Discrete Applied Mathematics 379 (2026) 355–378

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

An approximate dynamic programming approach for multi-stage stochastic lot-sizing under a Decision–Hazard–Decision information structure

Victor Spitzer [a,b,*], Céline Gicquel [b], Evgeny Gurevsky [c], François Sanson [a]

[a] Lhyfe, Nantes, France
[b] LISN, Université Paris-Saclay, Gif-sur-Yvette, France
[c] LS2N, Université de Nantes, France

ARTICLE INFO

Article history:
Received 14 January 2025
Received in revised form 7 July 2025
Accepted 20 August 2025

Keywords:
Lot-sizing
Multi-stage stochastic integer programming
Decision–Hazard–Decision information structure
Approximate stochastic dynamic programming
Two-stage stochastic integer programming
Benders decomposition

ABSTRACT

This work studies a combinatorial optimization problem encountered in industrial production planning: the single-item multi-resource lot-sizing problem with inventory bounds and lost sales. The demand to be satisfied by the production plan is subject to uncertainty and only probabilistically known. We consider a multi-stage decision process with a Decision–Hazard–Decision information structure in which decisions are made at each stage both before and after the uncertainty is revealed. Such a setting has not yet been studied for stochastic lot-sizing problems, and the resulting problem is modeled as a multi-stage stochastic integer program. We propose a solution approach based on an approximate stochastic dynamic programming algorithm. It relies on a decomposition of the problem into single-stage sub-problems and on the estimation at each stage of the expected future costs. Due to the Decision–Hazard–Decision information structure, each nested single-stage sub-problem is itself a two-stage stochastic integer program. We therefore introduce a Benders decomposition scheme to reduce the computational effort required to solve each nested sub-problem, and present a special-purpose polynomial-time algorithm to efficiently solve the single-scenario second-stage sub-problems involved in the Benders decomposition. The results of extensive simulation experiments carried out on large-size randomly generated instances are reported. They demonstrate the practical benefit, in terms of the actual production cost, of using the proposed approach as compared to a naive deterministic optimization approach based on the expected demand.

1. Introduction

Production planning and lot-sizing are among the key processes in the field of industrial operations management. Lot-sizing essentially consists in determining the quantity and timing of production of a set of items on a set of production resources, over a finite planning horizon comprising several time-steps. Lot-sizing aims at optimizing the interactions between the customer demand for the produced items, the capacity of the production resources generating these items, and the use of an inventory to store items produced in advance. The objective is to build a minimum-cost production

* Corresponding author.
E-mail address: victor.spitzer@universite-paris-saclay.fr (V. Spitzer).

https://doi.org/10.1016/j.dam.2025.08.051

plan satisfying as best as possible the customers' demand and complying with the practical limitations of the production system. The resulting problem can be represented as a mixed-integer linear program.

This work addresses the single-item multi-resource capacitated lot-sizing problem with inventory bounds and lost sales. In this variant of the lot-sizing problem, a set of parallel capacitated resources produce a single type of item, and the inventory capacity is bounded. Moreover, any customer demand which cannot be met on time is lost, which incurs a linear penalty.

The input data of a lot-sizing problem are often subject to uncertainty. In particular, the customers' demand to be satisfied may be difficult to accurately predict and may thus be only stochastically known at the time when the production plan has to be determined. In this work, we investigate a lot-sizing problem with uncertain demand. We assume that the information about the actual demand is revealed progressively over time and that production and inventory decisions may be regularly updated to adjust the plan to the realized demand. This leads to the use of a multi-stage stochastic integer programming approach.

When modeling a multi-stage stochastic optimization problem, it is important to define the information structure, *i.e.*, to describe how uncertainty is revealed over time and when decisions are made to adapt to this evolving knowledge. Most prior studies on multi-stage stochastic lot-sizing considered that the production and inventory decisions related to a given decision stage are determined at each stage once the demand for this stage is revealed (see, *e.g.*, Brandimarte [17], Hu and Hu [28], and Quezada et al. [40]). Such a setting corresponds to a Hazard–Decision information structure. However, in practice, it may happen that all or part of the production decisions related to a given stage have to be made right before the demand for this stage realizes. In this work, we wish to investigate such a setting. More precisely, we consider that the setup decisions related to a given decision stage have to be made just before the demand for this stage is disclosed while the other decisions related to production quantities, inventory holding and lost sales, can be made afterwards. This corresponds to a Decision–Hazard–Decision information structure.

The resulting multi-stage stochastic integer program is particularly challenging to solve. Namely, a first solution approach might consist in reformulating the Decision–Hazard–Decision problem as a Hazard–Decision problem with an augmented state space. As will be discussed in more detail later on in the paper, this Hazard–Decision problem is however numerically intractable in our case. This is why, in this work, we chose not to use such a reformulation. We instead introduce a novel solution methodology based on approximate stochastic dynamic programming to directly solve the Decision–Hazard–Decision problem. The proposed algorithm requires to solve multiple two-stage stochastic lot-sizing problems at each step of the dynamic programming recursion. We thus investigate a Benders decomposition scheme to efficiently solve these two-stage stochastic integer programs.

In the remainder of this introductory section, we first describe the use-cases which provided the initial incentive for this study. We then provide a brief overview of the articles most closely related to our work. We finally conclude this section with a discussion on our contributions.

1.1. Motivations

This work is originally motivated by use-cases in the green hydrogen production industry (see, *e.g.*, Riera et al. [45]). Green hydrogen is produced through a water electrolysis process consuming electricity and aiming at splitting water molecules into hydrogen and oxygen. The produced hydrogen is termed "green" if the consumed electricity originates from renewable energy sources. Green hydrogen production sites thus comprise one or several parallel heterogeneous machines (electrolyzers) producing a single type of item (pressurized hydrogen). Technical constraints impose that an electrolyzer is either inactive or producing above a given minimum threshold (see, *e.g.*, Baumhof et al. [8]). Green hydrogen production planning thus involve setup decisions to determine whether to keep each electrolyzer inactive or to switch it on. The hydrogen, once produced, is stored in a tank with a limited storage capacity.

Furthermore, there exist hybrid production sites that use electricity coming both from the regional power grid and from local renewable assets directly connected to the site. These production sites seek to ensure the on-time fulfillment of the customers' demand for hydrogen, while using as much as possible the locally generated renewable electricity. When it is not available due to the intermittency of renewable energy sources, the production sites may purchase electricity from the regional grid. In this use-case, the average amount of renewable power available at each time-step of the production plan is uncertain. Furthermore, a production plan must be designed each day to decide on the setups of all production units, as well as on the energy consumption on the grid, for each time-step of the following day. This anticipative decision on setups and grid consumption is made without the knowledge of the exact future availability of the renewable asset, while the remaining production decisions may be adjusted in real time to fit the actual renewable availability. We are thus faced with a multi-stage stochastic decision process involving a Decision–Hazard–Decision feature.

The purchasing price of grid electricity is considered to be much higher than the one of the locally generated green electricity, most often both for economical and ecological reasons. Consequently, resorting to the grid electricity may be associated to a significant penalty cost. Similarly, the inability to fully supply customers on time is penalized. Hence, in situations where the renewable availability is not accurately predicted, the uncertainty may lead to significant overcosts. Indeed, it may lead either to a costly over-reliance on grid electricity or to the inability to fully supply customers.

This situation is quite similar to a feature frequently encountered in the lot-sizing literature, namely, the use of lost sales by which any demand for the end-items which cannot be met on time is heavily penalized in the objective function.

We thus focus in this work on a generic single-item multi-resource lot-sizing problem with inventory bounds and lost sales in a Decision–Hazard–Decision stochastic setting.

The present work can be seen as a generalization and extension of a previous work carried out by Spitzer et al. [50]. In that previous study, the authors address a specific use-case related to the optimization of a hybrid hydrogen production plant with uncertain availability of one of its energy sources and consider a stochastic decision process involving only two stages. We extend their results to a more generic stochastic lot-sizing problem and to a multi-stage stochastic decision process. Moreover, the authors introduce a problem-specific polynomial-time algorithm to efficiently solve the single-scenario second-stage sub-problems involved in a Benders decomposition. We generalize the use of this algorithm to the single-scenario second-stage sub-problems arising in the generic stochastic lot-sizing problem at hand. Finally, the numerical experiments led in the previous study are limited to simulations for a particular use-case while we demonstrate the benefit of our extended approach for a large variety of randomly generated instances.

1.2. Literature review

We now briefly discuss the research works most closely related to our study.

Deterministic single-item lot-sizing. Single-item lot-sizing problems address various challenges in production planning, and are most often modeled as Mixed-Integer Linear Programs (MILP). The reader is referred, for example, to the literature review provided by Brahimi et al. [16] for a recent overview on the topic. We focus here on the works involving features similar to ours. Lot-sizing for a production system involving a single item and multiple parallel resources is investigated among others by Vanderbeck and Wolsey [53], Rapine et al. [44], and Koca and Koksalan [30]. Capacitated lot-sizing problem with inventory bounds is studied, for instance, by Erenguc and Aksoy [24] and Akbalik et al. [3]. Lost sales are sometimes introduced in lot-sizing models to account for the difficulty in meeting the demand on time due to capacity restrictions (see, *e.g.*, Aksen et al. [4]). Finally, the present work is also closely related to the single-item single-resource capacitated lot-sizing problems with inventory bounds and lost sales considered by Sandbothe and Thompson [46], Berk et al. [10], and Liu and Tu [32].

Robust lot-sizing. A first way to account for uncertainty in lot-sizing consists in relying on robust optimization (see, *e.g.*, Bertsimas et al. [11]). Such an approach may be well suited when data availability is too limited to obtain an accurate probabilistic representation of the uncertainty, or when computational efficiency is more important than achieving the most optimal solution (see, *e.g.*, Alem et al. [5]). It may also be relevant when focusing on the worst possible realization of the uncertainty, for example in our case when lost sales should be avoided at any cost due to extremely high penalty values. Single-stage robust lot-sizing models are investigated among others by Bertsimas and Thiele [12], Alem et al. [5] and Schlenkrich et al. [47]. These models assume that all decisions must be made before uncertainty realization. Models based on adaptive robust optimization (see, *e.g.*, Ben-Tal et al. [9]) are proposed among others by Postek and den Hertog [38] and Curcio et al. [21]. These models account for adjustable decisions — that is, decisions made only after the uncertainty has been revealed. See also Bindenwald et al. [13], for an extensive review of the various approaches to multistage lot sizing with uncertain demand, including robust optimization methods.

Multi-stage stochastic programming approaches for production planning problems. Uncertainty in a lot-sizing problem may also be handled through a stochastic programming approach. As explained, for example, by Alem et al. [5], stochastic programming is appropriate when sufficient data is available to generate scenarios, and when the quality of the solution is prioritized over the computational effort. The stochastic lot-sizing problem is a long-standing research topic in the literature, with a particular focus on the demand-side uncertainty in a multi-stage setting (see, *e.g.*, Brandimarte [17] and Tempelmeier [52]). A review of exact and heuristic methods to solve this problem is presented in what follows.

A first category of exact solution approaches relies on scenario trees to represent the time evolution of the uncertain demand, and models the problem as a large-size MILP. This MILP may be solved to exact optimality by a branch-and-cut algorithm. However, for this algorithm to be numerically efficient, it is crucial to strengthen the bounds provided by the linear relaxation of the problem. Such work exists among others for the single-item capacitated lot-sizing problem (see, *e.g.*, Di Summa and Wolsey [23]), and for the multi-item multi-level uncapacitated lot-sizing problem with lost sales (see, *e.g.*, Quezada et al. [42]). However, even if it is combined with the use of formulation strengthening techniques, this type of solution approaches is numerically efficient only on small scenario trees, and requires prohibitive computation time for large instances of the problem. This difficulty can be partly overcome through the use of a scenario-tree reduction technique which reduces the size of the MILP model while preserving the most valuable statistical information (see, *e.g.*, Hu and Hu [28]).

Stochastic dynamic programming approaches can also be used to optimally solve lot-sizing problems under demand uncertainty (see, *e.g.*, Dellaert and Melo [22], Li et al. [31], and Naeem et al. [36]). This type of method carries out a stage-wise decomposition of the problem and seeks at each stage to determine the best action to be made for each possible state of the system in order to minimize a "cost-to-go" function representing the expected future production costs. However, it requires to explicitly enumerate all possible states of the system and all the actions that might be taken at each state, and to evaluate each state–action combination to select the most cost-efficient one. This strongly limits the application range of this method, especially in terms of the number of state–action combinations that can be considered at each stage.

There exist approximate stochastic dynamic approaches to overcome these limitations (see, *e.g.*, Powell [39]). They rely on the approximation of the state–action space or of the cost-to-go function to obtain a tractable reformulation of the stochastic problem. Applications for stochastic energy production planning problems are reported in the literature: see, for example, Zhang and Nikovski [57] and Xiao [55] for a unit commitment problem, or Cervellera et al. [20] and Zéphyr et al. [56] for a hydropower production planning problem. Such an approach is used in this work to solve the stochastic lot-sizing problem by successive constructions of an approximate representation of the cost-to-go function at each stage. To the best of our knowledge, approximate stochastic dynamic approaches have not yet been studied for multi-stage stochastic lot-sizing problems.

The Stochastic Dual Dynamic integer Programming (SDDiP) algorithm, first introduced by Zou et al. [58] for general multi-stage stochastic integer programs, also relies on a stage-wise decomposition of the problem. It basically consists in solving a series of single-stage sub-problems taking the form of deterministic MILPs. These sub-problems are linked together by expected cost-to-go functions. The main idea of the SDDiP algorithm is to provide near-optimal solutions of the initial multi-stage stochastic integer program by iteratively building a piece-wise linear under-approximation of each expected cost-to-go function. Solution approaches based on the SDDiP algorithm are proposed by Quezada et al. [40] for the single-item uncapacitated lot-sizing problem and by Quezada et al. [41] for a multi-item uncapacitated lot-sizing problem with lost-sales. The SDDiP algorithm is an exact solution method whose finite and optimal convergence is theoretically guaranteed. However, in practice, it suffers from shortcomings in computational efficiency, mainly because its iterative scheme requires numerous cost estimations.

As for heuristic solution approaches, many of them are also based on a stage-wise decomposition of the problem. Curcio et al. [21] propose a heuristic to solve their multi-stage capacitated lot-sizing problem in a rolling horizon manner, by incrementally fixing decisions at each stage while relaxing integrality constraints at future stages. A similar rolling horizon approach is implemented by Balasubramanian and Grossmann [7]. In their solution method, decisions are incrementally fixed by solving a sequence of two-stage stochastic programs while rolling forward the time window by one stage at each iteration. The two-stage stochastic program solved at a given iteration considers uncertainties and decisions for the immediately following stage but not for the stages further beyond. The solution approach proposed in this work also relies on solving a series of two-stage stochastic problems. However, it allows each of these two-stage stochastic problems to explicitly consider uncertainties and decisions at all future stages through the use of cost-to-go functions, thus providing solutions of improved quality.

Decision–Hazard–Decision information structure in multi-stage stochastic programming. Most multi-stage stochastic programming models implicitly assume that, at each stage of the decision process, the decisions can be made after the realization of all the random parameters related to this stage. Carpentier et al. [18] introduce the term "Hazard–Decision" to designate this information structure, while the term "Decision–Hazard" refers to the case where decisions must be made before the uncertainty realization. The situation in which decisions are made both before and after the uncertainty realization was later called the "Decision–Hazard–Decision" setting (see, *e.g.*, Carpentier et al. [19] and Martinez Parra et al. [35]). In particular, Martinez Parra et al. [35] use an approximate stochastic dynamic programming approach to address a problem closely related to stochastic unit commitment in a "Decision–Hazard–Decision" setting. The authors solve two-stage stochastic problems at each iteration of the dynamic scheme in a similar fashion to what is proposed in our work.

Street et al. [51] study a multi-stage stochastic programming model using a Decision–Hazard–Decision information structure for hydrothermal scheduling under uncertain water inflows. They compare two formulations of their problem, respectively relying on a Decision–Hazard–Decision and a Hazard–Decision information structure. The latter is derived from the former by considering, at the previous stage, the decisions to be made at a given stage before uncertainty realization. This however leads to a state space of increased size, since all the information on the anticipative decisions made at the previous stage must be transferred to the next. The authors propose to solve their problem using the stochastic dual dynamic programming (SDDP) algorithm (see, *e.g.*, Shapiro [49]), an approach well known to be efficient for hydro power system scheduling under uncertainty (see, *e.g.*, Gjelsvik [25]). Their numerical results show that the Hazard–Decision reformulation leads to shorter computing times than the initial Decision–Hazard–Decision formulation. This benefit can be partly explained by the moderate increase of the state space dimension caused by the reformulation (thanks to the use of aggregate state variables), and by the fact that their problem involves only continuous variables (which allows the use of an efficient implementation of the SDDP algorithm). This contrasts with our case. Namely, in our problem, using a Hazard–Decision reformulation results in a sharp increase of the state space dimension. Furthermore, the presence of binary variables requires the use of the SDDiP algorithm, an algorithm that already has some trouble solving large instances of the classic Hazard–Decision problem involving a single state variable. These two aspects make the Hazard–Decision reformulation numerically intractable in our case.

Benders decomposition for two-stage stochastic lot-sizing. Formulating the problem explicitly using a Decision–Hazard–Decision information structure implies that each stage sub-problem is itself a two-stage stochastic lot-sizing problem. Such problems are far from prevalent in the lot-sizing literature, noticeable exceptions can be found, for example, in Macedo et al. [33] and Hu and Hu [27]. Furthermore, the use of Benders decomposition schemes (see, *e.g.*, Rahmaniani et al. [43]) for their efficient resolution is also scarcely documented. The topic, first introduced by Bahl and Zionts [6], remained unmentioned for almost three decades. Lately, Benders decomposition algorithms have been applied to solve problems

closely related to the lot-sizing problem under demand uncertainty (see, *e.g.*, Adulyasak et al. [1], Witthayapraphakorn and Charnsethikul [54], and Gruson et al. [26]). These three approaches assume that all the binary decision variables are first-stage decision variables so that the resulting second-stage sub-problems are linear programs. In particular, Gruson et al. [26] propose to exploit the fact that, in their case, these second-stage sub-problems have a special structure corresponding to the one of a minimum cost flow problem in a network (see, *e.g.*, Bowman [15] and Pochet and Wolsey [37]), and that any such problem with integer demand and capacities may in turn have its dual solution be found efficiently (see, *e.g.*, Ahuja et al. [2]). This allows a significant reduction of the overall computing time for their Benders decomposition algorithm. We implement in this work a Benders decomposition scheme specific to our case with a novel polynomial time algorithm to solve the dual sub-problem. This new algorithm has the benefit of not relying on the assumption of integer demand and capacities.

1.3. Contributions

The main novelty of this article thus lies in the following three aspects.

Firstly, we investigate a multi-stage stochastic programming model for the single-item multi-resource stochastic lot-sizing problem with inventory bounds and lost sales under uncertain demand. We consider an information structure in which, at each stage, anticipative decisions (setups) are made before the uncertainty is revealed and recourse decisions (production quantities, inventory and lost sales) are made afterwards. To the best of our knowledge, such setting in which part of the production decisions related to a stage have to be made just before uncertainty realizes at this stage has not yet been studied in the context of multi-stage stochastic lot-sizing.

Secondly, we propose to handle the problem while explicitly relying on a Decision–Hazard–Decision information structure, since reformulating the problem using a Hazard–Decision one is numerically intractable in our case. This is achieved by designing a novel approximate stochastic dynamic programming approach. In this approach, the sub-problem to be solved at each decision stage is itself a two-stage stochastic integer program. This one comprises the anticipative and recourse decisions to be made at each stage, and seeks to optimize the sum of the immediate production costs to be paid at the current stage and of the production costs to be paid in the stages onward. We represent those latter costs by an approximate expected cost-to-go function of a single variable: the entering inventory level at the beginning of the next decision stage. This approach differs from exact stochastic dynamic programming methods since the decisions to be implemented at each stage are determined by solving an optimization problem taking the form of a MILP rather than by enumerating all possible state–action combinations. This allows us to consider a much larger number of uncertainty scenarios and actions at each stage. The proposed approach also differs from the SDDiP algorithm. This algorithm is already numerically cumbersome but still feasible under a Hazard–Decision setting in which each stage sub-problem is a deterministic MILP. It may however be computationally intractable under a Decision–Hazard–Decision setting in which each stage sub-problem is a two-stage stochastic integer program. Although our proposed approach does not possess the same guarantees of optimal convergence as the SDDiP algorithm, it is able to address large and realistic instances of the problem in practice. It thus represents an original contribution toward the efficient solving of multi-stage stochastic lot-sizing problems under a Decision–Hazard–Decision setting.

Thirdly, we introduce a Benders decomposition scheme to efficiently solve the two-stage stochastic integer programs encountered at each step of the approximate stochastic dynamic programming approach, while taking into account a large number of demand scenarios. It is made possible only thanks to a convex under-approximation of the cost-to-go functions built at each stage of the dynamic programming approach. To the best of our knowledge, this work is the first to propose this technique to implement a Benders decomposition scheme within an approximate stochastic dynamic programming framework. Furthermore, we introduce a novel cubic-time algorithm to reduce the computational effort needed to solve the dual of each of these sub-problems at each iteration of the decomposition scheme. This algorithm does not rely on the assumption of integer demand and capacities, as opposed to existing network flow approaches.

The remainder of the paper is organized as follows. Section 2 formally describes the stochastic lot-sizing problem under study. Section 3 presents the approximate stochastic dynamic programming approach proposed to solve this problem. The Benders decomposition scheme implemented to efficiently solve each two-stage stochastic integer program is discussed in Section 4. Finally, the results of our numerical experiments are provided in Section 5. They demonstrate that, even if the proposed approach is approximate, it displays practical benefits and is able to provide good quality production plans in numerous cases.

2. Problem definition and modeling

This section provides a formal definition of the lot-sizing problem under study. We first present a mixed-integer linear programming formulation for the deterministic case. We then extend it to consider a stochastic demand and introduce a new multi-stage stochastic programming model to handle this case.

2.1. Deterministic model

A mathematical model is introduced for the single-item capacitated lot-sizing problem with inventory bounds and lost sales. The objective is to find a feasible production plan that minimizes the costs of production, setup, inventory holding and lost sales (see, *e.g.*, Sandbothe and Thompson [46]). Borrowing from the nomenclature of Pochet and Wolsey [37], this production planning problem is denoted as LS-C-L-SUB for the Lot-Sizing problem with Capacities, Lost-sales and Storage Upper Bounds.

We plan the operation of a set $\mathcal{N} = \{1, 2, \ldots, N\}$ of production units over a set $\mathcal{T} = \{1, 2, \ldots, T\}$ of time-steps. Parameters h_t and g_t denote respectively the inventory holding cost and lost sales cost for any time-step $t \in \mathcal{T}$, while S_t corresponds to the storage capacity in t. Parameters $p_{t,n}$ and $q_{t,n}$ denote respectively the production cost and setup cost for the production unit $n \in \mathcal{N}$ at time-step $t \in \mathcal{T}$, while $C_{t,n}$ corresponds to the production capacity of that unit at that time step. Finally, d_t denotes the demand at time-step $t \in \mathcal{T}$.

We represent, at time-step $t \in \mathcal{T}$, the inventory level by a non-negative variable s_t, the lost sales by a non-negative variable l_t, the quantity produced on production unit $n \in \mathcal{N}$ by a non-negative variable $x_{t,n}$ and the setup state of production unit $n \in \mathcal{N}$ by a binary variable $y_{t,n}$. Suppose that the initial inventory level at the beginning of time-step 0, s^e, is given, one may formulate the single-item capacitated lot-sizing problem with inventory bounds and lost sales as follows.

Problem LS-C-L-SUB :

$$\min \quad \sum_{t\in\mathcal{T}} \left(h_t \cdot s_t + g_t \cdot l_t + \sum_{n\in\mathcal{N}} \Big[q_{t,n} \cdot y_{t,n} + p_{t,n} \cdot x_{t,n} \Big] \right) \tag{1a}$$

$$\text{s.t. } s^e + \sum_{n\in\mathcal{N}} x_{1,n} = d_1 - l_1 + s_1, \tag{1b}$$

$$s_{t-1} + \sum_{n\in\mathcal{N}} x_{t,n} = d_t - l_t + s_t, \qquad \forall t \in \mathcal{T}\backslash\{1\} \tag{1c}$$

$$x_{t,n} \le C_{t,n} \cdot y_{t,n}, \qquad \forall n \in \mathcal{N}, \quad \forall t \in \mathcal{T} \tag{1d}$$

$$s_t \le S_t, \qquad \forall t \in \mathcal{T} \tag{1e}$$

$$s_t, l_t \ge 0, \qquad \forall t \in \mathcal{T} \tag{1f}$$

$$x_{t,n} \ge 0, \ y_{t,n} \in \{0, 1\}, \qquad \forall n \in \mathcal{N}, \quad \forall t \in \mathcal{T} \tag{1g}$$

Objective (1a) is to minimize the total inventory, lost-sales, production and setup costs over the planning horizon. Constraints (1b)–(1c) are inventory balance equations: they ensure that, in each time-step, the overall quantity of available product, *i.e.*, the starting inventory s_{t-1} plus the quantity produced by the production units $\sum_{n\in\mathcal{N}} x_{t,n}$, is equal to the sum of the part of the demand that will be actually satisfied $d_t - l_t$ and of the ending inventory s_t. Constraints (1d) and (1e) ensure that the maximum production and storage capacities are respected in each time-step. Furthermore, constraints (1d) ensure that the setup costs are correctly incurred in each time-step when production occurs on a production unit. Finally, (1f)–(1g) are variables domain constraints.

Note that problem LS-C-L-SUB is NP-hard even if there is a single production unit ($N = 1$), an unlimited storage capacity ($S_t = +\infty$ for all $t \in \mathcal{T}$), and no lost sales ($l_t = +\infty$ for all $t \in \mathcal{T}$). The proof is based on a reduction from the single-item single-resource capacitated lot-sizing problem with general time-varying production capacities (see, *e.g.*, Bitran and Yanasse [14]).

2.2. Stochastic model

Most often, the information on the demand is obtained through imperfect forecasting procedures and is thus subject to uncertainties. Consequently, the actual demand value is not known at the beginning of the planning horizon but is rather gradually revealed over time. The realized value of the demand may be higher or lower than the initially forecasted value, leading to costly over-production or lost sales.

In this work, we assume that new information about the actual demand becomes available at the end of K predefined time-steps denoted by $\{T^1, T^2, \ldots, T^K\}$. Furthermore, we consider the case where anticipative setup decisions must be made for all time-steps belonging to $\{T^k + 1, \ldots, T^{k+1}\}$ at the end of time-step T^k before the actual demand over $\{T^k + 1, \ldots, T^{k+1}\}$ is revealed.

This leads to a decomposition of the planning horizon $\mathcal{T}$ into a set of $\mathcal{K} = \{1, 2, \ldots, K\}$ smaller time intervals. Time interval $\mathcal{T}^k$, $k \in \mathcal{K}$, is defined as $\mathcal{T}^k = \{T^k + 1, \ldots, T^{k+1}\}$. Let $\widetilde{d}_t$ be the random variable representing the demand at time-step t. We assume that the actual value of $\widetilde{d}_t$ for each time-step $t \in \mathcal{T}^k$ is disclosed at the end of time-step T^k.

In line with the periodical decision making and revelation of the actual demand, we model this production planning problem as a multi-stage decision process involving K decision stages, such that the stage $k \in \mathcal{K}$ corresponds to time-interval $\mathcal{T}^k$. Indeed, all decisions are not made once and for all at the beginning of the planning horizon. Instead, some of

them are postponed to later points in times when more information on the demand become available. In this work, we thus assume that the production planning decisions relative to any stage may be postponed up to that stage.

At each stage, anticipative decisions are thus made before the uncertainty is revealed while other decisions may be made afterwards at the last moment. Hence, we consider an information structure in which each stage k comprises two decision sub-stages. The first sub-stage corresponds to the anticipative decisions made right at the end of time-step T^k, when the demand is known up to time-step T^k but not yet for time-steps in $\mathcal{T}^k$. These are the setup decisions relative to stage k, *i.e.*, relative to the time-steps in $\mathcal{T}^k$. The second sub-stage corresponds to the decisions to be made once the actual value of the demand is revealed for each time-step in $\mathcal{T}^k$. These decisions define the production quantity, the inventory level and the lost sales throughout $\mathcal{T}^k$. They can be seen as recourse actions taken to ensure that the setup decisions made in the first sub-stage lead to a feasible production plan.

The dynamics of the decision-making process can thus be described as follows:

Setup decisions for $t \in \mathcal{T}^1 \rightarrow$ Observation of $(\widetilde{d}_t)_{t\in\mathcal{T}^1} \rightarrow$ Recourse actions for $t \in \mathcal{T}^1$
$\cdots$ Setup decisions for $t \in \mathcal{T}^k \rightarrow$ Observation of $(\widetilde{d}_t)_{t\in\mathcal{T}^k} \rightarrow$ Recourse actions for $t \in \mathcal{T}^k$
$\cdots$ Setup decisions for $t \in \mathcal{T}^K \rightarrow$ Observation of $(\widetilde{d}_t)_{t\in\mathcal{T}^K} \rightarrow$ Recourse actions for $t \in \mathcal{T}^K$

Following the terminology used by Carpentier et al. [19] and Martinez Parra et al. [35], our stochastic problem relies on a Decision–Hazard–Decision setting in which decisions relative to a given stage are to be made before and after observing the uncertainty realization for this stage.

In terms of uncertainty modeling, we assume that the discrete-time stochastic process describing the demand evolution over time is stage-wise independent. This means that the random demand vector observed at stage k, $(\widetilde{d}_t)_{t\in\mathcal{T}^k}$, is independent of its past realizations $\left((\widetilde{d}_t)_{t\in\mathcal{T}^1}, (\widetilde{d}_t)_{t\in\mathcal{T}^2}, \ldots, (\widetilde{d}_t)_{t\in\mathcal{T}^{k-1}}\right)$. Furthermore, at each stage k, we represent the potential realizations of the demand over time-interval $\mathcal{T}^k$ by a discrete set of scenarios $\mathcal{U}^k$ of known probability distribution. Scenario $u \in \mathcal{U}^k$ thus corresponds to a possible realization $(d_{u,T^k+1}, d_{u,T^k+2}, \ldots, d_{u,T^{k+1}})$ of the demand during stage k. Note that dependencies between the random demand values within a given stage are allowed, *i.e.*, at stage k the random variables $(\widetilde{d}_t)_{t\in\mathcal{T}^k}$ do not have to be independent.

The above description leads to a mathematical modeling of the multi-stage stochastic programming problem through the following dynamic programming formulation involving recursive equations. Note that a similar approach was used by Naeem et al. [36] and Quezada et al. [41], to formulate multi-stage stochastic lot-sizing problems.

Problem MS-LS-C-L-SUB:

$$\min \sum_{t\in\mathcal{T}^1}\sum_{n\in\mathcal{N}} q_{t,n}\cdot y_{t,n} + \mathbb{E}_{u\in\mathcal{U}^1}\left[r^1(x_u, s_u, l_u) + \Gamma^2(s_{u,T^2})\right] \tag{2a}$$

$$\text{s.t. } s^e + \sum_{n\in\mathcal{N}} x_{u,1,n} = d_{u,1} - l_{u,1} + s_{u,1}, \qquad \forall u \in \mathcal{U}^1 \tag{2b}$$

$$s_{u,t-1} + \sum_{n\in\mathcal{N}} x_{u,t,n} = d_{u,t} - l_{u,t} + s_{u,t}, \qquad \forall u \in \mathcal{U}^1,\ \ \forall t \in \mathcal{T}^1\backslash\{1\} \tag{2c}$$

$$x_{u,t,n} \le C_{t,n}\cdot y_{t,n}, \qquad \forall n \in \mathcal{N},\ \ \forall u \in \mathcal{U}^1,\ \ \forall t \in \mathcal{T}^1 \tag{2d}$$

$$s_{u,t} \le S_t, \qquad \forall u \in \mathcal{U}^1,\ \ \forall t \in \mathcal{T}^1 \tag{2e}$$

$$x_{u,t,n}, s_{u,t}, l_{u,t} \ge 0, \qquad \forall n \in \mathcal{N},\ \ \forall u \in \mathcal{U}^1,\ \ \forall t \in \mathcal{T}^1 \tag{2f}$$

$$y_{t,n} \in \{0, 1\}, \qquad \forall n \in \mathcal{N},\ \ \forall u \in \mathcal{U}^1,\ \ \forall t \in \mathcal{T}^1 \tag{2g}$$

Given a stage k, function r^k gives the total costs of production, inventory holding and lost sales at the current stage k depending on the value of variables x, s and l. Hence, $r^k(x, s, l) = \sum_{t\in\mathcal{T}^k}(h_t\cdot s_t + g_t\cdot l_t + \sum_{n\in\mathcal{N}} p_{t,n}\cdot x_{t,n})$. As for function $\Gamma^k(\cdot)$, it is recursively defined as the value function of the single-stage sub-problem referred to as problem $\mathcal{P}^k(\sigma^k)$ for an entry inventory level σ^k at time-step $T^k + 1$:

Problem $\mathcal{P}^k(\sigma^k)$:

$$\Gamma^k(\sigma^k) = \min \sum_{t\in\mathcal{T}^k}\sum_{n\in\mathcal{N}} q_{t,n}\cdot y_{t,n} + \mathbb{E}_{u\in\mathcal{U}^k}\left[r^k(x_u, s_u, l_u) + \Gamma^{k+1}(s_{u,T^{k+1}})\right] \tag{3a}$$

$$\text{s.t. } \sigma^k + \sum_{n\in\mathcal{N}} x_{u,T^k+1,n} = d_{u,T^k+1} - l_{u,T^k+1} + s_{u,T^k+1}, \qquad \forall u \in \mathcal{U}^k \tag{3b}$$

$$s_{u,t-1} + \sum_{n\in\mathcal{N}} x_{u,t,n} = d_{u,t} - l_{u,t} + s_{u,t}, \qquad \forall u \in \mathcal{U}^k,\ \ \forall t \in \mathcal{T}^k\backslash\{T^k+1\} \tag{3c}$$

$$x_{u,t,n} \le C_{t,n}\cdot y_{t,n}, \qquad \forall n \in \mathcal{N},\ \ \forall u \in \mathcal{U}^k,\ \ \forall t \in \mathcal{T}^k \tag{3d}$$

$$s_{u,t} \le S_t, \qquad \forall u \in \mathcal{U}^k,\ \ \forall t \in \mathcal{T}^k \tag{3e}$$

$$x_{u,t,n}, s_{u,t}, l_{u,t} \ge 0, \qquad \forall n \in \mathcal{N},\ \ \forall u \in \mathcal{U}^k,\ \ \forall t \in \mathcal{T}^k \tag{3f}$$

$$y_{t,n} \in \{0, 1\}, \qquad \forall n \in \mathcal{N}, \quad \forall u \in \mathcal{U}^k, \quad \forall t \in \mathcal{T}^k \tag{3g}$$

where $\Gamma^{K+1}(\cdot) \equiv 0$.

Function $\Gamma^k(\cdot)$ is the expected cost-to-go function estimating the future expected total costs at stage k and beyond depending on the state of the system at the beginning of stage k. In our case, the decision stages are weakly coupled to each other. Indeed, the only coupling comes from the fact that the ending inventory at stage k becomes the entering inventory at stage $k+1$. Hence, the corresponding state space at each stage is of dimension 1 and only comprises the entry inventory level. Consequently, function $\Gamma^k(\cdot)$ only depends on the inventory level at the beginning of stage k, *i.e.*, it is a function of $\sigma^k \in [0, S_{T^k}]$. Note that $\Gamma^{K+1}(\cdot)$, the expected cost-to-go function at the last stage, is equal to 0 everywhere, since the production is not planned beyond stage K.

Problem MS-LS-C-L-SUB thus consists in a sequence of nested single-stage production planning sub-problems. The sub-problem to be solved at stage k, *i.e.*, problem $\mathcal{P}^k(\cdot)$, aims at finding the optimal production decisions for the current stage, given a value of the entering inventory level σ^k and an expected cost-to-go function $\Gamma^{k+1}(\cdot)$ providing an estimation of the future production costs as a function of the inventory level at the end of stage k.

Furthermore, problem $\mathcal{P}^k(\cdot)$ is itself a stochastic programming problem involving two decision sub-stages as mentioned above. The first sub-stage decisions correspond to the setup variables $y_{t,n}$. The second sub-stage decisions corresponds to the recourse actions taken after the demand is revealed. They are modeled by the production quantities variables $x_{u,t,n}$, the inventory level variables $s_{u,t}$, and the lost sales variables $l_{u,t}$ for each considered uncertainty scenario u. The objective function (3a) aims at minimizing the sum of the immediate setup costs to be paid at stage k, of the expected cost of the recourse actions taken at stage k, and of the future expected costs beyond stage k. For each scenario $u \in \mathcal{U}^k$ at the stage $k \in \mathcal{K}$, $r^k(x_u, s_u, l_u)$ represents the cost of the recourse actions as a function of the value of the second sub-stage variables x_u, s_u and l_u, while $\Gamma^{k+1}(s_{u,T^{k+1}})$ estimates the costs beyond stage k as a function of the ending inventory $s_{u,T^{k+1}}$.

Constraints (3b)–(3f) can be seen as second sub-stage constraints. They ensure that the recourse actions taken in each scenario u comply with the setup decisions made at the first sub-stage and with the limited storage capacity, and that these decisions lead to a time evolution of the inventory consistent with the realization of the demand in scenario u. Note that in our setting, the recourse is relatively complete: for any value of the entering inventory σ^k, of our first sub-stage setup decisions, and of the stochastic demand realization, there exist feasible recourse actions at stage k and later.

Our end goal is to determine the optimal setup decisions for the first stage, while taking into account all possible uncertainty realizations and subsequent recourse actions. This could be done directly by solving problem MS-LS-C-L-SUB if a closed-form expression of the expected cost-to-go function $\Gamma^2(\cdot)$ were available. This is however not the case, therefore we investigate in the following section a way to solve Problem MS-LS-C-L-SUB through a backward dynamic programming recursion iteratively building an approximation of the expected cost-to-go function $\Gamma^k(\cdot)$ at each stage k.

Note that there exists an alternative formulation to problem MS-LS-C-L-SUB that would fit a Hazard–Decision information structure. In this case, we would have to consider all the setup decisions relative to stage k as anticipative decisions made at stage $k-1$ and to handle them as state decisions. These decisions correspond to determining the setup state of each resource $n \in \mathcal{N}$ for each time-step $t \in \mathcal{T}^k$, which means that the dimension of the state space would be increased from 1 to $1 + N \cdot |\mathcal{T}^k|$. This would lead to significant numerical difficulties due to the necessity to consider an exponential number $2^{N \cdot |\mathcal{T}^k|}$ of resource setup state combinations while building the approximation of the expected cost-to-go functions. We thus rely in what follows on the Decision–Hazard–Decision formulation presented in this section.

3. An approximate stochastic dynamic programming approach

For any given stage $k \in \mathcal{K}$, optimally solving problem $\mathcal{P}^k(\cdot)$ would require to compute the exact value of the expected cost-to-go function $\Gamma^{k+1}(\cdot)$, for any possible value of the inventory level at time-step T^{k+1}. Furthermore, one would have to efficiently integrate that knowledge into the formulation of problem $\mathcal{P}^k(\cdot)$. This is made especially difficult by the fact that $\Gamma^{k+1}(\cdot)$ is generally a non-linear, non-continuous function defined on a continuous domain $[0, S_{T^{k+1}}]$, for which no obvious closed-form expression is available.

We thus introduce in this section an approximate stochastic dynamic programming approach for this problem. It is based on a backward recursion in which a piece-wise linear approximation of each expected cost-to-go function is iteratively built, starting from the last stage and going back to the first (see, *e.g.*, Powell [39]).

In what follows, we first describe some general characteristics of the expected cost-to-go function. We then explain how the piece-wise linear approximation of each expected cost-to-go function is constructed. Finally, this approximation is leveraged to reformulate each problem $\mathcal{P}^k(\cdot)$, $k \in \mathcal{K}$, as a mixed-integer linear program and to solve MS-LS-C-L-SUB through an approximate stochastic dynamic algorithm.

Let us first discuss the overall characteristics of the cost-to-go function $\Gamma^{k+1}(\cdot)$ for a given stage k. If we assume that the setup costs $q_{t,n}$ are null for each production unit $n \in \mathcal{N}$ and each time-step $t > T^{k+1}$ beyond stage k, the exact expected cost-to-go function $\Gamma^{k+1}(\cdot)$ is the value function of a linear program and is thus a convex piece-wise linear function (see, *e.g.*, Shapiro [49]). However, considering positive setup costs beyond the end of stage k introduces discontinuities in $\Gamma^{k+1}(\cdot)$.

These discontinuities arise for a discrete set of break-point values of the entering inventory. These values are such that the production in a given time-step may be entirely replaced by inventory use so that no more setup costs are to be

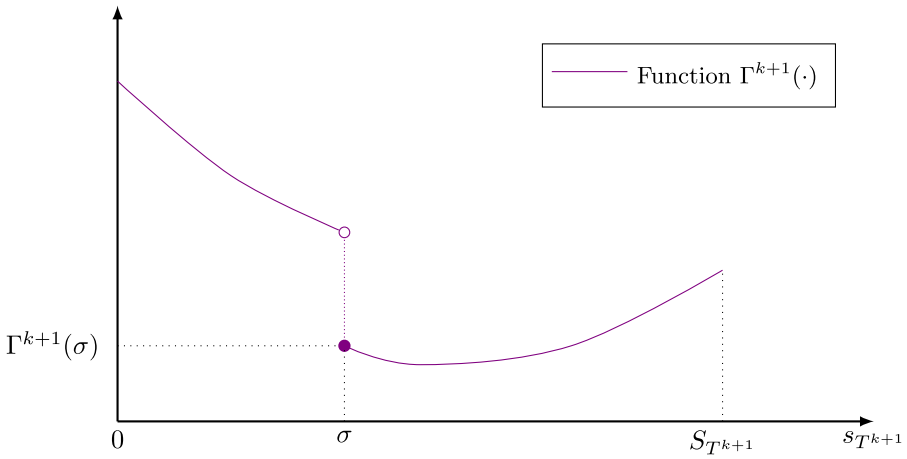

Fig. 1. Illustration of the exact cost-to-go function.

paid in this time-step. At each of these break-point values, function $\Gamma(\cdot)$ presents a step discontinuity, corresponding to a sharp decrease in the cost due to the elimination of a setup in the production plan. As a result, $\Gamma(\cdot)$ is continuous and convex between two break-points, but not necessarily across one. In other words, function $\Gamma(\cdot)$ is a piece-wise continuous piece-wise convex function.

A cost-to-go function is illustrated in Fig. 1. Here, the cost-to-go function $\Gamma^{k+1}(\cdot)$ is defined on a bounded interval $[0, S_{T^{k+1}}]$, with a single step discontinuity at inventory level $s_{T^{k+1}} = \sigma$, such that $\Gamma^{k+1}(\cdot)$ is continuous and convex in intervals $[0, \sigma)$ and $[\sigma, S_{T^{k+1}}]$ but not in interval $[0, S_{T^{k+1}}]$.

We derive from that study that a piece-wise linear function may be effective to approximate the expected cost-to-go function. Let us assume that, for a given stage $k \in \mathcal{K}$, we are able to compute the value of the expected cost-to-go function $\Gamma^{k+1}(\cdot)$ for any given entering inventory value $\sigma^{k+1} \in [0, S_{T^{k+1}}]$. We propose to construct a piece-wise linear approximation $\widetilde{\Gamma}^{k+1}(\cdot)$ of $\Gamma^{k+1}(\cdot)$, by decomposing the interval $[0, S_{T^{k+1}}]$ into a finite set $\mathcal{I}^{k+1} = \{1, \ldots, I^{k+1}\}$ of sub-intervals.

Let δ_i^{k+1} denote the width of interval $i \in \mathcal{I}^{k+1}$, then $\sigma_i^{k+1} = \sigma_{i-1}^{k+1} + \delta_i^{k+1}$ is the inventory level value corresponding to the supremum of sub-interval $i \in \mathcal{I}^{k+1}$, with $\sigma_0^{k+1} = 0$ and $\sigma_{I^{k+1}}^{k+1} = S_{T^{k+1}}$. Let Γ_i^{k+1} represent the value of function $\Gamma^{k+1}(\cdot)$ evaluated in σ_i^{k+1}, *i.e.*, $\Gamma_i^{k+1} = \Gamma^{k+1}(\sigma_i^{k+1})$. The slope of the piece-wise linear approximation of $\Gamma^{k+1}(\cdot)$ over interval $i \in \mathcal{I}^{k+1}$ is then given by $\kappa_i^{k+1} = (\Gamma_i^{k+1} - \Gamma_{i-1}^{k+1})/\delta_i^{k+1}$.

Finally, the approximate cost-to-go function $\widetilde{\Gamma}^{k+1}(\cdot)$ is defined over $[0, S_{T^{k+1}}]$ by

$$\widetilde{\Gamma}^{k+1}(s) = \Gamma_0^{k+1} + \sum_{i \in \mathcal{I}^{k+1}} \widetilde{\Gamma}_i^{k+1}(s)$$

with function $\widetilde{\Gamma}_i^{k+1}(\cdot)$, $i \in \mathcal{I}^{k+1}$, given by:

$$\widetilde{\Gamma}_i^{k+1}(s) = \begin{cases} 0, & \text{if } s < \sigma_{i-1}^{k+1}, \\ (s - \sigma_{i-1}^{k+1}) \cdot \kappa_i^{k+1}, & \text{if } s \in [\sigma_{i-1}^{k+1}, \sigma_i^{k+1}], \\ 0, & \text{if } s > \sigma_i^{k+1}. \end{cases}$$

Note that function $\widetilde{\Gamma}^{k+1}(\cdot)$ is generally neither an under- nor an over-approximation of the expected cost-to-go function $\Gamma^{k+1}(\cdot)$. In case the exact function $\Gamma^{k+1}(\cdot)$ is convex, $\widetilde{\Gamma}^{k+1}(\cdot)$ is naturally an over-approximation of $\Gamma^{k+1}(\cdot)$. However, in any other case, in particular when the exact function $\Gamma^{k+1}(\cdot)$ presents some discontinuities as discussed above, function $\widetilde{\Gamma}^{k+1}(\cdot)$ is only piece-wise convex and is not guaranteed to be an over-approximation of $\Gamma^{k+1}(\cdot)$.

This is illustrated by Fig. 2. Following the example in Fig. 1, we build an approximation of the cost-to-go function $\Gamma^{k+1}(\cdot)$ over the bounded interval $[0, S_{T^{k+1}}]$. To this aim, the interval $[0, S_{T^{k+1}}]$ is decomposed into four sub-intervals $[0, \sigma_1^{k+1}]$, $[\sigma_1^{k+1}, \sigma_2^{k+1}]$, $[\sigma_2^{k+1}, \sigma_3^{k+1}]$ and $[\sigma_3^{k+1}, S_{T^{k+1}}]$. Then, function $\Gamma^{k+1}(\cdot)$ is evaluated at each break-point in $\{0, \sigma_1^{k+1}, \sigma_2^{k+1}, \sigma_3^{k+1}, S_{T^{k+1}}\}$. Finally, the piece-wise linear approximation $\widetilde{\Gamma}^{k+1}(\cdot)$ is derived from these evaluations as discussed above. Clearly, $\widetilde{\Gamma}^{k+1}(\cdot)$ is piece-wise convex, here respectively in intervals $[0, \sigma_1^{k+1}]$ and $[\sigma_1^{k+1}, S_{T^{k+1}}]$. However, over the whole interval $[0, S_{T^{k+1}}]$, it is neither convex nor an under- or over-approximation of function $\Gamma^{k+1}(\cdot)$.

By replacing $\Gamma^{k+1}(\cdot)$ by its approximation $\widetilde{\Gamma}^{k+1}(\cdot)$ in the objective function of problem $\mathcal{P}^k(\cdot)$ and using an incremental method to handle the piece-wise linear function $\widetilde{\Gamma}^{k+1}(\cdot)$, we obtain the following formulation of the approximate single-stage sub-problem to be solved at stage k, denoted as problem $\widetilde{\mathcal{P}}^k(\sigma^k)$.

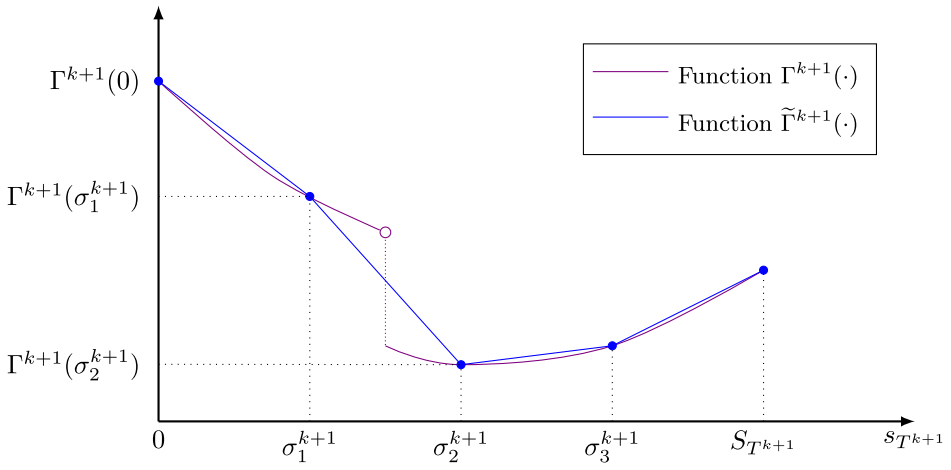

Fig. 2. Illustration of the approximate piece-wise linear cost-to-go function.

Problem $\widetilde{\mathcal{P}}^k(\sigma^k)$:

$$\min \sum_{t\in\mathcal{T}^k}\sum_{n\in\mathcal{N}} q_{t,n}\cdot y_{t,n} + \mathbb{E}_{u\in\mathcal{U}^k}\left[r^k(x_u, s_u, l_u) + \Gamma_0^{k+1} + \sum_{i\in\mathcal{I}^{k+1}} \kappa_i^{k+1}\cdot w_{u,i}\right] \tag{4a}$$

s.t. (3b)–(3g),

$$\sum_{i\in\mathcal{I}^{k+1}} w_{u,i} = s_{u,T^{k+1}}, \qquad \forall u\in\mathcal{U}^k \tag{4b}$$

$$z_{u,i-1} \geq z_{u,i}, \qquad \forall i\in\mathcal{I}^{k+1}\backslash\{1\},\ \ \forall u\in\mathcal{U}^k \tag{4c}$$

$$w_{u,i-1} \geq \delta_{i-1}^{k+1}\cdot z_{u,i}, \qquad \forall i\in\mathcal{I}^{k+1}\backslash\{1\},\ \ \forall u\in\mathcal{U}^k \tag{4d}$$

$$w_{u,i} \leq \delta_i^{k+1}\cdot z_{u,i}, \qquad \forall i\in\mathcal{I}^{k+1},\ \ \forall u\in\mathcal{U}^k \tag{4e}$$

$$z_{u,i}\in\{0,1\},\ w_{u,i}\geq 0, \qquad \forall i\in\mathcal{I}^{k+1},\ \ \forall u\in\mathcal{U}^k \tag{4f}$$

The incremental model used to handle the piece-wise linear function $\widetilde{\Gamma}^{k+1}(\cdot)$ originates from the work of Markowitz and Manne [34] and computes the value of this function as the sum of incremental quantities in each segment. This formulation involves a continuous variable $w_{u,i}$ and a binary variable $z_{u,i}$ for each scenario $u\in\mathcal{U}^k$ and each segment $i\in\mathcal{I}^{k+1}$. More precisely, variable $z_{u,i}$ is equal to 1 if $s_{u,T^{k+1}}\geq\sigma_{i-1}^{k+1}$, *i.e.*, if segment i is involved in the computation of $\widetilde{\Gamma}^{k+1}(s_{u,T^{k+1}})$, and $z_{u,i}=0$ otherwise. Variable $w_{u,i}$ represents the incremental amount of inventory in interval $[\sigma_{i-1}^{k+1},\sigma_i^{k+1}]$ used to compute $s_{u,T^{k+1}}$.

Constraints (4b) thus compute for each scenario u the ending inventory value $s_{u,T^{k+1}}$ as the sum, over all segments $i\in\mathcal{I}^{k+1}$, of the incremental amount $w_{u,i}$ of inventory used in segment i. Constraints (4c) ensure, for each scenario u, that, if segment i is involved in the computation of $\widetilde{\Gamma}^{k+1}(s_{u,T^{k+1}})$, then all segments to its left are also involved. Constraints (4d) guarantee that segment $i-1$ is fully used, *i.e.*, that $w_{u,i-1}=\delta_{i-1}^{k+1}$, if segment i is used. Constraints (4e) impose that $w_{u,i}$ stays below its maximum acceptable value δ_i^{k+1} if segment i is used and is equal to 0 otherwise. Finally, in the objective function (4a), the value of $\widetilde{\Gamma}^{k+1}(s_{u,T^{k+1}})$ in scenario u is computed as the sum, over each segment i, of the incremental cost in each segment $i\in\mathcal{I}^{k+1}$, *i.e.*, $\widetilde{\Gamma}^{k+1}(s_{u,T^{k+1}}) = \Gamma_0^{k+1} + \sum_{i\in\mathcal{I}^{k+1}}\kappa_i^{k+1}\cdot w_{u,i}$.

The use of the proposed piece-wise linear approximation of $\Gamma^{k+1}(\cdot)$ allows us to unnest the two minimization problems involved in the objective function of $\mathcal{P}^k(\sigma^k)$. The approximate problem $\widetilde{\mathcal{P}}^k(\sigma^k)$ is namely a stand-alone MILP which can be solved directly, *e.g.*, by a mathematical programming solver. This means that, if we are able to compute the value of the expected cost-to-go function $\Gamma^{k+1}(\cdot)$ for any given entering inventory value σ^{k+1} at a given stage $k+1\in\mathcal{K}$, then we are able to compute the value of $\widetilde{\Gamma}^k(\cdot)$ for any entry inventory value σ^k by solving problem $\widetilde{\mathcal{P}}^k(\sigma^k)$.

A key observation is that, for any inventory level σ^{k+1}, the optimal objective of problem $\widetilde{\mathcal{P}}^{k+1}(\sigma^{k+1})$ is an estimation of the value $\Gamma^{k+1}(\sigma^{k+1})$. Hence, function $\Gamma^{k+1}(\cdot)$ can be estimated by solving $\widetilde{\mathcal{P}}^{k+1}(\cdot)$ at each break-point σ_i^{k+1} for $i\in\mathcal{I}^{k+1}$ and by building the corresponding piece-wise linear approximation. Since $\Gamma^{K+1}(\cdot)\equiv 0$, it is thus possible to dynamically deduce a piece-wise linear approximation of all expected cost-to-go functions, starting from the last stage K and going backward up to the first stage, by solving multiple MILPs at each stage.

This dynamic procedure leads to a description of the approximate expected cost-to-go function $\widetilde{\Gamma}^2(\cdot)$, under the form of a constant initial value Γ_0^2 and a slope κ_i^2 on each sub-interval $i \in \mathcal{I}^2$ of length δ_i^2 of interval $[0, S_{T^2}]$. From this description, we obtain the following approximate formulation of problem MS-LS-C-L-SUB.

Approximate Problem MS-LS-C-L-SUB :

$$\min \sum_{t\in\mathcal{T}^1}\sum_{n\in\mathcal{N}} q_{t,n}\cdot y_{t,n} + \mathbb{E}_{u\in\mathcal{U}^1}\left[r^1(x_u, s_u, l_u) + \Gamma_0^2 + \sum_{i\in\mathcal{I}^2} \kappa_i^2 \cdot w_{u,i} \right] \tag{5a}$$

s.t. (2b)–(2g),

$$\sum_{i\in\mathcal{I}^2} w_{u,i} = s_{u,T^2}, \qquad \forall u \in \mathcal{U}^1$$

$$z_{u,i-1} \geq z_{u,i}, \qquad \forall i \in \mathcal{I}^2\backslash\{1\},\ \ \forall u \in \mathcal{U}^1 \tag{5b}$$

$$w_{u,i-1} \geq \delta_{i-1}^2 \cdot z_{u,i}, \qquad \forall i \in \mathcal{I}^2\backslash\{1\},\ \ \forall u \in \mathcal{U}^1 \tag{5c}$$

$$w_{u,i} \leq \delta_i^2 \cdot z_{u,i}, \qquad \forall i \in \mathcal{I}^2,\ \ \forall u \in \mathcal{U}^1 \tag{5d}$$

$$z_{u,i} \in \{0, 1\},\ w_{u,i} \geq 0, \qquad \forall i \in \mathcal{I}^2,\ \ \forall u \in \mathcal{U}^1 \tag{5e}$$

The approximate dynamic stochastic programming algorithm proposed to solve the Approximate Problem MS-LS-C-L-SUB is formally described below.

Algorithm 1: Approximate stochastic DP algorithm to solve problem MS-LS-C-L-SUB

Input: Parameters $(q, h, g, p, s^e, d, C, S)$, set $\mathcal{K}$ of stages, time interval $\mathcal{T}^k$ for each $k \in \mathcal{K}$, number of segments I^k for each $k \in \mathcal{K}$.

Output: Setup decisions to be implemented for Stage 1.

Set $k \leftarrow K$ and $\widetilde{\Gamma}^{K+1}(\cdot) \leftarrow 0$.
while $k > 1$ **do**
  Find value Γ_0^k (and resp. $\widetilde{\Gamma}_0^k$) by solving problem $\widetilde{\mathcal{P}}^k(0)$.
  Sample I^k inventory values, $(\sigma_i^k)_{i\in\mathcal{I}^k}$, from interval $[0, S_{T^k}]$ such that $\sigma_0^k = 0$ and $\sigma_{I^k}^k = S_{T^k}$.
  for $i \in \mathcal{I}^k$ **do**
    Find value $\widetilde{\Gamma}_i^k$ by solving problem $\widetilde{\mathcal{P}}^k(\sigma_i^k)$.
    Set $\delta_i^k \leftarrow \sigma_i^k - \sigma_{i-1}^k$ and $\kappa_i^k \leftarrow (\widetilde{\Gamma}_i^k - \widetilde{\Gamma}_{i-1}^k)/\delta_i^k$.
  Set $k \leftarrow k - 1$.
Solve Approximate Problem MS-LS-C-L-SUB and retrieve setup decisions involved in Stage 1.

Since the problem has a relatively complete recourse, the solution provided by Algorithm 1, *i.e.*, the setup decisions relative to all time-steps in $\mathcal{T}^1$, is guaranteed to be a feasible solution of Problem MS-LS-C-L-SUB. This solution is nevertheless a sub-optimal solution of Problem MS-LS-C-L-SUB, the quality of which is difficult to accurately measure. Furthermore, it is difficult to estimate how the error propagates from the expected cost-to-go function estimate $\widetilde{\Gamma}^k(\cdot)$ to the one at the previous stage $\widetilde{\Gamma}^{k-1}(\cdot)$. However, the accuracy level of the piece-wise linear approximate of each expected cost-to-go function $\Gamma^k(\cdot)$, $k \in \mathcal{K}$, directly depends on the number of break-points I^k used for its construction. In other words, the number of break-points I^k is a heuristic measure of the precision of the proposed solution method. Increasing I^k will thus have a direct positive impact on the accuracy of the estimation of the cost-to-go function $\Gamma^k(\cdot)$, and consequently on the quality of the first-stage solution provided by Algorithm 1.

4. A time-efficient Benders decomposition algorithm

Running Algorithm 1 requires to solve $\sum_{k=2}^{K} I^k + 1$ mixed-integer linear programs (MILP): one to solve each approximate problem $\widetilde{\mathcal{P}}^k(\sigma_i^k)$, $k \in \{2, \ldots, K\}$, $i \in \mathcal{I}^k$, and one to solve the approximate problem MS-LS-C-L-SUB. As discussed above, the number of segments I^k used to approximate each expected cost-to-go function $\Gamma^k(\cdot)$, $k \in \{2, \ldots, K\}$, should be as high as possible to ensure a good quality of the obtained first-stage solution. Consequently, a significant number of MILPs has to be solved over the course of Algorithm 1.

Furthermore, each approximate problem $\widetilde{\mathcal{P}}^k(\sigma_i^k)$ is itself a two-stage stochastic program whose size is broadly proportional to the number of scenarios in the set $\mathcal{U}^k$. As a large number of scenarios may be needed to accurately represent the uncertainty distribution, this means that each of the $\sum_{k=2}^{K} I^k + 1$ MILPs to be solved during Algorithm 1 involves a large number of variables and constraints. This may lead to significant numerical difficulties and prohibitive computation times.

In order to overcome this difficulty, we propose in this section to leverage the block-decomposable structure of each approximate problem $\widetilde{\mathcal{P}}^k(\sigma_i^k)$. Indeed, this structure is particularly suitable for a Benders decomposition approach. To implement this decomposition scheme, we propose to further approximate the expected cost-to-go functions, transforming

the presented approach into an inherently heuristic method. It is nevertheless proving effective in providing good solutions within a reasonable computation time for large-size instances of the problem, which would be otherwise numerically intractable.

In what follows, for the sake of readability, we drop index k and consider a generic approximate problem denoted as $\widetilde{\mathcal{P}}$ with a fixed entry inventory value s^e. Firstly, we present a reformulation of this problem highlighting its decomposable structure. Secondly, we introduce a convex under-approximation of the expected cost-to-go function and reformulate the second-stage sub-problems as linear programs. Thirdly, we derive from that reformulation a classical Benders decomposition scheme. Finally, we show how the dual of each relaxed second-stage sub-problem can be efficiently solved in polynomial time.

4.1. Reformulation of the generic problem $\widetilde{\mathcal{P}}$

Problem $\widetilde{\mathcal{P}}$ consists in planning production over a set $\mathcal{T} = \{1, \ldots, T\}$ of time-steps. Parameter s^e denotes the initial inventory value, *i.e.*, the inventory level at the beginning of time-step 1. The possible realizations of the stochastic demand over $\mathcal{T}$ are represented by a set of scenarios $\mathcal{U} = \{1, \ldots, U\}$. The impact of the production planning decisions on the future costs are represented by the piece-wise linear approximate expected cost-to-go function $\widetilde{\Gamma}(\cdot)$. The allowed interval for the ending inventory value $[0, S_T]$ is cut out into I sub-intervals, and each sub-interval $i \in \mathcal{I}$ is described by its two extreme points $[\sigma_{i-1}, \sigma_i]$, with $\sigma_0 = 0$ and $\sigma_I = S_T$. The piece-wise linear function $\widetilde{\Gamma}(\cdot)$ is defined over the set $\mathcal{I} = \{1, \ldots, I\}$ of sub-intervals, so that its slope over sub-interval $i \in \mathcal{I}$ of width $\delta_i = \sigma_i - \sigma_{i-1}$ is denoted by κ_i. Finally, the value $\widetilde{\Gamma}(0)$ of expected cost-to-go function for a zero value initial inventory level is denoted by Γ_0.

Problem $\widetilde{\mathcal{P}}$ can be formulated using only the first-stage decision variables $y_{t,n}$ as follows:

$$\min \sum_{t\in\mathcal{T}} \sum_{n\in\mathcal{N}} q_{t,n} \cdot y_{t,n} + \mathbb{E}_{u\in\mathcal{U}}\Big[\Gamma_0 + \mathcal{Q}(y, u)\Big] \tag{6a}$$

$$\text{s.t. } y_{t,n} \in \{0, 1\}, \qquad \forall t \in \mathcal{T}, \quad \forall n \in \mathcal{N} \tag{6b}$$

Here, $\mathcal{Q}(y^*, u)$ denotes the optimal value of the second-stage sub-problem to be solved for scenario $u \in \mathcal{U}$ when the production setup is given by y^*. This sub-problem aims at finding the optimal recourse decisions for the realization of a single scenario $u \in \mathcal{U}$. Hence, its variables are not indexed by the scenario number, and the sub-problem $\widetilde{SP}(y^*, u)$ is defined as follows:

Sub-problem $\widetilde{SP}(y^*, u)$:

$$\mathcal{Q}(y^*, u) = \min \sum_{t\in\mathcal{T}} \left(h_t \cdot s_t + g_t \cdot l_t + \sum_{n\in\mathcal{N}} p_{t,n} \cdot x_{t,n} \right) + \sum_{i\in\mathcal{I}} \kappa_i \cdot w_i \tag{7a}$$

$$\text{s.t. } s^e + \sum_{n\in\mathcal{N}} x_{1,n} = d_{u,1} - l_1 + s_1, \tag{7b}$$

$$s_{t-1} + \sum_{n\in\mathcal{N}} x_{t,n} = d_{u,t} - l_t + s_t, \qquad \forall t \in \mathcal{T}\backslash\{1\} \tag{7c}$$

$$x_{t,n} \le C_{t,n} \cdot y^*_{t,n}, \qquad \forall n \in \mathcal{N}, \quad \forall t \in \mathcal{T} \tag{7d}$$

$$s_t \le S_t, \qquad \forall t \in \mathcal{T} \tag{7e}$$

$$\sum_{i\in\mathcal{I}} w_i = s_T, \tag{7f}$$

$$z_{i-1} \ge z_i, \qquad \forall i \in \mathcal{I}\backslash\{1\} \tag{7g}$$

$$w_{i-1} \ge \delta_{i-1} \cdot z_i, \qquad \forall i \in \mathcal{I}\backslash\{1\} \tag{7h}$$

$$w_i \le \delta_i \cdot z_i, \qquad \forall i \in \mathcal{I} \tag{7i}$$

$$s_t, l_t \ge 0, \qquad \forall t \in \mathcal{T} \tag{7j}$$

$$x_{t,n} \ge 0, \qquad \forall n \in \mathcal{N}, \quad \forall t \in \mathcal{T} \tag{7k}$$

$$z_i \in \{0, 1\}, w_i \ge 0, \qquad \forall i \in \mathcal{I} \tag{7l}$$

This second-stage sub-problem thus seeks to find the recourse actions (quantity to be produced, inventory level and lost-sales) to be taken at each time-step in $\mathcal{T}$ so as to find a feasible production plan complying both with the demand realization in scenario u and with the first-stage setup decisions y^*. Its objective function (7a) aims at minimizing the sum of the immediate costs, *i.e.*, the production, inventory, and lost-sales costs over the planning horizon $\mathcal{T}$, and of the future expected costs, *i.e.*, the costs to be paid beyond the planning horizon.

Reformulation (6a)–(6b) clearly shows that, if the setup decisions are fixed, problem $\widetilde{\mathcal{P}}$ decomposes into U independent single-scenario sub-problems. However, due to the presence of the binary variable z_i, each of these sub-problems is an MILP. Directly applying a Benders decomposition scheme on problem $\widetilde{\mathcal{P}}$ is therefore not straightforward as it requires to

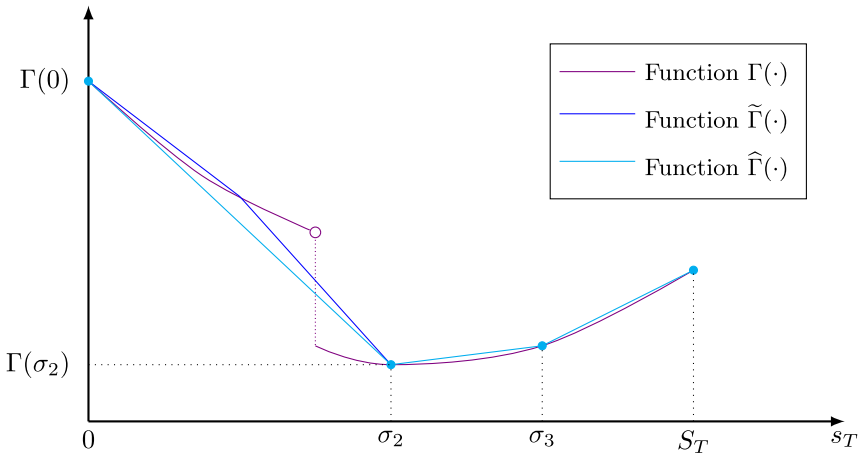

Fig. 3. Illustration of the convex approximate piece-wise linear cost-to-go function.

formulate the dual of each sub-problem. We thus discuss in the following subsection a convex under-approximation of $\widetilde{\Gamma}(\cdot)$, which will allow us to formulate the second-stage sub-problem as linear program.

4.2. Continuous approximation of the second-stage sub-problem

The proposed continuous approximation of the second-stage sub-problem relies on a convex under approximation of the approximate cost-to-go function $\widetilde{\Gamma}(\cdot)$.

As discussed in Section 3, the exact expected cost-to-go function $\Gamma(\cdot)$ is discontinuous piece-wise continuous piece-wise convex. This observation leads to the assumption that a convex piece-wise linear approximation of the cost-to-go function may be constructed with fair precision, despite its original non-convex shape. Hence, we propose to replace the approximate cost-to-go function $\widetilde{\Gamma}(\cdot)$ by a convex piece-wise linear under-approximation of $\widetilde{\Gamma}(\cdot)$ denoted as function $\widehat{\Gamma}(\cdot)$.

Such a function is obtained by constructing the convex hull of the set of break-points of $\widetilde{\Gamma}(\cdot)$, $\{(\sigma_i, \widetilde{\Gamma}(\sigma_i)) : i \in \mathcal{I}\}$, using, for example, Jarvis algorithm (see Jarvis [29]), and by selecting the subset of break-points of $\widetilde{\Gamma}(\cdot)$ belonging to this convex hull. Let $\{(\widehat{\sigma}_i, \widetilde{\Gamma}(\widehat{\sigma}_i)) : i \in \widehat{\mathcal{I}}\}$ be the subset of $\widehat{I}$ ($\widehat{I} \leq I$) break-points of $\widetilde{\Gamma}(\cdot)$ belonging to this convex hull, with $\widehat{\sigma}_0 = 0$ and $\widehat{\sigma}_{\widehat{I}} = S_T$. The set of points $\{\widehat{\sigma}_i : i \in \widehat{\mathcal{I}}\}$ defines a second sub-division of interval $[0, S_T]$. $\widehat{\Gamma}(\cdot)$ is constructed as a piece-wise linear function such that, for each $i \in \widehat{\mathcal{I}}$, $\widehat{\Gamma}(\cdot)$ is linear over each sub-interval $[\widehat{\sigma}_{i-1}, \widehat{\sigma}_i]$ of length $\widehat{\delta}_i$, with a slope $\widehat{\kappa}_i = (\widetilde{\Gamma}(\widehat{\sigma}_i)) - (\widetilde{\Gamma}(\widehat{\sigma}_{i-1}))/\widehat{\delta}_i$.

This is depicted by Fig. 3. Following the example in Fig. 2, we consider the set of break-points of $\widetilde{\Gamma}(\cdot)$ comprising $(0, \widetilde{\Gamma}(0))$, $(S_T, \widetilde{\Gamma}(S_T))$ and $(\sigma_i, \widetilde{\Gamma}(\sigma_i))$ for all $i \in \{1, 2, 3\}$. We retain only those belonging to the convex hull of this set, *i.e.*, $(0, \widetilde{\Gamma}(0))$, $(\sigma_2, \widetilde{\Gamma}(\sigma_2))$, $(\sigma_3, \widetilde{\Gamma}(\sigma_3))$ and $(S_T, \widetilde{\Gamma}(S_T))$, and use them to define the convex piece-wise linear function $\widehat{\Gamma}(\cdot)$ over the bounded interval $[0, S_T]$. Fig. 3 shows that $\widehat{\Gamma}(\cdot)$ is an approximation of the exact cost-to-go function $\Gamma(\cdot)$, as well as a convex under-approximation of the approximate cost-to-go function $\widetilde{\Gamma}(\cdot)$.

We now replace function $\widetilde{\Gamma}(\cdot)$ by its convex under-approximation $\widehat{\Gamma}(\cdot)$ in the objective function of $\widetilde{SP}(y^*, u)$. As $\widehat{\Gamma}(\cdot)$ is convex, the $\widehat{I}$ segments will always be considered in the correct order in the incremental model used to handle $\widehat{\Gamma}(\cdot)$ as a piece-wise linear function in the mathematical formulation. This means that the binary variables z are redundant and can be eliminated from the formulation. We can thus reformulate the second-stage sub-problem as the following linear program denoted as

Sub-problem $\widehat{SP}(y^*, u)$:

$$\widehat{\mathcal{Q}}(y^*, u) = \min \sum_{t \in \mathcal{T}} \left(h_t \cdot s_t + g_t \cdot l_t + \sum_{n \in \mathcal{N}} p_{t,n} \cdot x_{t,n} \right) + \sum_{i \in \widehat{\mathcal{I}}} \widehat{\kappa}_i \cdot w_i \tag{8a}$$

$$\text{s.t. } s^e + \sum_{n \in \mathcal{N}} x_{1,n} = d_{u,1} - l_1 + s_1, \tag{8b}$$

$$s_{t-1} + \sum_{n \in \mathcal{N}} x_{t,n} = d_{u,t} - l_t + s_t, \qquad \forall t \in \mathcal{T} \backslash \{1\} \tag{8c}$$

$$x_{t,n} \leq C_{t,n} \cdot y^*_{t,n}, \qquad \forall n \in \mathcal{N}, \quad \forall t \in \mathcal{T} \tag{8d}$$

$$s_t \leq S_t, \qquad \forall t \in \mathcal{T} \tag{8e}$$

$$\sum_{i\in\widehat{\mathcal{I}}} w_i = s_T, \tag{8f}$$

$$w_i \le \widehat{\delta}_i, \qquad \forall i \in \widehat{\mathcal{I}} \tag{8g}$$

$$s_t, l_t \ge 0, \qquad \forall t \in \mathcal{T} \tag{8h}$$

$$x_{t,n} \ge 0, \qquad \forall n \in \mathcal{N},\ \ \forall t \in \mathcal{T} \tag{8i}$$

$$w_i \ge 0, \qquad \forall i \in \widehat{\mathcal{I}} \tag{8j}$$

If the exact cost-to-go function $\Gamma(\cdot)$ was convex, function $\widehat{\Gamma}(\cdot)$ would be an approximation of $\Gamma(\cdot)$ as precise as function $\widetilde{\Gamma}(\cdot)$. However, as previously mentioned, positive setup costs introduce non-convexities in function $\Gamma(\cdot)$. Replacing $\widetilde{\Gamma}(\cdot)$ by $\widehat{\Gamma}(\cdot)$ will thus lead to an inevitable degradation in the precision of the representation of the future costs in problem $\widetilde{\mathcal{P}}$, especially when the setup costs are high compared to the production and holding costs. Thus, the proposed convex under-approximation might under-perform when applied to extreme cases where setup costs dominate production.

Nonetheless, this convex under-approximation leads to the formulation of a continuous second-stage sub-problem and the implementation of a Benders decomposition scheme. From a computational point of view, it enables the consideration of large instances with a high number of scenarios, *i.e.*, with a more detailed modeling of the probability distribution of the uncertain demand. Hence, the loss of accuracy in terms of representation of the future costs is counterbalanced by a precise representation of the uncertainty at each stage. This original contribution may also apply for more general multi-stage stochastic decision–hazard–decision problems.

4.3. Benders decomposition algorithm

We discuss how to solve the approximation of problem $\widetilde{\mathcal{P}}$ obtained by replacing $\mathcal{Q}(y^*, u)$ by the convex under-approximation $\widehat{\mathcal{Q}}(y^*, u)$ for each scenario $u \in \mathcal{U}$. Let us denote by problem $\widehat{\mathcal{P}}$ the following problem:

$$\min \sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}} q_{t,n}\cdot y_{t,n} + \mathbb{E}_{u\in\mathcal{U}}\left[\Gamma_0 + \widehat{\mathcal{Q}}(y,u)\right] \tag{9a}$$

$$\text{s.t. } y_{t,n} \in \{0,1\}, \qquad \forall t \in \mathcal{T},\ \ \forall n \in \mathcal{N} \tag{9b}$$

$\widehat{SP}(y^*, u)$ is a linear program whose dual problem $\widehat{DSP}(y^*, u)$ is given by:

$$\widehat{\mathcal{Q}}(y^*,u) = \max \left\{ s^e\cdot\alpha_1 + \sum_{t\in\mathcal{T}}\Big[-d_{u,t}\cdot\alpha_t - \sum_{n\in\mathcal{N}} C_{t,n}\cdot y^*_{t,n}\cdot\beta_{t,n} - S_t\cdot\eta_t \Big] - \sum_{i\in\widehat{\mathcal{I}}}\widehat{\delta}_i\cdot\mu_i \right\} \tag{10a}$$

$$\text{s.t. } -\alpha_{t+1} + \alpha_t - \eta_t \le h_t, \qquad \forall t \in \mathcal{T}\backslash\{T\} \quad [s_t] \tag{10b}$$

$$\lambda + \alpha_T - \eta_T \le h_T, \qquad [s_T] \tag{10c}$$

$$-\alpha_t - \beta_{t,n} \le p_{t,n}, \qquad \forall n \in \mathcal{N},\ \ \forall t \in \mathcal{T} \quad [x_{t,n}] \tag{10d}$$

$$-\alpha_t \le g_t, \qquad \forall t \in \mathcal{T} \quad [l_t] \tag{10e}$$

$$-\lambda - \mu_i \le \widehat{\kappa}_i, \qquad \forall i \in \widehat{\mathcal{I}} \quad [w_i] \tag{10f}$$

$$\mu_i \ge 0, \qquad \forall i \in \widehat{\mathcal{I}} \tag{10g}$$

$$\eta_t \ge 0, \qquad \forall t \in \mathcal{T} \tag{10h}$$

$$\beta_{t,n} \ge 0, \qquad \forall t \in \mathcal{T},\ \ \forall n \in \mathcal{N} \tag{10i}$$

where

- $\alpha \in \mathbb{R}^T$ are the dual variables of constraints (8b)–(8c),
- $\beta \in \mathbb{R}_+^{T\times N}$ are the dual variables of constraints (8d),
- $\eta \in \mathbb{R}_+^T$ are the dual variables of constraints (8e),
- $\lambda \in \mathbb{R}$ is the dual variable of constraint (8f),
- $\mu \in \mathbb{R}_+^I$ are the dual variable of constraints (8g).

Let us denote by $\Pi(u)$ the polyhedron corresponding to the feasible space of $\widehat{DSP}(y^*, u)$. Let $\mathcal{E}(u)$ be the finite set of extreme points of $\Pi(u)$ and $(\alpha^\epsilon, \beta^\epsilon, \eta^\epsilon, \lambda^\epsilon, \mu^\epsilon)$ be an extreme point belonging to $\mathcal{E}(u)$. An optimal solution of $\widehat{DSP}(y^*, u)$ can be found at one of these extreme points. We can thus reformulate $\widehat{DSP}(y^*, u)$ as

$$\widehat{\mathcal{Q}}(y^*,u) = \max_{\epsilon\in\mathcal{E}(u)} \left\{ s^e\cdot\alpha^\epsilon_1 + \sum_{t\in\mathcal{T}}\Big[-d_{u,t}\cdot\alpha^\epsilon_t - \sum_{n\in\mathcal{N}} C_{t,n}\cdot y^*_{t,n}\cdot\beta^\epsilon_{t,n} - S_t\cdot\eta^\epsilon_t \Big] - \sum_{i\in\widehat{\mathcal{I}}}\widehat{\delta}_i\cdot\mu^\epsilon_i \right\}. \tag{11}$$

We replace $\widehat{\mathcal{Q}}(y, u)$ in the objective function of problem $\widehat{\mathcal{P}}$ by its expression (11) and introduce a continuous variable Q_u to linearize the obtained expression. This leads to the following reformulation of problem $\widehat{\mathcal{P}}$.

$$\min \sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}} q_{t,n}\cdot y_{t,n} + \mathbb{E}_{u\in\mathcal{U}}\Big[\Gamma_0 + Q_u\Big] \tag{12a}$$

$$\text{s.t. } y_{t,n}\in\{0,1\}, \qquad \forall t\in\mathcal{T},\quad \forall n\in\mathcal{N} \tag{12b}$$

$$Q_u \geq \sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}} \pi^{\epsilon}_{u,t,n}\cdot y_{t,n} + \rho^{\epsilon}_u, \qquad \forall \epsilon\in\mathcal{E}(u),\quad \forall u\in\mathcal{U} \tag{12c}$$

where coefficients $\pi^{\epsilon}_{u,t,n}$ and ρ^{ϵ}_u are defined by:

$$\pi^{\epsilon}_{u,t,n} = -C_{t,n}\cdot\beta^{\epsilon}_{t,n}, \tag{13}$$

$$\rho^{\epsilon}_u = s^e\cdot\alpha^{\epsilon}_1 + \sum_{t\in\mathcal{T}}\Big[-d_{u,t}\cdot\alpha^{\epsilon}_t - S_t\cdot\eta^{\epsilon}_t\Big] - \sum_{i\in\widehat{\mathcal{I}}}\widehat{\delta}_i\cdot\mu^{\epsilon}_i. \tag{14}$$

Formulation (12a)–(12c) has the advantage of involving a number of variables much smaller than the initial formulation of problem $\widehat{\mathcal{P}}$. However, adding a priori all constraints (12c) into the formulation requires to explicitly enumerate, for each scenario $u\in\mathcal{U}$, all the extreme points belonging to the set $\mathcal{E}(u)$, which is numerically impractical.

The Benders decomposition approach thus seeks to overcome this difficulty by iteratively constructing a convex piece-wise under-approximation of each function $\widehat{\mathcal{Q}}(y^*, u)$. This under-approximation is obtained by iteratively adding a subset of constraints (12c) in the formulation.

More precisely, the Benders decomposition algorithm starts by solving a relaxation of formulation (12a)–(12c) in which all constraints (12c) have been removed. This relaxation is called the master problem. Once the corresponding first-stage decisions (y^*, Q^*) are made, the algorithm solves a sequence of single-scenario sub-problems $\widehat{SP}(y^*, u)$. For each scenario u, the dual solution of sub-problem $\widehat{SP}(y^*, u)$ is used to generate a constraint (12c). If this constraint is violated by (y^*, Q^*), it is added to the formulation of the master problem as an optimality cut in order to improve the current under-approximation of function $\widehat{\mathcal{Q}}(\cdot, u)$. The algorithm then proceeds with resolving the master problem using its improved formulation and generating new optimality cuts till no more violated optimality cuts can be found.

Running this Benders decomposition algorithm thus requires to repeatedly solve a large number of single-scenario dual sub-problems. Consequently, the time efficiency of the algorithm heavily depends on the computation time needed to solve each single-scenario dual sub-problem $\widehat{DSP}(y^*, u)$. This topic is addressed in the following subsection.

4.4. Polynomial-time algorithm to solve the second-stage sub-problem

We present in this subsection a polynomial-time algorithm with guaranteed worst-case cubic complexity to efficiently solve each single-scenario dual sub-problem $\widehat{DSP}(y^*, u)$. We start by rewriting problem $\widehat{DSP}(y^*, u)$ as an equivalent minimization problem:

$$\widehat{\mathcal{Q}}(y^*, u) = -\min\ \Big\{ -s^e\cdot\alpha_1 + \sum_{t\in\mathcal{T}}\Big[\, d_{u,t}\cdot\alpha_t + \sum_{n\in\mathcal{N}} C_{t,n}\cdot y^*_{t,n}\cdot\beta_{t,n} + S_t\cdot\eta_t\,\Big] + \sum_{i\in\widehat{\mathcal{I}}}\widehat{\delta}_i\cdot\mu_i\Big\} \tag{15a}$$

$$\text{s.t. } \alpha_{t+1} - \alpha_t + \eta_t \geq -h_t, \qquad \forall t\in\mathcal{T}\backslash\{1\} \tag{15b}$$

$$\eta_T - \alpha_T - \lambda \geq -h_T, \tag{15c}$$

$$\alpha_t + \beta_{t,n} \geq -p_{t,n}, \qquad \forall t\in\mathcal{T},\quad \forall n\in\mathcal{N} \tag{15d}$$

$$\alpha_t \geq -g_t, \qquad \forall t\in\mathcal{T} \tag{15e}$$

$$\lambda + \mu_i \geq -\widehat{\kappa}_i, \qquad \forall i\in\widehat{\mathcal{I}} \tag{15f}$$

$$\mu_i \geq 0, \qquad \forall i\in\widehat{\mathcal{I}} \tag{15g}$$

$$\eta_t \geq 0, \qquad \forall t\in\mathcal{T} \tag{15h}$$

$$\beta_{t,n} \geq 0, \qquad \forall t\in\mathcal{T},\quad \forall n\in\mathcal{N} \tag{15i}$$

In order to design a polynomial algorithm, we first highlight that this dual problem may be formulated as an optimization problem comprising a reduced number of variables and linear constraints but involving a piece-wise linear objective. In the following, we denote by $[\cdot]^+$ the operator $\max\{0, \cdot\}$.

Firstly, notice that variable μ_i, $i\in\widehat{\mathcal{I}}$, is only constrained by (15f) and (15g). Since its coefficient $\widehat{\delta}_i$ is positive in the objective function to be minimized, one has $\mu_i = [-\widehat{\kappa}_i - \lambda]^+$ in any optimal solution. The associated costs may thus be

written as a piece-wise linear function involving only variable λ:

$$c^{\mu}(\lambda) = \sum_{i \in \widehat{\mathcal{I}}} \widehat{\delta}_i \cdot \mu_i = \sum_{i \in \widehat{\mathcal{I}}} \widehat{\delta}_i \cdot [-\widehat{\kappa}_i - \lambda]^+.$$

Secondly, observe that for all $t \in \mathcal{T}$ and $n \in \mathcal{N}$, η_t and $\beta_{t,n}$ are non-negative variables depending only on constraints (15b)–(15c) and (15d) respectively. Hence, for $t \in \mathcal{T} \setminus \{T\}$, one has $\beta_{t,n} = [-\alpha_t - p_{t,n}]^+$ and $\eta_t = [\alpha_t - \alpha_{t+1} - h_t]^+$ by optimality assumption since coefficients S_t and $C_{t,n} \cdot y^*_{t,n}$ are positive in the objective function. The associated costs at the time-step $t \in \mathcal{T} \setminus \{T\}$ may thus be written as a piece-wise linear function involving only variables α_t and α_{t+1}:

$$\begin{aligned} c_t(\alpha_{t+1}, \alpha_t) &= d_{u,t} \cdot \alpha_t + \sum_{n \in \mathcal{N}} C_{t,n} \cdot y^*_{t,n} \cdot \beta_{t,n} + S_t \cdot \eta_t \\ &= d_{u,t} \cdot \alpha_t + \sum_{n \in \mathcal{N}} C_{t,n} \cdot y^*_{t,n} \cdot [-\alpha_t - p_{t,n}]^+ + S_t \cdot [\alpha_t - \alpha_{t+1} - h_t]^+. \end{aligned}$$

Using similar arguments, the optimal value of the variables $\beta_{T,n}$, $n \in \mathcal{N}$, and η_T at time-step T is given by $\beta_{T,n} = [-\alpha_T - p_{T,n}]^+$ and $\eta_T = [\alpha_T + \lambda - h_T]^+$. The associated costs at time-step T may then be written as a piece-wise linear function involving only variables α_T and λ:

$$\begin{aligned} c_T(\lambda, \alpha_T) &= d_{u,T} \cdot \alpha_T + \sum_{n \in \mathcal{N}} C_{T,n} \cdot y^*_{T,n} \cdot \beta_{T,n} + S_T \cdot \eta_T \\ &= d_{u,T} \cdot \alpha_T + \sum_{n \in \mathcal{N}} C_{T,n} \cdot y^*_{T,n} \cdot [-\alpha_T - p_{T,n}]^+ + S_T \cdot [\alpha_T + \lambda - h_T]^+. \end{aligned}$$

Finally, at time-step $t = 1$, an additional linear cost is represented by c^e such that $c^e(\alpha_1) = -s^e \cdot \alpha_1$.

Problem $\widehat{DSP}(y^*, u)$ can then be reformulated using only $T + 1$ variables, *i.e.*, variables α_t, $t \in \mathcal{T}$, and variable λ, and a set of T linear constraints defining lower-bounds on variables α_t, $t \in \mathcal{T}$. This reformulation involves a piece-wise linear objective function and is given by:

$$\widehat{\mathcal{Q}}(y^*, u) = -\min \left\{ c^e(\alpha_1) + \sum_{t \in \mathcal{T} \setminus \{T\}} c_t(\alpha_{t+1}, \alpha_t) + c_T(\lambda, \alpha_T) + c^{\mu}(\lambda) \right\} \tag{16a}$$

$$\text{s.t. } \alpha_t \geq -g_t, \qquad \forall t \in \mathcal{T} \tag{16b}$$

Since the objective function (16a) is piece-wise linear, its minimum value is found at one of its break-points. Hence, one may solve problem $\widehat{DSP}(y^*, u)$ by evaluating its objective function at each of these break-points and by selecting the one providing the best value.

We thus now focus on enumerating these break-points. All are generated by the use of $[\cdot]^+$ operators in the objective function as well as by the lower bound on variables α. Let Λ be the finite set of break-point values for variable λ. Let $\tilde{\mathcal{A}}_t$ be the finite set of break-point values for variable α_t, $t \in \mathcal{T}$, including break-points generated by $[\cdot]^+$ operators in the objective and below the lower bound $-g_t$ defined in constraint (16b). These sets are defined according to the following observations:

- For all $t \in \mathcal{T}$, the only constraint on variable α_t is its lower bound defined by constraint (16b). This lower bound thus defines a break-point $-g_t \in \tilde{\mathcal{A}}_t$.
- For all $t \in \mathcal{T}$ and $n \in \mathcal{N}$, the non-linear term $[-\alpha_t - p_{t,n}]^+$ generates a break-point $-p_{t,n} \in \tilde{\mathcal{A}}_t$ for variable α_t.
- For all $t \in \mathcal{T} \setminus \{T\}$, the non-linear term $[\alpha_t - \alpha_{t+1} - h_t]^+$ generates break-points depending on both variables α_t and α_{t+1}. It follows that for any break-point $a_{t+1} \in \tilde{\mathcal{A}}_{t+1}$ for variable α_{t+1}, there exists a break-point $a_{t+1} + h_t \in \tilde{\mathcal{A}}_t$ for variable α_t. Conversely, for any break-point $a_t \in \tilde{\mathcal{A}}_t$ for variable α_t, there exists a break-point $a_t - h_t \in \tilde{\mathcal{A}}_{t+1}$ for variable α_{t+1}. However, this does not lead to an exponential explosion in the number of break-points in $\tilde{\mathcal{A}}_t$. Namely, the break-point $a_t \in \tilde{\mathcal{A}}_t$ generates the break-point $a_t - h_t \in \tilde{\mathcal{A}}_{t+1}$, which itself regenerates the break-point $a_t - h_t + h_t = a_t \in \tilde{\mathcal{A}}_t$.
- For $t = T$, the non-linear term $[\alpha_T + \lambda - h_T]^+$ generates break-points depending on both α_T and λ. It follows that for any break-point $b \in \Lambda$ for variable λ, there exists a break-point $-b + h_T \in \tilde{\mathcal{A}}_T$ for variable α_T.

These observations enable the definition of the finite set of possible break-points for variable α_t, $t \in \mathcal{T}$, $\tilde{\mathcal{A}}_t = \mathcal{P}_t \cup \mathcal{G}_t \cup \Lambda_t$, with

$$\mathcal{P}_t = \left\{ -p_{t',n} - \sum_{t''=t'}^{t-1} h_{t''} + \sum_{t''=t}^{t'-1} h_{t''} : t' \in \mathcal{T} \text{ and } n \in \mathcal{N} \right\},$$

$$\mathcal{G}_t = \left\{ -g_{t'} - \sum_{t''=t'}^{t-1} h_{t''} + \sum_{t''=t}^{t'-1} h_{t''} : t' \in \mathcal{T} \right\},$$

$$\Lambda_t = \left\{-b + \sum_{t'=t}^{T} h_{t'} : b \in \Lambda\right\}.$$

Finally, remember that constraint (16b) imposes a lower bound $-g_t$ on variable α_t, for all time-steps t. Hence, it is not possible to consider any break-point below that bound. Let us denote by $\mathcal{A}_t$ the finite set of feasible break-points for variable α_t: it is defined by $\mathcal{A}_t = \{\max\{-g_t, a\} : a \in \tilde{\mathcal{A}}_t\}$.

The finite set Λ of feasible break-points for variable λ is defined according to the following observations:

- For all $i \in \widehat{\mathcal{I}}$, variable λ admits a break-point $-\widehat{\kappa}_i \in \Lambda$ because of the non-linear term $[-\widehat{\kappa}_i - \lambda]^+$ in the objective function.
- For $t = T$, the non-linear term $[\alpha_T + \lambda - h_T]^+$ generates break-points depending on both α_T and λ. It follows that for any break-point $a_T \in \tilde{\mathcal{A}}_T$ for variable α_T, there exists a break-point $-a_T + h_T \in \Lambda$ for variable λ.

Hence, Λ may be defined as

$$\Lambda = \{-\widehat{\kappa}_i : i \in \widehat{\mathcal{I}}\} \cup \{-a + h_T : a \in \mathcal{P}_T\} \cup \{-a + h_T : a \in \mathcal{G}_T\} \cup \{-a + h_T : a \in \Lambda_T\}.$$

Notice that $\{-a + h_T : a \in \Lambda_T\} = \{b - h_T + h_T : b \in \Lambda\} = \Lambda$. We deduce that

$$\Lambda = \{-\widehat{\kappa}_i : i \in \widehat{\mathcal{I}}\} \cup \{-a + h_T : a \in \mathcal{P}_T\} \cup \{-a + h_T : a \in \mathcal{G}_T\}.$$

Let $\mathcal{A}$ denotes the cartesian product of all sets $\mathcal{A}_t$, $t \in \mathcal{T}$. Problem $\widehat{DSP}(y^*, u)$ is equivalent to

$$\widehat{\mathcal{Q}}(y^*, u) = -\min_{\substack{\alpha \in \mathcal{A} \\ \lambda \in \Lambda}} \left\{ c^e(\alpha_1) + \sum_{t \in \mathcal{T}\setminus\{T\}} c_t(\alpha_{t+1}, \alpha_t) + c_T(\lambda, \alpha_T) + c^\mu(\lambda) \right\}. \tag{17}$$

Note that the size of each break-point set $\mathcal{A}_t$ and Λ is linear in the number of time-steps T and linear pieces $\widehat{I}$. Hence, problem (17) becomes exponentially difficult to solve, since it requires that all possible combinations of break-points of all sets be enumerated.

Nevertheless, problem (17) can be equivalently reformulated under the following recursive form:

$$\widehat{\mathcal{Q}}(y^*, u) = -\min_{\alpha_1 \in \mathcal{A}_1} \left\{ c^e(\alpha_1) + v_1(\alpha_1) \right\}, \tag{18}$$

where $v_t(\cdot)$ represents the cost-to-go function at the time-step $t \in \mathcal{T}$ and is defined over $\mathbb{R}$ by:

$$v_T(a) = \min_{\lambda \in \Lambda} \left\{ c_T(\lambda, a) + c^\mu(\lambda) \right\}, \tag{19}$$

$$v_t(a) = \min_{\alpha_{t+1} \in \mathcal{A}_{t+1}} \left\{ c_t(\alpha_{t+1}, a) + v_{t+1}(\alpha_{t+1}) \right\}, \qquad \forall t \in \mathcal{T}\setminus\{T\} \tag{20}$$

Problem $\widehat{DSP}(y^*, u)$ can thus be solved dynamically using a backward recursion. First, one may solve problem (19) for all possible break-point values $a \in \mathcal{A}_T$, by enumerating in each case all possible break-point values $\lambda \in \Lambda$. Then, the values of function $v_T(\cdot)$ for all break-points in $\mathcal{A}_T$ are known, and it is possible to proceed in the recursion by solving problem (20) for all possible break-point values $a \in \mathcal{A}_{T-1}$ by the enumeration of all possible break-point values in $\mathcal{A}_T$. Hence, we proceed backward by enumerating the break-point values for variables α from the last time-step T to the first one.

The set Λ as well as each set $\mathcal{A}_t$ for $t \in \mathcal{T}$ are of worst-case size $\mathcal{O}(N \cdot T + \widehat{I})$. To solve problem (17) by means of a dynamic recursion over problems (18)–(20) requires the double enumeration of two break-point sets at each time-step. Therefore the following proposition holds.

Proposition 1. *Consider the dual problem (15a)–(15i) with T time-steps, N production units and I linear pieces. It is dynamically solved in $\mathcal{O}\big(T \cdot (N \cdot T + I)^2\big)$ time complexity.*

Retrieving the optimal values α_t^*, $t \in \mathcal{T}$, and λ^* is straightforward. For all $t \in \mathcal{T}$, $i \in \widehat{\mathcal{I}}$ and $n \in \mathcal{N}$, the optimal values of μ, β, and η are obtained according to previous observations such that $\mu_i^* = [-\widehat{\kappa}_i - \lambda^*]^+$, $\beta_{t,n}^* = [-\alpha_t^* - p_{t,n}]^+$, and $\eta_t^* = [\alpha_t^* - \alpha_{t+1}^* - h_t]^+$ if $t < T$, otherwise $\eta_T^* = [\alpha_T^* + \lambda^* - h_T]^+$.

We conclude that the single-scenario second-stage problem can be solved in cubic complexity.

5. Numerical experiments

We introduced in the previous sections two approximate stochastic dynamic algorithms for solving problem MS-LS-C-L-SUB. The first one relies on the piece-wise linear approximation of the expected cost-to-go functions presented in Section 3, its precision thus directly depends on the number of break-points used to approximate these functions at each decision stage. The second one corresponds to a variant of the first algorithm proposed in Section 4. In this variant, the

two-stage stochastic integer sub-problems found at each iteration of the algorithm are heuristically solved by a Benders-based decomposition scheme: this might reduce the computation time of the algorithm but at the cost of a potential deterioration of the solution quality.

We provide in this section the results of computational experiments carried out on randomly generated instances of the problem. The main objective of these experiments is to assess the effectiveness of our two algorithms by comparing it with the one of a naive deterministic optimization model without knowledge on the demand uncertainty. In what follows, we first introduce the setting used to randomly generate instances. We then present in more detail the experimental setup used for our numerical experiments. Finally, we report the outcome of the receding horizon simulations carried out on large-size instances. Overall, these results show the practical benefit, in terms of the quality of the solution, of using the proposed solution approaches rather than a simpler deterministic production planning model.

5.1. Instances generation

In our experiments, we use randomly generated instances of various sizes. The size of an instance is broadly proportional to the number T of time-steps in the planning horizon, the number K of decision stages, the number N of production units and the number U of scenarios used to represent the possible demand realizations at each decision stage. Note that for the sake of simplicity, we consider that each decision stage $k \in \mathcal{K}$ is of identical length $\frac{T}{K}$. Furthermore, we assume that the number of scenarios used at each stage $k \in \mathcal{K}$ is the same: $|\mathcal{U}_k| = U = R^{\frac{T}{K}}$, where R denotes the number of discrete realizations considered for the demand in a given time-step. We generated instances using the combinations of the following parameters: $(T, K, N, U) \in \{(36, 12, 1, 27), (36, 12, 4, 27), (25, 5, 1, 1024)\}$.

For each considered instance size, we generate several sets of instances differing from each other with respect to some key indicators. These ones depend on the average demand D, the average setup cost Q, the average production cost P, the average holding cost H and the average lost sales cost G. They are defined as follows:

- The Cost to Produce ratio (CP) represents the setup-to-production ratio: $CP = \frac{Q}{C \cdot P}$. It corresponds to the minimum value of the ratio between the setup costs and the linear production costs in a period in which production occurs. In our experiments, we consider values of CP in $\{1, 10\}$.
- The Economic Order Quantity (EOQ) represents the cost-per-order ratio: $EOQ = \sqrt{\frac{2 \cdot D \cdot Q}{H}}$. EOQ can be seen as a measure of the benefits to produce ahead of time. A low value of EOQ namely corresponds to situations in which anticipating the production of a demand several time-steps in advance is penalized by high inventory holding costs and is thus not profitable. In contrast, a high value of EOQ corresponds to situations where the setup costs are high as compared to the inventory holding costs so that using large lot sizes, thereby anticipating the production of a demand several time-steps in advance, can be a good production planning policy. In our experiments, we consider values of EOQ in $\{4, 14\}$.
- The Lost Sales Penalty ratio (LSP) represents the losses induced by an unmet demand as a percentage of the costs to meet that demand: $LSP = \frac{G}{C \cdot P + Q}$. In our experiments, we consider values of LSP in $\{1.25, 2\}$.

A set of instances is thus defined by a combination of the parameters (T, K, N, U) broadly determining its size and by a combination of the key indicators (CP, EOQ, LSP). Each considered set comprises 25 instances, each one being randomly generated as follows:

- The value of the average demand D is arbitrarily set to 1.
- The demand uncertainty follows a discrete uniform distribution law $DU\left(\frac{3 \cdot D}{4}, \frac{5 \cdot D}{4}\right)$ over a finite set of R equi-probable outcomes. The set $\mathcal{U}_k$ of uncertainty scenarios corresponds to the collection of all possible combinations of uncertainty realizations at the stage $k \in \mathcal{K}$.
- The production capacity is assumed to be time-invariant and machine-invariant. It is defined such that the utilization rate is equal to 80%, *i.e.*, $C_{t,n} = C = 1.25 \cdot \frac{D}{N}$, for each production unit $n \in \mathcal{N}$ and each time-step $t \in \mathcal{T}$.
- The storage capacity is also assumed to be time-invariant. It is defined such that the utilization rate is equal to 20%, *i.e.*, $S_t = S = 0.2 \cdot C$ for each time-step $t \in \mathcal{T}$.
- The average setup cost Q is arbitrarily set to 1. The value of the average production cost P and average inventory holding cost H are then deduced from the value of the indicators CP and EOQ for this set of instance.
- The setup and production costs are time- and machine-dependent. The production (resp. setup) cost on the unit $n \in \mathcal{N}$ at the time-step $t \in \mathcal{T}$, $p_{t,n}$ (resp. $q_{t,n}$), is randomly sampled from the uniform distribution $U\left(\frac{P}{2 \cdot N}, \frac{3 \cdot P}{2 \cdot N}\right)$ (resp. $U\left(\frac{Q}{2 \cdot N}, \frac{3 \cdot Q}{2 \cdot N}\right)$).
- The inventory holding cost is also time-varying. The inventory holding cost at time-step $t \in \mathcal{T}$, h_t, is randomly sampled from the uniform distribution $U\left(\frac{H}{2}, \frac{3 \cdot H}{2}\right)$.
- The lost sales cost is assumed to be time-invariant. Its value is set according to the one of the lost sales penalty indicator for the current set of instances, *i.e.*, $g_t = LSP \cdot (C \cdot P + Q)$ for each time-step $t \in \mathcal{T}$.

All test instances are made available in an open-access repository.[1]

[1] https://zenodo.org/records/15772804.

5.2. Experimental setup

In our numerical experiments, we aim at assessing the practical benefit of using the models and solutions approaches discussed in Sections 2–4 for lot-sizing under uncertain demand in a Decision–Hazard–Decision setting. To this end, we propose to use a receding-horizon framework to simulate as realistically as possible the multi-stage decision process involving a sequence of decision making and uncertainty realizations. This receding-horizon simulation is implemented for a given initial number of stages such that decisions are made stage after stage from the first one to the last one, without further extending the horizon as the simulation progresses. An example of this simulation setting can be found, for example, in Sereshti et al. [48], where the authors use this framework to assess two-stage stochastic lot-sizing models.

Each experiment consists in simulating the application, over all stages, of the first-stage planning decisions obtained by using one of the considered production planning approaches, and in recording the total "true cost" incurred when applying these first-stage decisions in a stochastic environment. Note that the "true cost" considered in this simulation is not the objective function of the optimization model, but the sum over time of the actual cost incurred by the application of the first-stage planning decisions over a "true scenario".

More precisely, the simulation procedure can be described as follows:

1. We first generate a true scenario representing the actual realization of the demand over the whole simulation horizon comprising K stages. This true scenario is randomly sampled from the same discrete uniform distribution law as the one used to generate the scenario sets $\mathcal{U}_k$, $k \in \mathcal{K}$.
2. We set the value of the "true cost" TC to 0, the value of the current stage k to 1, and the value of the entering stock σ^0 to s^e.
3. We then use one of the production planning approaches described below to build a production plan over a horizon involving stages from k to K and taking into account the entering inventory σ^{k-1}. This production plan determines in particular the setup value of each production unit for each time-step of stage k.
4. The demand for all time-steps belonging to stage k realizes and is assumed to be equal to that of the true scenario built at Step 1.
5. Based on the setup decisions determined at Step 3 and on the demand realizations uncovered at Step 4, a sub-problem is solved to identify the best recourse actions to be made for stage k.
6. The total actual cost for stage k (comprising the costs of the setup decided at Step 3 and the cost of the recourse actions taken at Step 5) is recorded and added to the value of the true cost TC. The actual ending inventory at stage k is recorded in σ^k.
7. If $k = K$, we stop. Otherwise, we increase k by one and go back to Step 3.

Regarding the production planning approaches used at Step 3 of the simulation procedure, several alternatives are considered:

- A perfect-information deterministic production planning model in which the demand to be satisfied is equal to the one of the true scenario built at Step 1. This corresponds to the very optimistic situation where we would have a perfect information on the demand over the whole horizon. This model takes the form of the small-size MILP (1a)–(1g), which is solved directly by a mathematical programming solver at each iteration of the simulation procedure. The obtained value of the true cost, denoted by V_{PI}, corresponds to the lowest achievable production cost and will be used as a reference to compare the performance of the other studied production planning approaches.
- A naive deterministic production planning model in which the demand to be satisfied in each time-step is equal to the average demand D. This corresponds to the situation where we neither know the actual demand in advance nor leverage the information of the demand uncertainty distribution. This model takes the form of the small-size MILP (1a)–(1g), which is solved directly by a mathematical programming solver at each iteration of the simulation procedure. The obtained value of the true cost is denoted by V_N, and the avoidable overcost related to the uncertainty is expressed as a percentage of the lowest achievable production cost by means of a gap GAP_N computed such that $GAP_N = (V_N - V_{PI})/V_{PI}$.
- A multi-stage stochastic programming model corresponding to the one presented in Section 2. This matches the situation where the knowledge of the distribution probability of the stochastic demand is explicitly exploited during the production planning process. At iteration k of the simulation procedure, this model involves $K + 1 - k$ decision stages, and uses a discrete set of U scenarios at each stage to model the possible realizations of the demand. This model is solved by the approximate stochastic dynamic programming algorithm described in Section 3. It is designated as the *approximate dynamic method* in the remainder of this section. The obtained value of the true cost is denoted by V_{AD}, and the avoidable overcost related to the uncertainty is expressed as a percentage of the lowest achievable production cost by means of a gap GAP_{AD} computed such that $GAP_{AD} = (V_{AD} - V_{PI})/V_{PI}$.
- The same multi-stage stochastic programming model also solved by the approximate stochastic programming algorithm described in Section 3. But we now use the Benders decomposition scheme introduced in Section 4 to heuristically solve the two-stage stochastic integer programs encountered at each iteration of this algorithm. It is designated as the *approximate dynamic decomposition method* in the remainder of this section. The obtained value of the true cost is denoted by V_{ADD}, and the avoidable overcost related to the uncertainty is expressed as a percentage of the lowest achievable production cost by means of a gap GAP_{ADD} computed such that $GAP_{ADD} = (V_{ADD} - V_{PI})/V_{PI}$.

All the solution methods are implemented in Python, using the Pyomo library to model the mathematical programs and CPLEX.12.10 to solve them. For the sake of simplicity, the approximate stochastic programming algorithm is implemented using the same number of break-points to build the piece-wise linear approximation of the expected cost-to-go function at each stage, *i.e.*, $I_k = I$, for each stage $k \in \mathcal{K}$. We consider values of I in $\{1, 10, 25\}$. Furthermore, note that the polynomial-time algorithm, used to solve the second-stage sub-problems, involved in the Benders decomposition scheme (see Section 4.4), is implemented in C++ with parallelization on the enumeration of all elements of the break-point sets.

For each instance set, we report the following performance measures:

- The average value, over the corresponding 25 instances, of the gaps GAP_N, GAP_{AD} and GAP_{ADD} between the true cost obtained with the corresponding approaches and the true cost obtained with the perfect-information approach.
- The average computing time in seconds, over the corresponding 25 instances, needed to solve the multi-stage stochastic programming model at the first iteration of the simulation procedure, using the approximate stochastic programming algorithm with and without decomposition. This computation time is namely representative of the overall time required to carry out Step 3 at each iteration of the receding-horizon simulation. Namely, we denote by CT_{AD} the average computing time of the dynamic approach without decomposition, while CT_{ADD-PA} and CT_{ADD-LP} represents the average computing time of the dynamic approach with decomposition, respectively using the polynomial algorithm developed in Section 4.4 and a general-purpose linear programming solver to address each single-scenario second-stage sub-problem.

5.3. Results

Instances with a large number of decision stages and a small number of scenarios at each stage. As a first step, we focus on the results obtained for the 16 instance sets corresponding to a decision process involving a rather large number of short decision stages ($T = 36$ time-steps and $K = 12$ stages for $T/K = 3$ time-steps per stage) and a small number of scenarios at each stage ($U = 27$).

Setup costs and production costs are equivalent for a parameter CP of value close to one while setup costs are much more important than production costs for greater values, leading to a production prone to operate at full capacity. A higher value of parameter EOQ improves the potential benefits to produce in advance and to store items. Production is much more tolerant to penalties when the parameter LSP is low, while solutions could otherwise be overly conservative. Finally, a greater value of parameter I has a direct impact on the precision of the approximate cost-to-go function constructed at each stage, leading to a solution of potentially improved quality. Hence, different parameter values lead to different optimal production plans. We consider the following combination of parameters to account for various categories of instances: $CP \in \{1, 10\}$, $EOQ \in \{4, 14\}$, $LSP \in \{1.25, 2\}$, and $I \in \{1, 10, 25\}$.

The results are displayed in Table 1 for the case with $N = 1$ production unit and in Table 2 for the case with $N = 4$ production units.

These results first show that using a piece-wise linear approximation of the expected cost-to-go function involving $I \in \{10, 25\}$ break-points instead of a simpler linear approximation corresponding to $I = 1$ has a strong positive impact on the quality of the obtained production plan. Namely, the average value of GAP_{AD} over the 16 considered instance sets is significantly reduced from 3.9% when $I = 1$ to 1.9% when $I = 10$ and to 1.8% when $I = 25$. Furthermore, increasing I from 10 to 25 leads to a reasonable increase of the average computation time from 25.3 s to 99.7 s.

Let us now compare the performance of the proposed approximate stochastic dynamic approach (*i.e.*, the AD approach) implemented with $I = 25$ break-points with the one of the naive deterministic approach. We observe that the average value of GAP_N, over the 16 considered instance sets, is equal to 3.0% while the one of GAP_{AD} is equal to 1.8%. This means that by using the AD approach instead of the naive one, we are able to divide the overcost of the production plan caused by the demand stochasticity by an average factor of nearly 2. Furthermore, we note that this overcost reduction is particularly important when the value of the indicators CP, EOQ and LSP is high. Namely, for the two instance sets corresponding to $(CP, EOQ, LSP) = (10, 14, 2)$, we are able to divide the average overcost by a factor of 3 (from $GAP_N = 4.5\%$ to $GAP_{AD} = 1.5\%$). Using the proposed AD approach to plan production is thus particularly beneficial if the production of a demand needs to be anticipated one or several time-steps in advance because the setup costs are high, in situations where generating storage does not represent a large risk since holding costs are marginal, and when a poor anticipation of the demand satisfaction leads to expensive lost sales due to production capacity limitations.

Regarding the approximate dynamic decomposition (ADD) approach, on these sets of instances, it provides an overall performance similar to the one of the AD approach. We note that $GAP_{ADD} = GAP_{AD}$ when $I = 1$, which is expected as the cost-to-go function used by the AD approach is linear in this case and thus equal to its convex approximation built by the ADD approach. Furthermore, when I is increased to 10 or 25, the gaps obtained by the AD and ADD approaches are close to each other. This indicates that the convex approximation of the cost-to-go function carried out in Section 4.2 does not negatively impact the practical quality of the obtained production plan.

Finally, the average computing time reported by the approximate stochastic dynamic (AD) approach and the approximate dynamic decomposition with polynomial algorithm (ADD-PA) approach are very similar. Note however that the approximate dynamic decomposition approach with standard LP solver (ADD-LP) has a far worse average computing time, with it being ten times slower than the previous two approaches on average. The Benders decomposition approach

Table 1
Simulation results for the instances with $T = 36$, $K = 12$, $N = 1$ and $U = 27$.

CP	EOQ	LSP	I	GAP_N (%)	GAP_{AD} (%)	GAP_{ADD} (%)	CT_{AD} (s)	CT_{ADD-PA} (s)	CT_{ADD-LP} (s)
10	14	1.25	1	1.04	2.39	2.39	1.6	4.7	77.0
			10		0.88	0.84	15.3	27.3	455.0
			25		0.85	0.84	73.8	72.4	1076.8
10	4	1.25	1	1.70	3.95	3.95	1.91	5.0	79.7
			10		1.34	1.40	16.6	26.0	440.5
			25		1.48	1.50	79.3	67.9	1024.0
10	14	2	1	3.23	2.65	2.65	1.6	4.2	70.4
			10		1.13	0.97	21.4	28.2	471.6
			25		1.06	0.96	77.3	74.3	1144.7
10	4	2	1	3.85	6.02	6.02	1.9	4.3	70.8
			10		3.01	2.91	19.3	25.8	412.7
			25		2.85	2.93	106.9	67.6	1012.1
1	14	1.25	1	0.6	1.96	1.96	1.5	4.3	80.4
			10		0.48	0.41	17.4	26.9	325.9
			25		0.48	0.44	64.1	71.8	728.4
1	4	1.25	1	1.15	2.66	2.66	2.0	5.0	76.7
			10		0.92	1.03	17.1	25.5	416.8
			25		0.88	1.02	84.4	66.4	993.9
1	14	2	1	1.83	2.71	2.71	1.6	4.1	67.5
			10		0.88	0.79	21.7	28.1	436.9
			25		0.78	0.79	76.2	74.1	1043.7
1	4	2	1	3.22	3.55	3.55	1.8	3.7	61.3
			10		1.73	1.78	19.8	26.3	404.0
			25		1.71	1.81	75.8	69.2	969.6

Table 2
Simulation results for the instances with $T = 36$, $K = 12$, $N = 4$ and $U = 27$.

CP	EOQ	LSP	I	GAP_N (%)	GAP_{AD} (%)	GAP_{ADD} (%)	CT_{AD} (s)	CT_{ADD-PA} (s)	CT_{ADD-LP} (s)
10	14	1.25	1	3.64	4.16	4.16	2.1	5.7	71.3
			10		1.48	1.34	36.5	40.6	468.8
			25		1.34	1.31	114.2	109.6	1104.2
10	4	1.25	1	3.84	5.59	5.59	2.3	6.9	84.1
			10		3.33	3.31	24.4	37.4	415.4
			25		3.12	3.15	127.3	102.2	985.9
10	14	2	1	5.67	4.32	4.32	2.1	6.1	78.4
			10		1.88	1.75	41.7	43.3	479.6
			25		1.67	1.69	131.8	111.4	1134.0
10	4	2	1	7.00	7.27	7.27	2.4	6.5	75.3
			10		5.75	5.83	28.8	35.8	403.0
			25		5.42	5.48	152.3	91.5	781.8
1	14	1.25	1	1.58	3.02	3.02	2.2	6.8	83.1
			10		0.90	0.94	34.2	48.6	505.5
			25		0.87	0.91	109.7	124.4	1195.0
1	4	1.25	1	2.36	3.62	3.62	2.2	7.0	86.6
			10		1.81	1.84	26.0	41.8	449.4
			25		1.80	1.85	94.9	105.6	1073.8
1	14	2	1	2.68	3.05	3.05	2.1	6.2	81.0
			10		1.62	1.43	36.4	45.6	519.1
			25		1.40	1.45	117.8	121.3	1214.5
1	4	2	1	4.55	4.75	4.75	2.3	7.0	56.7
			10		3.10	3.16	27.4	38.4	458.1
			25		3.01	3.26	109.5	105.4	1107.1

investigated in Section 4 thus seems to have a limited added value when the number of scenarios U involved in each two-stage stochastic integer sub-problems is small, as is the case here. Moreover in that case, the special-purpose polynomial algorithm designed in Section 4.4 appears to be more beneficial than a standard LP solver when solving each single-scenario second-stage sub-problem in the decomposition approach.

Instances with a small number of decision stages and a large number of scenarios at each stage. We now consider 2 instance sets corresponding to a decision process involving a rather small number of long decision stages ($T = 25$ time-steps and

Table 3
Simulation results for the instances with $T = 25$, $K = 5$, $N = 1$ and $U = 1024$.

CP	EOQ	LSP	I	GAP_N (%)	GAP_{AD} (%)	GAP_{ADD} (%)	CT_{AD} (s)	CT_{ADD-PA} (s)	CT_{ADD-LP} (s)
10	14	2	10	4.16	1.75	1.88	5183.6	282.3	7437.4
			25		Na	1.82	>6 h	720.1	17 377.9
1	14	2	10	1.99	1.37	1.40	4850.3	307.7	7878.2
			25		Na	1.40	>6 h	786.7	18 811.9

$K = 5$ decision stages with $T/K = 5$ time-steps per stage) and a large number of scenarios at each stage ($U = 1024$). The results are presented in Table 3. Note that we focus only on the case where the setup costs are high as compared to the inventory holding costs and the lost sales costs are high as compared to the production costs. As discussed above, this corresponds to the situations where it may be beneficial to produce the demand several time-steps in advance and to store it in-between two decision stages, *i.e.*, to situations where the expected cost-to-go functions will play an important role in the building of the production plan.

Results from Table 3 confirm the good performance of both the AD and ADD approaches in terms of solution quality as compared to the naive approach. We thus obtain an average gap of $GAP_{AD} = 1.56\%$ (resp. $GAP_{AD} = 1.64\%$) with the AD (resp. ADD) approach with $I = 10$, which is significantly smaller that the average gap of $GAP_N = 3.08\%$ obtained with the naive approach. However, we observe that the AD approach leads to prohibitively long computation times, especially when the number of break-points I is increased to 25. In contrast, the ADD approach with polynomial algorithm (ADD-PA) displays a much shorter average computation time, with it being at least ten times faster than the AD approach and the ADD approach with LP solver (ADD-LP).

We observe that the Benders decomposition scheme introduced in Section 4 is a key element to ensure the computational tractability of the approximate stochastic dynamic algorithm when the number of scenarios used at each stage is high and when the number of break-points used in the cost-to-go function approximation increases. Moreover, we notice that the Benders decomposition improves the average computation time only thanks to the special-purpose polynomial algorithm designed in Section 4.4.

To conclude, the computational results presented in this section show that the proposed approximate stochastic dynamic approaches are able to solve instances of realistic size, and to provide production plans of good quality within a manageable computing time. In particular, these approaches were shown to be efficient for instances with a large number of stages and scenarios.

6. Conclusion

This work addressed a variant of the single-item multi-resource lot-sizing problem under demand uncertainty. We investigated a multi-stage stochastic integer programming approach for this problem and considered a Decision–Hazard–Decision information in which, at each stage of the decision process, decisions must be made both before and after the actual demand realizes. We proposed an approximate stochastic dynamic programming approach to solve this problem. This approach relies on a backward recursion to build a piece-wise linear approximation of the expected cost-to-go function found at each stage, starting from the last stage and going back to the first one. Building this approximation requires to solve a large number of two-stage stochastic integer programs. In order to tackle instances involving a large number of scenarios at each stage, we proposed to use a convex under-approximation of the cost-to-go functions built by the approximate stochastic dynamic algorithm. This enables the design a Benders decomposition method to heuristically solve the two-stage stochastic integer programs encountered at each iteration of the algorithm. The numerical results of our simulations show the practical benefits of the proposed approach. In particular, we were able to divide, on average, the overcosts caused by the demand stochasticity by a factor of nearly 2 as compared to a naive deterministic optimization approach based on the expected demand. Furthermore, the computing time required to solve each instance remains compatible with an industrial use, and this even for realistic instances involving a large number of stages and scenarios per stage.

This work presents several interesting research perspectives. From a practical point of view, it could be worthwhile to extend this work to be able to consider several industrially relevant features such as demand correlation between decision stages, demand backlogging, or production of multiple end-items. All these extensions would require however to handle expected cost-to-go functions depending not on a single continuous variable as it the case here but on a set of continuous variables. This would complicate the construction of the piece-wise linear approximation of the expected cost-to-go function, however it would still be numerically tractable at least for small to medium-size instances. In particular, note that the requirement that the state variables should be bounded is not that strong as theoretical bounds can often be obtained from the analysis of the studied problem (*e.g.*, for capacitated lot-sizing problems, theoretical inventory bounds may be deduced from the bounds on production capacity). Furthermore, we could also consider a situation involving other stochastic input parameters for the production planning problem, such as the production capacity availability or the price and availability of the energy sources supplying the machines.

From a theoretical point of view, a possible perspective could be to establish precision bounds to measure the accuracy of the approximation of the expected cost-to-go function built at each iteration, and to more precisely control the quality of the solution provided by approximate stochastic dynamic programming approach. Such an approach was designed for particular cases (see, *e.g.*, Zéphyr et al. [56]). It however remains to be extended to stochastic integer programming problems under a Decision–Hazard–Decision setting such as the lot-sizing problem investigated here.

Finally, the polynomial time algorithm designed to solve the single-scenario second-stage sub-problem at each stage could be adapted for similar continuous production planning problems. Such is the case in the work of Spitzer et al. [50], of which the present study is an extension: a similar yet simpler problem specific to their use-case is solved using a polynomial time algorithm close to the one proposed here.

Data availability

Data is made available in a DOI repository indicated in the manuscript.

References

[1] Y. Adulyasak, J.-F. Cordeau, R. Jans, Benders decomposition for production routing under demand uncertainty, Oper. Res. 63 (4) (2015) 851–867.
[2] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, 1993.
[3] A. Akbalik, B. Penz, C. Rapine, Capacitated lot sizing problems with inventory bounds, Ann. Oper. Res. 229 (1) (2015) 1–18.
[4] D. Aksen, K. Altınkemer, S. Chand, The single-item lot-sizing problem with immediate lost sales, European J. Oper. Res. 147 (3) (2003) 558–566.
[5] D. Alem, E. Curcio, P. Amorim, B. Almada-Lobo, A computational study of the general lot-sizing and scheduling model under demand uncertainty via robust and stochastic approaches, Comput. Oper. Res. 90 (2018) 125–141.
[6] H.C. Bahl, S. Zionts, Multi-item scheduling by Benders' decomposition, J. Oper. Res. Soc. 38 (12) (1987) 1141–1148.
[7] J. Balasubramanian, I.E. Grossmann, Approximation to multistage stochastic optimization in multiperiod batch plant scheduling under demand uncertainty, Ind. Eng. Chem. Res. 43 (14) (2004) 3695–3713.
[8] M.T. Baumhof, E. Raheli, A.G. Johnsen, J. Kazempour, Optimization of hybrid power plants: When is a detailed electrolyzer model necessary? in: IEEE PowerTech 2023 Conference, Belgrade, Serbia, 2023.
[9] A. Ben-Tal, A. Goryashko, E. Guslitzer, A. Nemirovski, Adjustable robust solutions of uncertain linear programs, Math. Program. 99 (2) (2004) 351–376.
[10] E. Berk, A.O. Toy, O. Hazır, Single item lot-sizing problem for a warm/cold process with immediate lost sales, European J. Oper. Res. 187 (3) (2008) 1251–1267.
[11] D. Bertsimas, D.B. Brown, C. Caramanis, Theory and applications of robust optimization, SIAM Rev. 53 (3) (2011) 464–501.
[12] D. Bertsimas, A. Thiele, A robust optimization approach to inventory theory, Oper. Res. 54 (1) (2006) 150–168.
[13] V. Bindewald, F. Dunke, S. Nickel, Comparison of different approaches to multistage lot sizing with uncertain demand, Int. Trans. Oper. Res. 30 (6) (2023) 3771–3800.
[14] G.R. Bitran, H.H. Yanasse, Computational complexity of the capacitated lot size problem, Manag. Sci. 28 (10) (1982) 1174–1186.
[15] E.H. Bowman, Production scheduling by the transportation method of linear programming, Oper. Res. 4 (1) (1956) 100–103.
[16] N. Brahimi, N. Absi, S. Dauzère-Pérès, A. Nordli, Single-item dynamic lot-sizing problems: An updated survey, European J. Oper. Res. 263 (3) (2017) 838–863.
[17] P. Brandimarte, Multi-item capacitated lot-sizing with demand uncertainty, Int. J. Prod. Res. 44 (15) (2006) 2997–3022.
[18] P. Carpentier, J.-P. Chancelier, G. Cohen, M. De Lara, Stochastic Multi-Stage Optimization, in: Probability Theory and Stochastic Modelling, vol. 75, Springer Cham, 2015.
[19] P. Carpentier, J.-P. Chancelier, M. De Lara, T. Martin, T. Rigaut, Time blocks decomposition of multistage stochastic optimization problems, J. Convex Anal. 30 (2) (2023) 627–658.
[20] C. Cervellera, V.C.P. Chen, A. Wen, Optimization of a large-scale water reservoir network by stochastic dynamic programming with efficient state space discretization, European J. Oper. Res. 171 (3) (2006) 1139–1151.
[21] E. Curcio, P. Amorim, Q. Zhang, B. Almada-Lobo, Adaptation and approximate strategies for solving the lot-sizing and scheduling problem under multistage demand uncertainty, Int. J. Prod. Econ. 202 (2018) 81–96.
[22] N.P. Dellaert, M.T. Melo, Heuristic procedures for a stochastic lot-sizing problem in make-to-order manufacturing, Ann. Oper. Res. 59 (1) (1995) 227–258.
[23] M. Di Summa, L.A. Wolsey, Lot-sizing on a tree, Oper. Res. Lett. 36 (1) (2008) 7–13.
[24] S.S. Erenguc, Y. Aksoy, A branch and bound algorithm for a single item nonconvex dynamic lot sizing problem with capacity constraints, Comput. Oper. Res. 17 (2) (1990) 199–210.
[25] A. Gjelsvik, B. Mo, A. Haugstad, Long- and medium-term operations planning and stochastic modelling in hydro-dominated power systems based on stochastic dual dynamic programming, in: S. Rebennack, P.M. Pardalos, M.V.F. Pereira, N.A. Iliadis (Eds.), Handbook of Power Systems I, Energy Systems, Springer Berlin Heidelberg, 2010, pp. 33–55.
[26] M. Gruson, J.-F. Cordeau, R. Jans, Benders decomposition for a stochastic three-level lot sizing and replenishment problem with a distribution structure, European J. Oper. Res. 291 (1) (2021) 206–217.
[27] Z. Hu, G. Hu, A two-stage stochastic programming model for lot-sizing and scheduling under uncertainty, Int. J. Prod. Econ. 180 (2016) 198–207.
[28] Z. Hu, G. Hu, A multi-stage stochastic programming for lot-sizing and scheduling under demand uncertainty, Comput. Ind. Eng. 119 (2018) 157–166.
[29] R.A. Jarvis, On the identification of the convex hull of a finite set of points in the plane, Inform. Process. Lett. 2 (1) (1973) 18–21.
[30] E. Koca, G.I. Koksalan, Capacitated lot sizing problem with periodic carbon emission constraints and multiple resources, Int. J. Prod. Res. 61 (19) (2023) 6589–6601.
[31] C. Li, F. Liu, H. Cao, Q. Wang, A stochastic dynamic programming based model for uncertain production planning of re-manufacturing system, Int. J. Prod. Res. 47 (13) (2009) 3657–3668.
[32] X. Liu, Y.L. Tu, Capacitated production planning with outsourcing in an OKP company, Int. J. Prod. Res. 46 (20) (2008) 5781–5795.
[33] P.B. Macedo, D. Alem, M. Santos, M.L. Junior, A. Moreno, Hybrid manufacturing and remanufacturing lot-sizing problem with stochastic demand, return, and setup costs, Int. J. Adv. Manuf. Technol. 82 (5–8) (2016) 1241–1257.
[34] H.M. Markowitz, A.S. Manne, On the solution of discrete programming problems, Econometrica 25 (1) (1957) 84–110.

[35] C. Martinez Parra, M. de Lara, J.-P. Chancelier, P. Carpentier, J.-M. Janin, M. Ruiz, A two-timescale Decision-Hazard-Decision formulation for storage usage values calculation, 2025, arXiv:2408.17113v2.
[36] M.A. Naeem, D.J. Dias, R. Tibrewal, P.C. Chang, M.K. Tiwari, Production planning optimization for manufacturing and remanufacturing system in stochastic environment, J. Intell. Manuf. 24 (4) (2013) 717–728.
[37] Y. Pochet, L.A. Wolsey, Production Planning By Mixed Integer Programming, in: Springer Series in Operations Research and Financial Engineering, Springer New York, 2006.
[38] K. Postek, D. den Hertog, Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set, INFORMS J. Comput. 28 (3) (2016) 553–574.
[39] W.B. Powell, Approximate Dynamic Programming: Solving the Curses of Dimensionality, in: Wiley Series in Probability and Statistics, Wiley, 2011.
[40] F. Quezada, C. Gicquel, S. Kedad-Sidhoum, Combining polyhedral approaches and stochastic dual dynamic integer programming for solving the uncapacitated lot-sizing problem under uncertainty, INFORMS J. Comput. 34 (2) (2022) 1024–1041.
[41] F. Quezada, C. Gicquel, S. Kedad-Sidhoum, A stochastic dual dynamic integer programming based approach for remanufacturing planning under uncertainty, Int. J. Prod. Res. 61 (17) (2023) 5992–6012.
[42] F. Quezada, C. Gicquel, S. Kedad-Sidhoum, D.Q. Vu, A multi-stage stochastic integer programming approach for a multi-echelon lot-sizing problem with returns and lost sales, Comput. Oper. Res. 116 (2020) 104865.
[43] R. Rahmaniani, T.G. Crainic, M. Gendreau, W. Rei, The benders decomposition algorithm: A literature review, European J. Oper. Res. 259 (3) (2017) 801–817.
[44] C. Rapine, B. Penz, C. Gicquel, A. Akbalik, Capacity acquisition for the single-item lot sizing problem under energy constraints, Omega 81 (2018) 112–122.
[45] J.A. Riera, R.M. Lima, O.M. Knio, A review of hydrogen production and supply chain modeling and optimization, Int. J. Hydrog. Energy 48 (37) (2023) 13731–13755.
[46] R.A. Sandbothe, G.L. Thompson, Decision horizons for the capacitated lot size model with inventory bounds and stockouts, Comput. Oper. Res. 20 (5) (1993) 455–465.
[47] M. Schlenkrich, S.N. Parragh, Capacitated multi-item multi-echelon lot sizing with setup carry-over under uncertain demand, Int. J. Prod. Econ. 277 (2024) 109379.
[48] N. Sereshti, Y. Adulyasak, R. Jans, The value of aggregate service levels in stochastic lot sizing problems, Omega 102 (2021) 102335.
[49] A. Shapiro, Analysis of stochastic dual dynamic programming method, European J. Oper. Res. 209 (1) (2011) 63–72.
[50] V. Spitzer, C. Gicquel, E. Gurevsky, F. Sanson, Day-ahead lot-sizing under uncertainty: An application to green hydrogen production, in: A. Basu, A.R. Mahjoub, J.J. Salazar González (Eds.), Combinatorial Optimization, in: Lecture Notes in Computer Science, vol. 14594, Springer Cham, 2024, pp. 402–416.
[51] A. Street, D. Valladão, A. Lawson, A. Velloso, Assessing the cost of the Hazard-Decision simplification in multistage stochastic hydrothermal scheduling, Appl. Energy 280 (2020) 115939.
[52] H. Tempelmeier, Stochastic lot sizing problems, in: J.M. Smith, B. Tan (Eds.), Handbook of Stochastic Models and Analysis of Manufacturing System Operations, in: International Series in Operations Research & Management Science, vol. 192, Springer New York, 2013, pp. 313–344.
[53] F. Vanderbeck, L.A. Wolsey, Valid inequalities for the Lasdon-Terjung production model, J. Oper. Res. Soc. 43 (5) (1992) 435–441.
[54] A. Witthayapraphakorn, P. Charnsethikul, Benders decomposition with special purpose method for the sub problem in lot sizing problem under uncertain demand, Oper. Res. Perspect. 6 (2019) 100096.
[55] J. Xiao, Grid Integration and Smart Grid Implementation of Emerging Technologies in Electric Power Systems Through Approximate Dynamic Programming (Ph.D. thesis), Purdue University, 2013.
[56] L. Zéphyr, P. Lang, B.F. Lamond, P. Côté, Approximate stochastic dynamic programming for hydroelectric production planning, European J. Oper. Res. 262 (2) (2017) 586–601.
[57] W. Zhang, D. Nikovski, State-space approximate dynamic programming for stochastic unit commitment, in: 2011 North American Power Symposium, Boston, USA, 2011.
[58] J. Zou, S. Ahmed, X.A. Sun, Stochastic dual dynamic integer programming, Math. Program. 175 (1–2) (2019) 461–502.