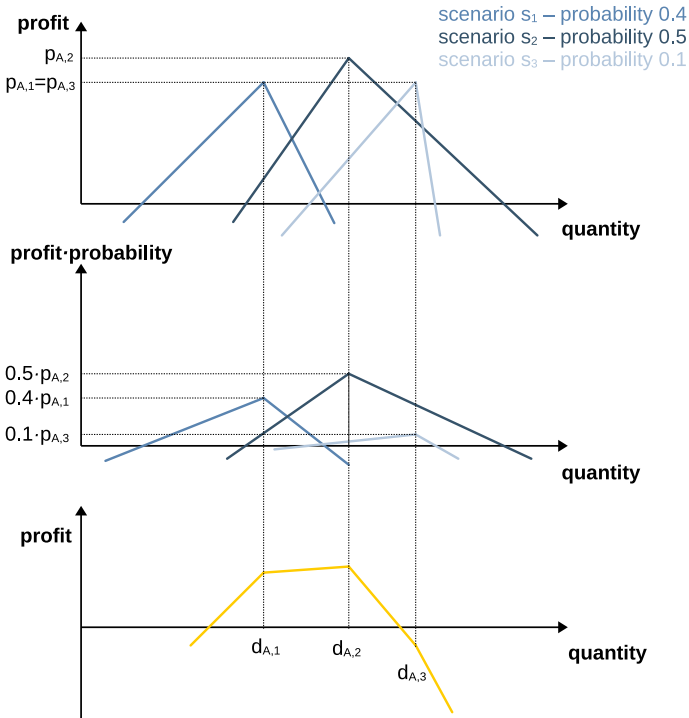


Fig. 8 The aggregate profit function of three scenarios of a product (product A)

- Maximizing expected- or best- case while having a lower bound on the worst-case
- Maximizing worst-case

Maximizing the worst-case scenario is simple, as it will happen at one of the ends of the given interval, whether  $q_p$  is inside it or not. Thus, one can add the profit expressions as in the deterministic case once with  $d_p = lb_p$  and then with  $d_p = ub_p$  for each product. Then, for each product a new profit variable is introduced as the lower bound of these two cases. These continuous variables are then summed in the maximization objective.

Implementing a lower bound on the worst case can be done in a similar fashion. The only difference is that the sum mentioned previously is present in a constraint with this lower bound. Maximization of the expected case is the same as the deterministic case. Maximizing the best case is similar, with the only difference that if  $q_p$  is within the interval, then  $d_p$  can be equal to it. If  $q_p < lb_p$ , then the profit should be calculated as if  $d_p = lb_p$ . The situation is similar if  $q_p > ub_p$ . This can be done easily if  $q_p^o$  models  $\max(0, q_p - ub_p)$  (instead of  $\max(0, q_p - d_p)$ ) and  $q_p^u$  does the same for  $\max(0, lb_p - q_p)$ . Note, that this modeling trick does not allow to use

$d_p + q_p^o - q_p^u$  instead of  $q_p$  in the other parts of the model anymore. One should add a new bounded  $q'_p \in [0, ub_p - lb_p]$  variable, and use  $d_p + q_p^o - q_p^u + q'_p$  instead.

### 4.2.3 Discrete demand distribution

In some cases, the probability distribution of the uncertain values can be estimated with a (finite) discrete distribution function. For the sake of simplicity, let us assume that there is a finite set of scenarios,  $S$ , and the expected values of all of the uncertain variables are given for all of the scenarios. In a more complex case, some sets of uncertain variables would have their scenarios independently. However, having independent scenario sets  $S_1, S_2, \dots$  would only translate to  $|S_1| \cdot |S_2| \cdot \dots$  global scenarios. For each  $s \in S$ , there is a probability assigned,  $P(s)$  such that  $\sum_{s \in S} P(s) = 1$ .

Evaluating the profit would require separate  $q_{p,s}^o$  and  $q_{p,s}^u$  variables for each scenario,  $s \in S$ , which would model  $\max(0, q_p - d_{p,s})$  and  $\max(0, d_{p,s} - q_p)$ , where  $d_{p,s}$  is the expected demand in the scenario. In the objective function there is just a weighted version of the original objective for each scenario:

$$\sum_{s \in S} P(s) \cdot \sum_{p \in P} \left( s_{p,s} \cdot d_{p,s} - (u_{p,s} + s_{p,s}) \cdot q_{p,s}^u - o_{p,s} \cdot q_{p,s}^o \right)$$

where  $s_{p,s}$ ,  $u_{p,s}$ , and  $o_{p,s}$  are the price, under- and overproduction costs for product  $p \in P$  in the scenario  $s \in S$ . Figure 8 shows an aggregate profit function of a product in case of three scenarios with probabilities 0.4, 0.5 and 0.1.

Thus the model requires  $2 \cdot (|S| - 1)$  additional continuous variables, and the constraints for setting them. Note, however, that these variables are strongly bound together, as for all  $s \in S$ ,  $d_{p,s} + q_{p,s}^o - q_{p,s}^u$  must represent the same value,  $q_p$  that is independent from  $s$ .

Incorporating continuous distribution functions into an MILP model is impossible except for trivial cases. Regardless of the detail of the information on the market data, if it changes, re-running the model is equally time consuming as it was in the deterministic case. Similarly, considering a more complex cost function is possible within similar limits as in the deterministic case.

Note, that the ideas above are general, not specific to the illustrative example. If there are uncertain variables and variables that depend on them, the scenario-based approach can be used in the same way even if those variables are binary decision variables.

Scenario-based approach is applied with uncertainty of the number of jobs using MILP and branch-and-price algorithm to minimize the expected costs (Wullink et al. 2004). Similarly, with uncertain time requirements of elective operations in theater scheduling decisions with the objective to maximize the expected profit is applied in (Freeman et al. 2016). Data-driven uncertainty set with probability distribution for market demand is used to maximize the net present value at each time period in Ning and You (2017).

### 4.3 $\mathcal{I}^P \mathcal{D}^S \mathcal{I}^M \mathcal{D}^B$ market-reactive batch-sizing

This case represents a middle point between the previously discussed cases, as it is clearly visible in Fig. 5. Compared to the deterministic case in Sect. 4.1, it is weaker, as scheduling decisions have to be made before market information is available. On the other hand, it is a stronger case than the market-sensitive one from Sect. 4.2, as the decision maker has the ability to react on the market information with the decisions about batch sizing.

If the possible outcomes from  $\mathcal{I}^M$  are finite, a two-stage model can be easily formulated. For example, having a finite set of scenarios,  $S$ , as discussed in a previous section. In such a situation, for each variable,  $x$  that either represents a decision from  $\mathcal{D}^B$  or a calculated value based on those variables, a set of variables,  $x_s$  has to be introduced, where  $s \in S$ . Obviously,  $q_p^u$  and  $q_p^o$  would be such derived variables, and any variables they derive from, possibly the independent sizing variables for the batches. Thus, instead of  $q_p^o$  and  $q_p^u$ , several ( $2 \cdot |S|$  to be precise) variables are introduced, denoted by  $q_{p,s}^o$  and  $q_{p,s}^u$ , respectively for  $s \in S$ . This is similar to expected profit maximization discussed in the previous chapter. However, a key difference is that these variables are more independent, as  $d_{p,s} + q_{p,s}^o - q_{p,s}^u$  (representing  $q_{p,s}$ ) can be different for each scenario.

Any constraint that involves any of these variables have to be added multiple times for all  $s \in S$ , as well. For example, if an objective variable,  $w_s$  is introduced, the following constraints may be added:

$$w_s = \sum_{p \in P} \left( s_{p,s} \cdot d_{p,s} - (u_{p,s} + s_{p,s}) \cdot q_{p,s}^u - o_{p,s} \cdot q_{p,s}^o \right) \quad \forall s \in S$$

Having these variables, an expected profit can simply be expressed as  $\sum_{s \in S} P(s) \cdot w_s$ . Imposing a lower bound on the worst case situation can be done by setting the lower bound for all  $w_s$  variables. If the worst case is to be maximized, a new variable  $w$  may be introduced with  $w \leq w_s$  for all  $s \in S$ , and then this variable is maximized.

Note, that this approach can significantly increase the size of the model, and thus, its computational need. Let us say that  $v^B$  variables represent or are infected by decisions from  $\mathcal{D}^B$  and  $\bar{v}^B$  are not. Similarly, let  $c^B$  denote the number of constraints that have any of those  $v^B$  variables, and  $\bar{c}^B$  those that rely only on the non-affected variables.

In the market-sensitive case from the previous section, the model would have  $v^B + \bar{v}^B$  variables and  $c^B + \bar{c}^B$  constraints. In the scenario-based market-reactive case, the model has  $|S| \cdot v^B + \bar{v}^B$  variables and  $|S| \cdot c^B + \bar{c}^B$  constraints. Clearly, by increasing the number of scenarios the complexity of the model increases, as well. This presents a practical trade-off between having a solvable model and having a more detailed scenario distribution that models the uncertain variables better.

Naturally, the number and type of those  $v^B$  variables and  $c^B$  constraints are significant. In this illustrative example, only a small portion of the variables is affected, and all of them are continuous. Similarly, only a few equality constraints are affected by those variables. Most of the variables and constraints deal with scheduling decisions, such as timing, allocation, precedence, etc, that are unaffected by  $\mathcal{D}^M$ . Thus,

**Table 4** Number of variables in the  $n$ -stage model with  $v_k = 1$  and uniform  $s$  values

s	n							
	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	1	3	7	15	31	63	127	255
3	1	4	13	40	121	364	1 093	3 280
4	1	5	21	85	341	1 365	5 461	21 845
5	1	6	31	156	781	3 906	19 531	97 656
6	1	7	43	259	1 555	9 331	55 987	335 923
7	1	8	57	400	2 801	19 608	137 257	960 800
8	1	9	73	585	4 681	37 449	299 593	2 396 745
9	1	10	91	820	7 381	66 430	597 871	5 380 840
10	1	11	111	1 111	11 111	111 111	1 111 111	11 111 111

this is a much more fortunate situation than  $\mathcal{I}^P \mathcal{D}^B \mathcal{I}^M \mathcal{D}^S$  would have been. The set of variables affected by scheduling decisions is a much larger subset of all of the variables, i.e.,  $v^S$  is much larger than  $v^B$ , and the same could be said about  $c^S$  in relation to  $c^B$ .<sup>3</sup> Moreover, plenty of these variables are binary, thus not only  $|S| \cdot v^S + \bar{v}^S$  is significantly larger than  $|S| \cdot v^B + \bar{v}^B$ , it is true for the affected number of discrete variables, as well. Obviously, solving the model in that case is a much more CPU extensive job than in this one. In addition, sequencing constraints that utilize big-M relaxation are also affected by  $\mathcal{D}^S$  in the  $\mathcal{I}^P \mathcal{D}^B \mathcal{I}^M \mathcal{D}^S$  case. Thus, the two-stage model with  $|S| \cdot c^S + \bar{c}^S$  constraints would not only have more constraints, but probably provide worse bounds due to the increased number of big-M constraints.

Note, that the idea above can be generalized to three-stage, four-stage, ...,  $n$ -stage cases, as well, that do not appear in our illustrative example. Let us consider a  $\mathcal{I}^0 \mathcal{D}^1 \mathcal{I}^1 \dots \mathcal{D}^n \mathcal{I}^n$  case, where  $v_k$  variables are affected by  $\mathcal{D}^k$  but unaffected by  $\mathcal{D}^{k+1}, \dots, \mathcal{D}^n$ , and the outcome for each  $\mathcal{I}^l$  is approximated by  $s_l$  scenarios. For the sake of simplicity, it is assumed, that these scenarios are independent from each other. In this case, there would be only one version of the  $v_1$  variables only affected by  $\mathcal{D}^1$ ,  $s_1$  different version of all of the variables affected only by  $\mathcal{D}^1$  and  $\mathcal{D}^2$ , as in the above case. For the variables that are affected by  $\mathcal{D}^1, \mathcal{D}^2$  and  $\mathcal{D}^3$ ,  $s_1 \cdot s_2$  different versions must be added, and so on. Finally, the  $v_n$  variables that are affected by  $\mathcal{D}^n$  have a total number of  $s_1 \cdot s_2 \cdot \dots \cdot s_{n-1}$ . Altogether, the total number of variables is:

$$v_1 + s_1 \cdot (v_2 + s_2 \cdot (v_3 + s_3 \cdot (\dots v_n) \dots)) = \sum_{k=1}^n \left( v_k \cdot \prod_{l=1}^{k-1} s_l \right)$$

As an illustration, Table 4 shows the number of variables for several  $n$  values while considering that  $v_k = 1$  and the number of scenarios for each stage is also uniform, e.g.,  $s_k$  is independent of  $k$  and simply denoted by  $s$ . In this simplified case the above

<sup>3</sup> Note, that  $v^S$  is not necessarily  $\bar{v}^B$ , as some redundant calculated variables may be affected by both set of decision variables.

formula reduces to  $n \cdot \sum_{k=0}^{n-1} s^k$ . Also, if the case were deterministic, the original number of variables would be  $n$  in each case, which is the same as if  $s = 1$ .

Re-running these models can be exhaustive, however, in this particular example, if the number of feasible solutions for the part of the model with  $\bar{v}^B$  variables and  $\bar{c}^B$  constraints are manageable, a two-stage solution approach can be implemented. First, the feasible solutions to this part of the model are generated and saved. This is a time consuming process, but it has to be done only when  $\mathcal{I}^P$  changes, which is rare. Next, each time the predictions on  $\mathcal{I}^M$  change, the only remaining part of the model needs to run for all of the saved schedules. Obviously, this is only reasonable, if

- The deterministic part is heavy, with many discrete variables and big-M constraints
- This part of the model has a relatively small number of feasible solutions
- The frequently changing part of the model is small and quick to solve

The advantage of this approach besides the reduced CPU times is that the models for the two parts are handled separately. It makes a possibility to apply different solution approaches. For example, the first part can be done by an MILP or SAT approach, while the profit optimization part can rely on something else. This allows the model to address more complex profit functions or continuous probability distributions for  $\mathcal{I}^M$ . For example, if only the demand changes, the news-vendor model (Edgeworth 1888) can be applied to address a continuous distribution for  $d_p$  (Hegyhati et al. 2010). An additional benefit is that the saved schedules can be evaluated independently which makes it easy to deploy the calculations on a cluster, and carry out the computations in parallel. By doing this, more complex profit functions or a higher number of considered scenarios becomes possible to consider.

#### 4.4 $\mathcal{I}^P \mathcal{D}^B \mathcal{I}^M \mathcal{D}^S$ market-reactive scheduling

At first glance, this case could seem unrealistic, however, it actually represents an integrated planning and scheduling problem if the equipment units must always be operated in 100% capacity, and their sizes have to be decided in the facility-design phase. In this case, the sizing of the equipment units means the sizing of the batches as well, which is done once before building the factory, and thus before the daily market data is revealed. Thus, batch sizing decisions can not reflect on the daily demand, however, scheduling decisions can.

Similarly to the market-sensitive batch-sizing case, a two-stage programming model can easily be derived if the distribution of market data can be described in the form of discrete scenarios. However, as mentioned previously, the number of variables and constraints affected by  $\mathcal{D}^S$  is much larger than that of  $\mathcal{D}^B$ . As a result, this model would become highly inefficient even with a small number of scenarios.

A more reasonable approach could be to generate all of the feasible schedules before making the decisions on batch sizing. Although this process would definitely take a considerable amount of time, most probably, that is not a significant issue in the planning phase of a facility. Generating all of the feasible schedules via MILP

models can be cumbersome, but other approaches, such as an S-graph algorithm or potential constraint programming model can do this in a straight-forward way.

The end result of this procedure is a list of batch number configurations ( $P$ -tuple of integers), that are feasible within the time horizon. Let  $C$  denote this set, and  $n_{p,c}$  refer to the number of batches produced from  $p \in P$  in the configuration  $c \in C$ .  $\mathcal{D}^S$  is reduced to a single selection from  $C$ , which would mean  $|C|$  binary variables, whose sum is 1.

In the two-stage model, that would result in  $|C| \cdot |S|$  binary variables, which may be a huge number, however, the overall model is very simple:

$$\begin{aligned} \sum_{c \in C} x_{c,s} &= 1 & \forall s \in S \\ d_{p,s} + q_{p,s}^o - q_{p,s}^u &= \sum_{c \in C} n_{p,c} \cdot b_p \cdot x_{c,s} & \forall p \in P, s \in S \end{aligned}$$

Where  $b_p$  is now a first stage continuous variable, and binary variable  $x_{c,s}$  denotes whether batch configuration  $c$  would be selected in scenario  $s$ . As a result, the latter equality contains a bilinear term, that could be linearized. This preventive model can provide the solution with the highest expected profit.

There are two more practical considerations to discuss for this case. When the planning phase is over in practice, and thus the  $b_p$  values are fixed, the problem reduces to a deterministic profit maximization scheduling problem, as all the remaining decisions can be made after all the input data is revealed. Thus, if the market data change for the same factory, the  $b_p$  values can be considered as input parameters at that point. However, if the  $C$  set is still kept, finding the optimal profit reduces to finding the minimal element between the values of the  $|C|$  profit options. The calculation of these profit options is a very simple formula, which can be done in parallel if necessary.

Another practically interesting situation is, when a similar plant (with the same set of units and products) is to be designed with different market estimates (scenarios and related parameters). If that were to happen, the  $|C|$  configurations would still be valid, and could be reused with the model above.

If other meta-information is also saved during the generation of the configurations, such as the minimal makespan for each configuration, the generation of the  $C$  set can be avoided and replaced by simple filtering if the new plant differs by having a shorter time horizon or having a subset of products. Even if the time horizon gets a bit longer, most of the computation can be skipped using the old  $C$  set as a cache.

## 5 Concluding remarks

Uncertainties are almost guaranteed for any application in the field of optimization. The contributions of this paper to this topic are twofold. In the first part, a generic characterization approach is presented for stochastic optimization problems. This method identifies a problem class by an alternating sequence of sets of information reveals and decisions to be made. The enumeration of such problem classes was illustrated by a stochastic scheduling example, where both the problem data, and

the set of decisions are split into two parts. Two relations among those classes were identified: an equivalence relation for sequences that define the same problem, and a partial ordering where a "stronger" problem class is unequivocally more beneficial for the decision maker, with the deterministic class being the maximal element.

For the illustrative example, 14 such distinctive classes were derived from the 24 possible sequences, and referred to as cases. The second part of the paper focused on the possible modeling and optimization techniques for 4 selected cases, where some or all of the decisions have to be made before market data is revealed. These options were discussed in detail, and investigated from the aspect of complexity, generalizability and reusability. While some of the statements hold for other optimization problems, regardless of the nature of their information and decision sets, others exploit problem specific features to propose more efficient approaches. This part of the discussion focuses not only on theoretical complexity, but possible implementation and practical application of the mentioned approaches, such as building a cache of feasible schedules a-priori. This second part leaves room for follow-up papers to conduct empirical tests on case studies, and compare the results based on quality, efficiency, and applicability.

**Acknowledgements** This work has been implemented by the TKP2021-NVA-10 project with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the 2021 Thematic Excellence Programme funding scheme.

**Funding** Open access funding provided by University of Pannonia.

## Declarations

**Conflict of interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. The authors declare the following financial interests/personal relationships which may be considered as potential competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Bakon K, Holczinger T, Süle Z, Jaskó S, Abonyi J (2022) Scheduling under uncertainty for industry 4.0 and 5.0. *IEEE Access* 10, 74977–75017. <https://doi.org/10.1109/ACCESS.2022.3191426>
- Balasubramanian J, Grossmann IE (2003) Scheduling optimization under uncertainty-an alternative approach. *Comput Chem Eng* 27(4):469–490. [https://doi.org/10.1016/S0098-1354\(02\)00221-1](https://doi.org/10.1016/S0098-1354(02)00221-1)
- Bankes S (1993) Exploratory modeling for policy analysis. *Oper Res* 41(3):435–449. <https://doi.org/10.1287/opre.41.3.435>

- Bicheno J, Holweg M, Niessmann J (2001) Constraint batch sizing in a lean environment. *Int J Prod Econ* 73(1):41–49. [https://doi.org/10.1016/S0925-5273\(01\)00091-3](https://doi.org/10.1016/S0925-5273(01)00091-3)
- Bonami P, Lodi A, Zarpellon G (2018) Learning a classification of mixed-integer quadratic programming problems. In: *International conference on the integration of constraint programming, artificial intelligence, and operations research*, pp. 595–604. [https://doi.org/10.1007/978-3-319-93031-2\\_43](https://doi.org/10.1007/978-3-319-93031-2_43). Springer
- Capón-García E, Bojarski AD, Espuña A, Puigjaner L (2011) Multiobjective optimization of multiproduct batch plants scheduling under environmental and economic concerns. *AIChE J* 57(10):2766–2782. <https://doi.org/10.1002/aic.12477>
- Chu Y, You F, Wassick JM, Agarwal A (2015) Integrated planning and scheduling under production uncertainties: Bi-level model formulation and hybrid solution method. *Comput Chem Eng*, 72, 255–272. A Tribute to Ignacio E. Grossmann. <https://doi.org/10.1016/j.compchemeng.2014.02.023>
- Dunn WN (2018) Using the method of context validation to mitigate Type III errors in environmental policy analysis. In: *Knowledge, power, and participation in environmental policy analysis*, pp. 417–436. <https://doi.org/10.4324/9781351325721-25>
- Edgeworth FY (1888) The mathematical theory of banking. *J Roy Stat Soc* 51(1):113–127
- Freeman NK, Melouk SH, Mittenthal J (2016) A scenario-based approach for operating theater scheduling under uncertainty. *Manuf Serv Oper Manag* 18(2):245–261. <https://doi.org/10.1287/msom.2015.0557>
- Glock CH (2010) Batch sizing with controllable production rates. *Int J Prod Res* 48(20):5925–5942. <https://doi.org/10.1080/00207540903170906>
- Glock CH (2011) Batch sizing with controllable production rates in a multi-stage production system. *Int J Prod Res* 49(20):6017–6039. <https://doi.org/10.1080/00207543.2010.528058>
- Gupta D, Maravelias CT, Wassick JM (2016) From rescheduling to online scheduling. *Chem Eng Res Design* 116, 83–97. <https://doi.org/10.1016/j.cherd.2016.10.035>. *Process Systems Engineering - A Celebration in Professor Roger Sargent's 90th Year*
- Harremoës P, Madsen H (1999) Fiction and reality in the modelling world-balance between simplicity and complexity, calibration and identifiability, verification and falsification. *Water Sci Technol* 39(9):1–8. [https://doi.org/10.1016/S0273-1223\(99\)00210-3](https://doi.org/10.1016/S0273-1223(99)00210-3)
- Hazır Ö, Kedad-Sidhoum S (2014) Batch sizing and just-in-time scheduling with common due date. *Ann Oper Res* 213(1):187–202. <https://doi.org/10.1007/s10479-012-1289-9>
- Hegyháti M, Friedler F (2010) Overview of industrial batch process scheduling. *Chem Eng Trans* 21:895–900. <https://doi.org/10.3303/CET1021150>
- Hegyháti M, Lainez JM, Puigjaner L, Friedler F (2010) Preemptive Optimization of Chemical Batch Processes with Continuous External Uncertainties Based Ona Novel S-graph Branching Strategy. In: *13th Conference process integration, modelling and optimisation for energy saving and pollution reduction*, Prague, Czech Republic
- Herroelen W, Leus R (2005) Project scheduling under uncertainty: survey and research potentials. *Eur J Oper Res* 165(2):289–306. <https://doi.org/10.1016/j.ejor.2004.04.002>
- Hu Z, Hu G (2016) A two-stage stochastic programming model for lot-sizing and scheduling under uncertainty. *Int J Prod Econ* 180:198–207. <https://doi.org/10.1016/j.ijpe.2016.07.027>
- Hu Z, Hu G (2018) A multi-stage stochastic programming for lot-sizing and scheduling under demand uncertainty. *Comput Ind Eng* 119:157–166. <https://doi.org/10.1016/j.cie.2018.03.033>
- Jamal A, Sarker BR, Mondal S (2004) Optimal manufacturing batch size with rework process at a single-stage production system. *Comput Ind Eng* 47(1):77–89. <https://doi.org/10.1016/j.cie.2004.03.001>
- Kim T, Kim Y-W, Cho H (2020) Dynamic production scheduling model under due date uncertainty in precast concrete construction. *J Clean Prod* 257:120527. <https://doi.org/10.1016/j.jclepro.2020.120527>
- Liu L, Gu H-y, Xi Y-g (2007) Robust and stable scheduling of a single machine with random machine breakdowns. *Int J Adv Manuf Technol* 31(7):645–654. <https://doi.org/10.1007/s00170-005-0237-0>
- Lu Z, Cui W, Han X (2015) Integrated production and preventive maintenance scheduling for a single machine with failure uncertainty. *Comput Ind Eng* 80:236–244. <https://doi.org/10.1016/j.cie.2014.12.017>
- Ma S, Li H, Zhu N, Fu C (2021) Stochastic programming approach for unidirectional quay crane scheduling problem with uncertainty. *J Sched* 24(2):137–174. <https://doi.org/10.1007/s10951-020-00661-8>
- McKenzie P, Jayanthi S (2007) Ball aerospace explores operational and financial trade-offs in batch sizing in implementing JIT. *Interfaces* 37(2):108–119. <https://doi.org/10.1287/inte.1060.0281>



- Mencarelli L, D'Ambrosio C (2019) Complex portfolio selection via convex mixed-integer quadratic programming: a survey. *Int Trans Oper Res* 26(2):389–414. <https://doi.org/10.1111/itor.12541>
- Méndez CA, Cerdá J, Grossmann IE, Harjunkoski I, Fahl M (2006) State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput Chem Eng* 30(6–7):913–946. <https://doi.org/10.1016/j.compchemeng.2006.02.008>
- Mihoubi B, Bouzouia B, Gaham M (2021) Reactive scheduling approach for solving a realistic flexible job shop scheduling problem. *Int J Prod Res* 59(19):5790–5808. <https://doi.org/10.1080/00207543.2020.1790686>
- Ning C, You F (2017) A data-driven multistage adaptive robust optimization framework for planning and scheduling under uncertainty. *AIChE J* 63(10):4343–4369. <https://doi.org/10.1002/aic.15792>
- O'Donovan R, Uzsoy R, McKay KN (1999) Predictable scheduling of a single machine with breakdowns and sensitive jobs. *Int J Prod Res* 37(18):4217–4233. <https://doi.org/10.1080/002075499189745>
- Petkov SB, Maranas CD (1997) Multiperiod planning and scheduling of multiproduct batch plants under demand uncertainty. *Ind Eng Chem Res* 36(11):4864–4881. <https://doi.org/10.1021/ie970259z>
- Prade H (1979) Using fuzzy set theory in a scheduling problem: a case study. *Fuzzy Sets Syst* 2(2):153–165. [https://doi.org/10.1016/0165-0114\(79\)90022-8](https://doi.org/10.1016/0165-0114(79)90022-8)
- Qin W, Zhang J, Song D (2018) An improved ant colony algorithm for dynamic hybrid flow shop scheduling with uncertain processing time. *J Intell Manuf* 29(4):891–904. <https://doi.org/10.1007/s10845-015-1144-3>
- Sanmartí E, Puigjaner L, Holczinger T, Friedler F (2002) Combinatorial framework for effective scheduling of multipurpose batch plants. *AIChE J* 48(11):2557–2570. <https://doi.org/10.1002/aic.690481115>
- Sarker BR, Jamal AMM, Mondal S (2008) Optimal batch sizing in a multi-stage production system with rework consideration. *Eur J Oper Res* 184(3):915–929. <https://doi.org/10.1016/j.ejor.2006.12.005>
- Shang C, You F (2018) Distributionally robust optimization for planning and scheduling under uncertainty. *Comput Chem Eng* 110:53–68. <https://doi.org/10.1016/j.compchemeng.2017.12.002>
- Shi H, You F (2016) A computational framework and solution algorithms for two-stage adaptive robust scheduling of batch manufacturing processes under uncertainty. *AIChE J* 62(3):687–703. <https://doi.org/10.1002/aic.15067>
- Sluijs J (1997) Anchoring amid uncertainty: on the management of uncertainties in risk assessment of anthropogenic climate change= houvast zoeken in onzekerheid: over het omgaan met onzekerheden in risicoanalyse van klimaatverandering door menselijk handelen. PhD thesis, Wageningen University & Research, The Netherlands
- Suresh V, Chaudhuri D (1993) Dynamic scheduling—a survey of research. *Int J Prod Econ* 32(1):53–63. [https://doi.org/10.1016/0925-5273\(93\)90007-8](https://doi.org/10.1016/0925-5273(93)90007-8)
- Van Asselt MB, Rotmans J (2002) Uncertainty in integrated assessment modelling. *Clim Change* 54(1):75–105. <https://doi.org/10.1023/A:1015783803445>
- Verderame PM, Elia JA, Li J, Floudas CA (2010) Planning and scheduling under uncertainty: a review across multiple sectors. *Ind Eng Chem Res* 49(9):3993–4017. <https://doi.org/10.1021/ie902009k>
- Vin JP, Ierapetritou MG (2001) Robust short-term scheduling of multiproduct batch plants under demand uncertainty. *Ind Eng Chem Res* 40(21):4543–4554. <https://doi.org/10.1021/ie0007724>
- Walker WE, Harremoës P, Rotmans J, Van Der Sluijs JP, Van Asselt MB, Janssen P, Kraayer von Krauss MP (2003) Defining uncertainty: a conceptual basis for uncertainty management in model-based decision support. *Integr Assess* 4(1):5–17. <https://doi.org/10.1076/iaij.4.1.5.16466>
- Wang Z, Hu H (2018) Dynamic response to demand variability for precast production rescheduling with multiple lines. *Int J Prod Res* 56(16):5386–5401. <https://doi.org/10.1080/00207543.2017.1414970>
- Wu W, Hayashi T, Haruyasu K, Tang L (2023) Exact algorithms based on a constrained shortest path model for robust serial-batch and parallel-batch scheduling problems. *Eur J Oper Res* 307(1):82–102. <https://doi.org/10.1016/j.ejor.2022.09.032>
- Wullink G, Gademann AJRM, Hans EW, van Harten A (2004) Scenario-based approach for flexible resource loading under uncertainty. *Int J Prod Res* 42(24):5079–5098. <https://doi.org/10.1080/002075410001733887>