ARTICLE IN PRESS
JID: OME [m5G;August 31, 2019;5:1]

Omega xxx (xxxx) xxx

Contents lists available at ScienceDirect

Omega

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

A structuring review on multi-stage optimization under uncertainty: Aligning concepts from theory and practice

Hannah Bakker*, Fabian Dunke, Stefan Nickel

Karlsruhe Institute of Technology, Institute for Operations Research, Discrete Optimization and Logistics, Kaiserstr 12, Karlsruhe, 76131 Germany

ARTICLE INFO

Article history:
Received 11 December 2018
Accepted 12 June 2019
Available online xxx

Keywords:
Multi-stage optimization
Uncertainty modelling
Sequential decision making

ABSTRACT

While methods for optimization under uncertainty have been studied intensely over the past decades, the explicit consideration of the interplay between uncertainty and time has gained increasing attention rather recently. Problems requiring a sequence of decisions in reaction to uncertainty realizations are of crucial relevance in real-world applications, e.g., supply chain planning, scheduling, or finance. Several methods emphasizing varying aspects of these problems have been developed, mainly triggered by a particular application. Although these methods all intend to solve a similar underlying problem, they differ strongly with respect to the uncertainty representation, the prescriptive solution information they provide and the means of performance evaluation. The result is a fragmented picture of uncertain multi-stage problems – both from a methodological and an application-oriented perspective. It fails to interconnect results from different disciplines or even comparing strengths and weaknesses of individual methods in particular applications. This review aims at integrating the different methods for solving uncertainty inflicted multi-stage optimization problems into a broader picture, thereby paving the way for more comprehensive approaches to sequential decision making under uncertainty. For this purpose, a description of the methods along with their historic development is given first. Secondly, an overview on their main areas of application is provided. We conclude that decoupling uncertainty models from solution methods and developing standardized performance measures represent key steps for organizing multi-stage optimization under uncertainty and for eliciting further potentials of yet unexplored combinations of uncertainty models and solution methods.

1. Introduction

Since the early beginnings of mathematical programming, uncertainty in optimization problems has been a topic of increasing interest. Triggered by Dantzig's seminal work "Linear programming under uncertainty" in 1955 [1], in which he introduced the concept of stochastic programming, numerous approaches have been developed, all addressing the phenomenon that real world applications often suffer from incomplete information on relevant input data – short: uncertainty. At the same time, real world optimization problems often appear in a temporal context. Therefore, the interplay between uncertainty and time is inherently important to any related decision making process. While multi-period formulations are already well-established, capturing the dynamics of successive information disclosure is a challenge that received increasing attention over the past several decades and still requires further exploration.

The problems referred to require a sequence of decisions which react to outcomes that evolve over time, and information on these outcomes is disclosed gradually [2,3]. In the following, these problems will be referred to as uncertain multi-stage optimization problems and their general outline is sketched in Fig. 1. Uncertain information is modelled by a sequence $\xi_{[T]} = \{\xi_t : t = 1, \dots, T\}$ of successively observable data vectors ξ_t over a planning horizon of T stages, with $T \in \mathbb{N}$. The time between two successive observations ξ_t and ξ_{t+1} of elements from $\xi_{[T]}$ marks a (decision-) stage. At each stage, a new (partial) decision x_t has to be irrevocably fixed based on the information available at this point.

While reviews exist on optimization under uncertainty [4,5] and application specific approaches (e.g., [6,7]), the authors feel that insufficient attention has been paid to methods focusing on problems with multi-stage decision structures. As depicted in Fig. 2 for uncertain multi-stage problems, solution methods have been developed in different research disciplines. While in the mathematical programming community established concepts of stochastic programming and robust optimization

* Corresponding author.
E-mail addresses: hannah.bakker@kit.edu (H. Bakker), fabian.dunke@kit.edu (F. Dunke), stefan.nickel@kit.edu (S. Nickel).

https://doi.org/10.1016/j.omega.2019.06.006

Please cite this article as: H. Bakker, F. Dunke and S. Nickel, A structuring review on multi-stage optimization under uncertainty: Aligning concepts from theory and practice, Omega, https://doi.org/10.1016/j.omega.2019.06.006

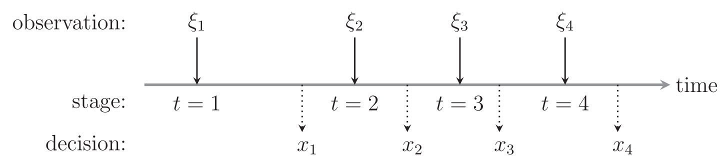

Fig. 1. Information disclosure in uncertain multi-stage optimization problems.

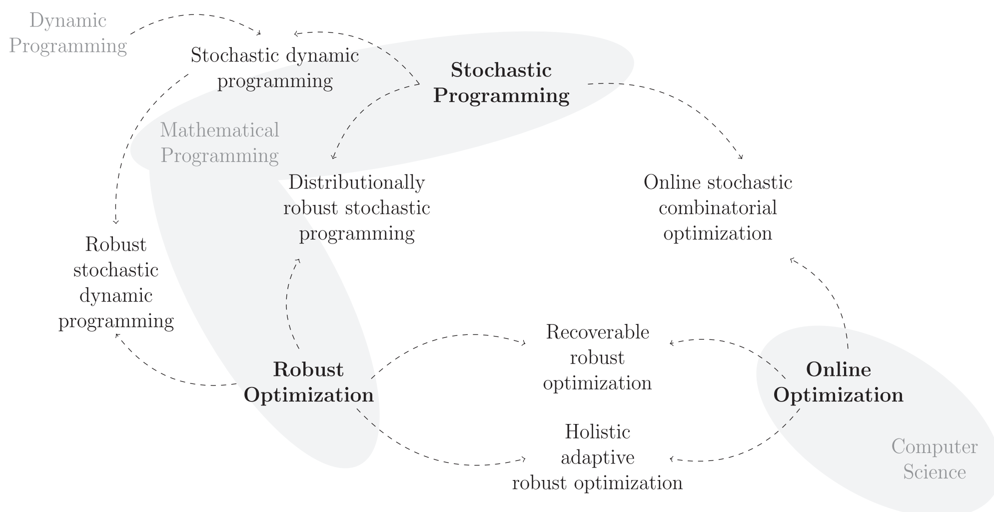

Fig. 2. Prominent methods for solving multi-stage stochastic programs have been derived from three basic concepts: stochastic programming, robust optimization, and online optimization, stemming from the fields of mathematical programming and computer science, respectively.

have been extended to multi-stage settings, the algorithm-based concept of online optimization evolved from the field of computer science which deals with sequential decision making by definition. Overviews are available for specific disciplines or specialized aspects dealing with optimization under uncertainty: [8] gives a survey on stochastic programming, Gabrel et al. [9] and Yanıkoğlu et al. [10] discuss robust optimization (with the latter reference focusing on the option of adjustable actions), Albers [11] provides a survey on online optimization, and [12] consider the use of Monte Carlo sampling in stochastic optimization. Furthermore, over the last two decades researchers have begun to combine ideas from the more established concepts leading to the emergence of new approaches such as online stochastic combinatorial optimization or recoverable robust optimization.

However, much of these developments took place in an application-driven context, and even though they all address the same basic problem of sequential decision making in the face of gradual information disclosure, the approaches differ largely in terms of formalism, uncertainty model and solution concept. These differences make a direct transfer from one concept to another difficult for any problem of a given application. Furthermore, the authors feel that when confronted with an uncertain multi-stage problem the choice of a concept is often based on personal preference or habit rather than suitability. This effect is amplified by the lack of adequate means to compare solutions between concepts as currently performance measures are only concept specific. From this background the authors believe that a systematic review of approaches to uncertain multi-stage optimization is overdue.

The outline of the paper is as follows: First we review relevant methods and concepts for solving uncertain multi-stage problems based on their conceptual approach, their formal model of uncertainty, and their historic development. Moreover, we give an overview on prominent performance measures of respective methods. We conclude this theory-related part by summarizing the main findings in terms of commonalities and differences of the methods. The second large part of the paper then gives an overview on applications with multi-stage character that were tackled by the different methods for uncertainty-inflicted optimization. We show that for each methodological approach there are typical application domains and typical time horizons that were treated preferentially. Finally, we draw a conclusion for the current state of multi-stage optimization under uncertainty and point out main directions for further research in this field.

2. Methods and concepts

Uncertainty in an optimization problem means that some or all of the problem's parameters are not known at the time the problem has to be solved. This implies that in an uncertain setting the deterministically defined concepts of feasibility and optimality of a solution are no longer defined as illustrated for the simple example of a linear program with two decision variables in Fig. 3. Consequently, the problem per se cannot be solved by state-of-the-art solvers. As will be highlighted in the following, approaches that evolved from mathematical programming largely focus on finding deterministic reformulations for these concepts.

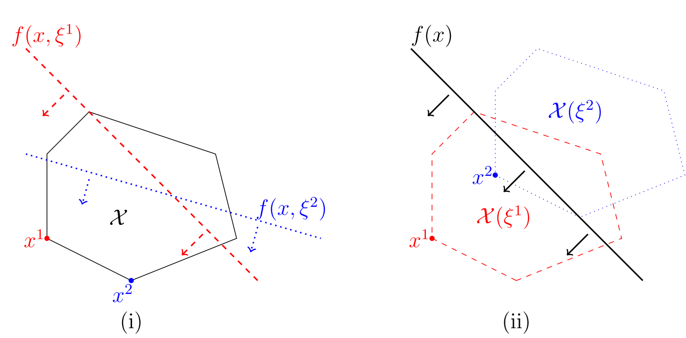

Fig. 3. (i) Optimality: while x^1 is optimal in case of ξ^1, solution x_2 would be optimal if ξ^2 was realized; (ii) Feasibility: while x^1 is optimal under feasible set $\mathcal{X}(\xi^1)$, it is infeasible when the set of feasible solutions realizes as $\mathcal{X}(\xi^2)$.

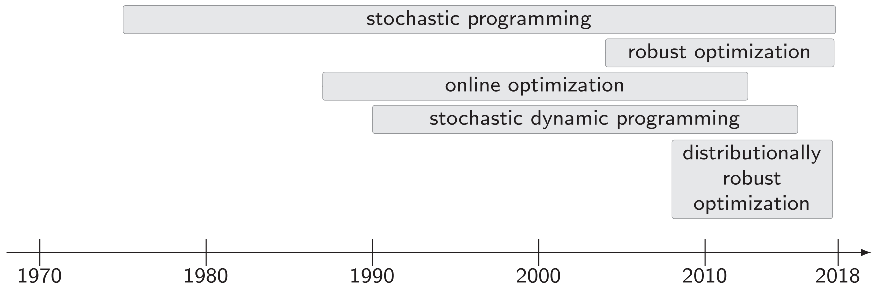

Fig. 4. Overview of flourishing research and application periods for the different methods with respect to the multi-stage setting.

Even though explicitly considering temporal relationships between information disclosure and (partial) decision making adds complexity to the problem, multi-stage models enable the decision maker to adapt decisions at later stages to the already observed realizations of the uncertain data. Thereby, multi-stage models yield the potential to lead to better solutions than their static counterparts which require that all decisions have to be fixed up front.

However, translating this advantage into an actual improvement remains a challenge, and in the following several approaches for this task will be reviewed. Each will be introduced by its general idea and then be embedded into its historical context. To this end, Fig. 4 provides a first guideline for a rough classification of the developments. This timeline will be further resolved in subsequent sections.

2.1. Stochastic programming

As a direct extension of deterministic mathematical programming, uncertainty is added to a problem by modelling some of the problem's parameters as a (multi-dimensional) random variable ξ following a probability distribution $\mathcal{F}$ which is assumed to be known to the decision maker. According to Rosenhead's classification of decision environments [13], stochastic programming depicts the situation of decision making under risk. Having its origin in Dantzig's seminal paper "Linear programming under uncertainty" [1], stochastic programming is, to our knowledge, the first approach from within the Operations Research community to deal with uncertainty in mathematical programming based optimization.

As mentioned earlier, if some or all the problem parameters are random, the concepts of optimality and feasibility need to be redefined. Regarding optimality, Dantzig proposed to comprise all the information on the probability of the uncertain parameter in a single, deterministic value, namely the expectation $\mathbb{E}_{\xi\sim\mathcal{F}}[\cdot]$, leading to a deterministic version of the objective function f:

$$\min_x \mathbb{E}_{\xi\sim\mathcal{F}}[f(x,\xi)]. \tag{1}$$

This reformulation assumes that the decision maker has a risk-neutral attitude as it implies that potential losses are equally offset by potential gains. However, particularly in applications with non-repetitive decisions decision makers often exhibit a risk averse attitude fearing losses more than cherishing gains. Therefore, alternative concepts of optimality which are often based on risk measures from finance, as e.g., the mean-variance criterion [14], the Value at Risk [15], or the Conditional Value at Risk [16], have been applied over the years.

Regarding feasibility, stochastic programming considered temporal relations between decisions and uncertainty observations early on by introducing the concept of recourse – a partial decision that is to be fixed after uncertainty has been disclosed so that feasibility is ensured, even if possibly at a high cost. This setting is known as two-stage stochastic programming and formalized by a nested problem formulation in which the first stage decision x_1 has to be taken prior to the observation of ξ and is solution to the problem

$$\min_{x_1\in\mathbb{R}^{n_1}} \; f_1(x_1) + \mathbb{E}_{\xi\sim\mathcal{F}}[Q(x_1,\xi)] \qquad \text{s.t. } x_1 \in \mathcal{X}_1 \tag{2}$$

where $Q(x_1, \xi)$ is the optimal value of the second-stage problem

$$Q(x_1,\xi) := \min_{x_2\in\mathbb{R}^{n_2}} f_2(x_2,\xi) \qquad \text{s.t. } x_2 \in \mathcal{X}_2(x_1,\xi) \tag{3}$$

and x_2 is the second stage decision. While the expectation of the optimal outcome of the second stage decision can be determined deterministically for a given x_1, the recourse actions x_2 depend on and have to be computed for every realization of the random outcome ξ. The resulting solution $(x_1, x_2(\xi))$ is called a policy.

The problem formulation from above can be extended to a multi-stage setting (cf. [17,18]) in order to capture the dynamics of real-world decision making. Thereby, the successive information disclosure is formally captured by extending the random variable to a stochastic process $\xi_{[T]} := \{\xi_t \mid t = 1,\dots,T\}$ where each observed element $\xi_t \sim \mathcal{F}_t$ of the process determines the parameters of the t-stage subproblem:

$$\min_{x_1\in\mathcal{X}_1} f_1(x_1) + \mathbb{E}_{\xi_2}\Bigg[\inf_{x_2\in\mathcal{X}_2(x_1,\xi_1)} f_2(x_2,\xi_2) + \mathbb{E}_{\xi_3}\Bigg[\cdots + \mathbb{E}_{\xi_T}\Bigg[\inf_{x_T\in\mathcal{X}_T(x_{T-1},\xi_T)} f_T(x_T,\xi_T)\Bigg]\Bigg]\Bigg]. \tag{4}$$

Modelling multi-stage uncertainty by means of a stochastic process is a generic description which – particularly in case of continuous distributions $\mathcal{F}_t$ – leads to practically unsolvable problems. Therefore, the model is often simplified to a so-called scenario tree. First introduced by [19], it can be thought of as a graphic representation of a discrete (or reduced and discretized) stochastic process. Starting at the root node each level of the tree corresponds to the possible outcomes at a stage of the problem. Hence, the paths $\xi^s_{[T]}$, $s = 1,\dots,S$, from root to leaves correspond to possible realizations of the (discretized) stochastic process, also referred to as the set of scenarios $\mathcal{S}$. The probability π^s of an individual scenario $\xi^s_{[T]}$ can be determined by multiplying the probabilities of the individual outcomes at each stage along the path in the tree. With this discrete form of uncertainty and the fact that the expectation of a discrete random variable equals a weighted sum, the multi-stage stochastic program in Eq. (4) can be transformed to what is called the deterministic counterpart which can be handed over to state-of-the-art solvers:

$$\min_{x_1,\{x_2^s\}_{s=1}^S,\dots,\{x_T^s\}_{s=1}^S} \sum_{s\in S} \pi_s f(x_1, x_2^s, \dots, x_T^s, \xi^s_{[T]}) \tag{5}$$

$$x_1 \in \mathcal{X}_1, x_t^s \in \mathcal{X}_t^s \qquad s \in S, t \in T \tag{6}$$

$$x_t^s = x_t^{s'} \qquad t \in T, s, s' \in S \text{ with } s \neq s', \xi_t^s = \xi_t^{s'}. \tag{7}$$

The resulting policy prescribes the optimal sequence of decisions for every scenario and hence gives stage-wise instructions of what

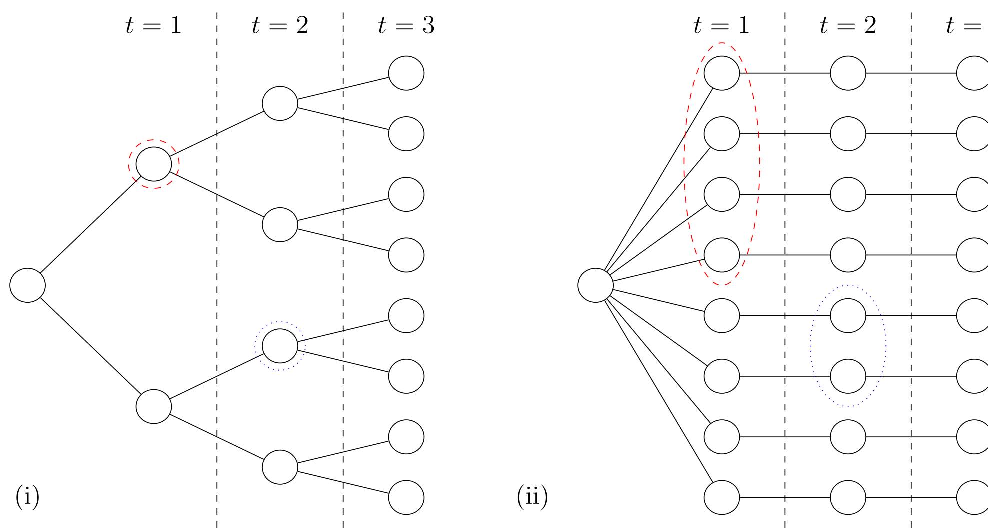

Fig. 5. (i) Scenario tree - each node represents the observation of a distinct uncertain data vector ξ_t^s; (ii) Fan of scenarios - each node corresponds to a decision vector x_t^s in the deterministic counterpart (Eq. (5)–(7)).

to do under each realization of the uncertain parameters (transient decision making). Fig. 5 illustrates that the deterministic counterpart ignores the information based on uncertainty evolution inherent to the scenario tree. Instead, each scenario is represented as an independent path in the tree leading to a “fan of scenarios”. This does not only lead to increased complexity as the number of decision variables increases but also requires the additional constraints in Eq. (7). These so-called non-anticipativity constraints ensure that decisions for individual scenarios do not differ before the associated scenarios can be distinguished from one another. Efficiently exploiting the scenario tree structure to reduce complexity is a topic which to our knowledge remains largely unexplored.

Albeit the extensive form of a multi-stage stochastic program may be handed over to a solver, practical applications are restrained by computational limits as a problem’s dimensions grow quickly in the number of stages and scenarios. This computational challenge is also reflected in the historic development of the field. Fig. 6 sketches the methodological developments in stochastic programming encompassing multi-stage problems.

The body of theory on multi-stage stochastic programming has been analyzed throughout the past decades in detail including topics of well-definedness [20,21], stability [22], approximations [23], complexity [24,25], and advanced algorithmic outlines [26,27]. However, researchers recognized the need for approaching stochastic programming models in applications with the aid of computer-supported decision making tools [28,29]) and that without further ado, the practical solvability of multi-stage stochastic programs is strongly delimited due to large dimensions ([30]).

Therefore, decomposition methods were proposed based on specific structures of stochastic programs. A comprehensive survey including a classification into primal and dual methods is given in [31]. Primal methods solve a series of subproblems to approximate the recourse costs with increasing accuracy. Dual methods rely on relaxing non-anticipativity constraints and considering them in the Lagrangian function which leads to one scenario corresponding to one subproblem. Moreover, numerous specialized versions have been developed: [32] introduces a parallel decomposition scheme where scenarios and their relation to each other (subscenarios for which non-anticipativity has to be ensured) are organized in a tree and the mathematical programming formulation is derived based upon the tree. Subproblems resulting from tree nodes allow for parallelized solving when taking into account information exchange along tree arcs. The information exchange also allows for cutting plane integration. Lulli and Sen [33] address multi-stage stochastic programs with integer variables by a branch and price approach where the original problem is decomposed into a master problem and scenario subproblems. The master problem can be obtained from the Dantzig-Wolfe decomposition resulting from the integer scenario polyhedron of the feasible set. Pricing subproblems are formulated for each scenario. A general method for generating cutting planes to be used in decomposition schemes is presented in [34] where several valid inequalities

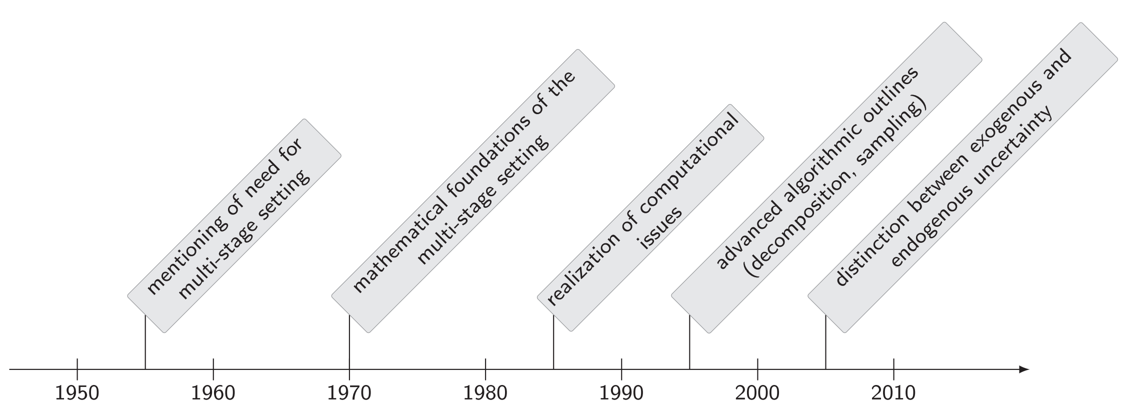

Fig. 6. Evolution of methods and concepts related to stochastic programming with regard to multi-stage settings.

are derived from the scenario tree. Decomposition schemes were also specifically tailored to problem settings. For instance, Singh et al. [35] exhibit how multi-stage stochastic mixed-integer programming can be applied to planning capacity expansion of production facilities and be solved using a Dantzig-Wolfe-based decomposition scheme which relies on a scenario tree representation and a technique called variable splitting which allows for a reformulation of the problem yielding stronger bounds from the master problem. The decomposition method of stochastic dual dynamic programming [26,27] relies on the generation of scenarios to approximate the recourse cost function in multi-stage settings using cutting planes. The advantage of stochastic dual dynamic programming lies in the fact that a problem need not be formulated in its entirety at the outset which makes it attractive for the multi-stage setting.

Besides decomposition, an attempt to make problem instances solvable in reasonable time and also to account for problems with infinitely many scenarios is by generating "significant" scenarios. To this end, the technique of scenario generation through scenario trees gained importance [19,36–38]. Likewise, general sampling outlines such as sample average approximation became popular also for multi-stage stochastic programming [39]. Lately, researchers have combined sampling and scenario generation [40].

Also in multi-stage models the use of risk-averse objective functions has gained prominence [41]. Thereby, concepts from the two-stage setting cannot be transferred ad hoc to multi-stage models as it is not evident how to evaluate recourse costs for the entire planning horizon. Opinions differ about whether to evaluate risk for the entire planning horizon, at every stage, or for individual scenarios. In this context the question of time-consistent risk measures, which give a persistent evaluation of risk across stages and scenarios, and a related definition of consistency arises [42,43]. Algorithmic implementations of risk averse multi-stage models can e.g. be found in [44,45]. Scenario-reduction methods and sampling approaches specified for risk-averse models are presented in [46]. Shapiro [47] provides a comprehensive outline on the approach of risk-averseness in multi-stage stochastic programming including a discussion of sample average approximation.

A distinction between exogenous uncertainty (as assumed in classical stochastic programming) and endogenous uncertainty is made in the research stream started by Goel and Grossmann [48]. In contrast to exogenous uncertainty where stochastic processes cannot be influenced (e.g., customer demands which cannot be influenced), endogenous uncertainty is tied to stochastic processes which depend on previous decisions (e.g., customer demands influenced by marketing campaigns). In case of endogenous uncertainty, two types of this uncertainty are described: First, decisions can influence parameter probability distributions (e.g., demands in case of marketing campaigns). Second, decisions can influence the timing of parameter realizations (e.g., when a previous decision has to be implemented first and it has to be waited to get knowledge about realized parameter values). Technically, a scenario tree describing the outcomes of all possible random processes becomes decision-dependent and requires to model all possible scenario trajectories. Translated into mathematical programming, disjunctive constraints reflecting logical and temporal relations between decision variables become necessary. Extending classical stochastic programming, the authors present a hybrid mixed-integer disjunctive programming formulation and advocate Lagrangean duality based branch and bound as a solution method. The coupling of the stochastic process with the optimization process is further considered in the multi-stage setting in [49] and [50]. To facilitate problem solving, conditional non-anticipativity constraints (as opposed to initial non-anticipativity constraints) are tackled by delimiting their number. Special emphasis is put on a temporal resolution of decision-dependent uncertainties by Tarhan et al. [51] who propose multi-stage stochastic disjunctive programming with dynamically generated non-anticipativity constraints that can be solved by combining global optimization and outer approximation. A comprehensive presentation of computational strategies for multi-stage stochastic programming under endogenous and exogenous uncertainties is given in [52].

2.2. Robust optimization

Like stochastic programming, robust optimization also extends a mathematical program by an uncertainty model. In contrast to stochastic programming, robust optimization overcomes the practical drawback that probabilities are often hard – if not impossible – to identify and hence bound to estimation errors [53]. Instead, solely information on the set of possible outcomes and not on their individual likelihoods is assumed available. According to the classification of [13], robust optimization as first proposed by Soyster [54] and eventually established in the 1990s by the seminal papers [55–57] covers the situation of decision making under ambiguity.

Uncertainty is formally described by an uncertainty set $\mathcal{U}$ which contains all possible realizations of the uncertain problem parameters. A possible parameter configuration i is called an instance to a problem, a perception which shifts the view of an uncertainty inflicted optimization problem $\mathcal{P}_{\mathcal{U}}$ to a set of deterministic problem instances where it is uncertain which one will be realized [58]. Thus, a robust optimization problem can be represented as

$$\mathcal{P}_{\mathcal{U}} = \left\{ \min_x f(i,x) : x \in \mathcal{X}(i) \right\}_{i \in \mathcal{U}}. \tag{8}$$

Notice that this model allows a strict separation between the model of uncertainty and the actual optimization problem. In particular, it is often the case that modelling the uncertainty set as a multi-dimensional set over several parameters is too complex. Instead, the uncertain phenomenon is reduced to a few so-called primitive uncertainties $\xi \in \Xi \subset \mathbb{R}^d$ (with $d \in \mathbb{N}$ and parameter space Ξ) which affinely perturb the individual problem parameters away from some nominal problem instance i^0:

$$i = i^0 + \sum_{d'=1}^{d} \hat{i} \cdot \xi^{d'}. \tag{9}$$

Thereby, the description of uncertainty can be reduced to the description of Ξ and much of the work on static robust optimization has focused on finding tractability results for its different shapes. The most prominent result in this context stems from [59] who introduce a tractable reformulation for problems with a cardinality-constrained uncertainty set. This allows decision makers to define a maximum number of uncertain parameters which may deviate for each constraint which leads to a considerable decrease in conservatism of the resulting solution. For a comprehensive overview on the different uncertainty sets we refer to [60].

As in stochastic programming, a deterministic reformulation of the uncertain problem is needed to define optimality and feasibility. Robust optimization generally optimizes the worst case in the sense of a guaranteed outcome under any possible realization:

$$\min_x \sup_{\xi \in \Xi} f(x, \xi). \tag{10}$$

Regarding feasibility, robust optimization applies what is also referred to as a fat solution approach where only solutions are considered that are feasible for any outcome:

$$x \in \mathcal{X}(\xi), \qquad \xi \in \Xi. \tag{11}$$

Not only does this approach ignore the temporal context, it is also very conservative.

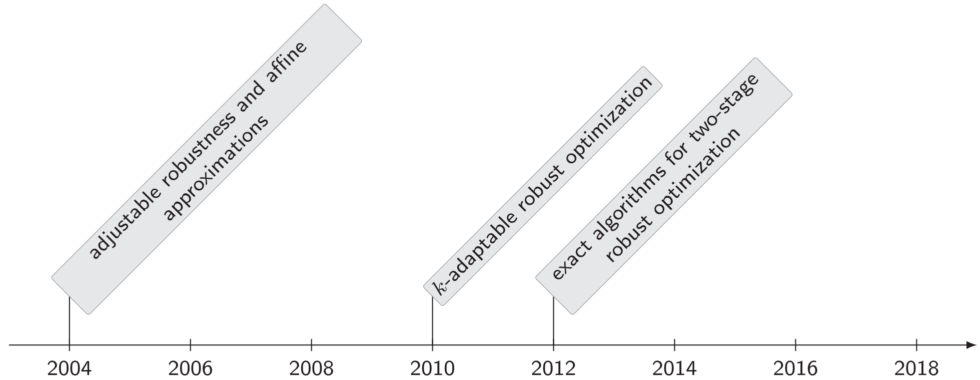

Fig. 7. Evolution of methods and concepts related to robust optimization with regard to multi-stage settings.

As seen in Fig. 7, the multi-stage aspect has been investigated in the robust optimization community only starting in the early 2000s. Ben-Tal et al. [61] introduced the concept of adjustable robust optimization (ARO) which distinguishes between "here-and-now" decisions x_1 that are to be fixed up front and "wait-and-see" decisions $x_2 = X_2(\xi)$ which are allowed to depend on observations of the uncertain data ξ. While the situation depicted resembles that of a two-stage stochastic program, ARO does not aim to determine wait-and-see decisions directly but only the functional form, the decision rule X_2, by which they depend on the uncertain data. The solution to an (affinely) adjustable robust counterpart is therefore giving a more general prescription than that of a stochastic program since decision sequences can be derived not only for a limited number of scenarios but for a range of realizations in a possibly unbounded set. Since optimizing over functions of arbitrary shape leads to intractable problems, the adjustable robust problem is often approximated by restricting decision rules to some predefined functional form. The most prominent restriction is to affine functions [61] such that one optimizes over the coefficients $q \in \mathbb{R}$, and $p \in \mathbb{R}^d$ in a decision rule of the form

$$X_2(\xi) = q + p^T\xi. \quad (12)$$

While affine approximations of the decision rules have been found to perform near-optimal in practice [62] and under certain conditions within a defined optimality gap [63] or even optimal [64], it becomes impracticable when it comes to binary or integer recourse decisions. The most promising way to handle such problem is finite adaptability (or k-adaptability) as presented by Bertsimas and Caramanis [65]. It models decision rules via piece-wise constant functions for k disjunctive partitions of the uncertainty set. In order to determine these partitions iterative splitting procedures have been proposed [66,67]. The resulting solution to a k-adaptable problem resembles that of a stochastic program even more, the only difference being that the policy is now not based on scenarios but on partial uncertainty sets. Bertsimas et al. [68] acknowledged this resemblance and introduced a generalization of a scenario tree in order to model uncertainty evolution in robust multi-stage problems. Hanasusanto et al. [69] extended the idea of k-adaptability to a min-max-min problem formulation with the help of which multi-stage problems with integer or binary recourse decisions can be handed over to state-of-the-art solvers. For a comprehensive overview on ARO it shall be referred to the recently published overview by Yanıkoğlu et al. [10].

A different stream of research focuses on the exact solution of two-stage robust problems. The second stage value function of the first-stage decisions can be gradually constructed by using the dual solutions of the second stage problem, leading to Benders-dual cutting plane algorithms [70,71]. Furthermore, Zeng and Zhao [72] show that two-stage robust problems can be reformulated as bi-level programming problems. In [73] they introduce a column-and-constraint-generation algorithm in which they solve the second stage problem for a subset of relevant scenarios in order to obtain lower bounds and then iteratively add non-trivial scenarios to strengthen these. However, it must be pointed out that these results cannot be transferred adhoc to a multi-stage setting.

2.3. Online optimization

Online optimization has its roots in computer science and is fundamentally different from stochastic programming and robust optimization which emanated from mathematical programming. Online optimization addresses sequential decision making where elements (or requests) σ_t of an input sequence $\sigma_{[T]} = (\sigma_1, \sigma_2, \ldots, \sigma_T)$ are presented sequentially and each element has to be processed by an irrevocable decision before the next element is revealed [2,74]. The original focus was to devise performance guarantees for online algorithms as opposed to an optimal offline algorithm which is not subject to any uncertainty. The processing decision is taken according to an online algorithm $\mathcal{A}$ which determines a decision for each request without knowing subsequent requests. When identifying the arrival of the t-th request σ_t with the observation of the uncertain element ξ_t at stage t and the action of an online algorithm with the partial decision x_t to be fixed at that stage, it is evident that online optimization also deals with uncertainty inflicted multi-stage optimization. However, it offers two new perspectives missing in mathematical programming. First, no information is required on subsequent requests to make a decision at a certain stage – neither of probabilistic nor of set-based nature. In particular, the complexity of the decision at stage t is unaffected by the overall problem size. Second, online optimization is generally associated with problems that require quick, short-term decision making under a constant inflow of information [75]. Considering the computational challenges of mathematical programming in a multi-stage setting, the authors feel that it is necessary to mention online optimization as an approach equally relevant to multi-stage optimization. This need has also been acknowledged by researchers since approaches combining stochastic programming and robust optimization with online optimization have gained importance over the past years as seen in Section 2.4.

The historic development of online optimization is sketched in Fig. 8. The only theoretical concept that is agreed upon widely is competitive analysis (see, e.g., [2,74]) which is formally explained in Section 2.6. Informally, online algorithms have to compete with an optimal offline algorithm which knows the whole input in advance and provable quality guarantees have to hold for arbitrary input sequences. Hence, competitive analysis is a worst-case consideration of a worst-case analysis since the quality guarantee not only has to hold over all inputs but also against the strongest pos-

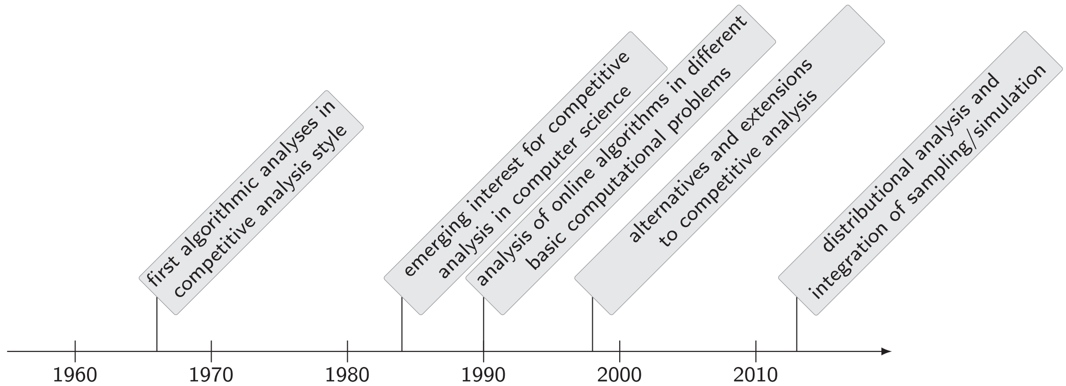

Fig. 8. Evolution of methods and concepts related to online optimization with regard to multi-stage settings.

sible hypothetical adversary in the form of an optimal offline algorithm. Technically, competitive analysis leads to a classification of online algorithms as approximation algorithms where the scarce resource is not time, but future information [2]. The first competitive analysis has been introduced by Graham [76] for machine scheduling; it has been picked up by Sleator and Tarjan [77] which marks the starting point of flourishing research interest.

Similar to robust optimization, competitive analysis is overly pessimistic and does not reflect an algorithm's ability to suitably deal with a given problem [74,78]. In order to overcome this disadvantage, several enhancements to competitive analysis were proposed (see, e.g., the surveys in [78–80]) such as increasing the power of the online algorithm, reducing the power of the offline algorithm, restricting the instance space, applying alternative objective functions, or randomizing the online processing.

Additionally, online optimization does not take into account any information on the probability of future input elements. Online algorithms are tailored to optimize such a worst case performance indicator which often leads to poor average behaviour. Hence, an alternative type of analysis is average case analysis where stochastic assumptions with regard to the elements of the input sequence are made. In this sense, the notion of competitiveness has also been transferred to stochastic settings ([81], [82]). On the other hand, assumptions about the input sequence are also made to soften the worst-case character of competitive analysis: The fair adversary model requires an input sequence to conform with a specified set of constraints that exclude pathological worst cases [83]; likewise, certain patterns or even distributional information about the input sequence may be assumed [84,85].

Lately, gaps between pure online optimization and other, previously unconsidered aspects of multi-stage optimization under uncertainty have been addressed: Dunke and Nickel [86] introduce a general framework for online optimization with look-ahead where look-ahead amounts to partial (deterministic) knowledge about immediately successive input elements. Dunke and Nickel [87] outline the use of simulation models to assess online algorithms in more realistic problem settings which would be inaccessible for competitive analysis. In [88], so-called double time horizons consisting of a short term time horizon with most of the data known and a long term horizon with most of the data uncertain are presented. Finally, there are concepts which allow to make parts of the input instances known in exchange for paying some costs: In the setting of explorable (or queryable) uncertainty (cf. [89,90]) additional problem information can be queried by an algorithm and competitive analyses are carried out in the corresponding setting. Likewise, online optimization with advice complexity (cf. [91,92]) asks through which amount of additional information to be queried an algorithm could reach optimal competitive ratios.

2.4. Combinations of stochastic programming, robust optimization, and online optimization

As the relevance of online optimization for uncertain multistage optimization problems was widely acknowledged, mixed forms with robust optimization and stochastic programming have evolved as seen in the timeline in Fig. 9, trying to combine the benefits of two methods.

2.4.1. Online stochastic optimization

Online stochastic optimization is based on the idea of integrating available stochastic data into an online algorithm via on-the-fly sampling of uncertain data. A framework of online stochastic (combinatorial) optimization has mainly been coined over several years by van Hentenryck and Bent starting with their initial contributions on the expectation, consensus, and regret algorithm [93]. These algorithms operate based on the fact that when the distribution $\mathcal{F}_\sigma$ of the elements of the input sequence σ is known, the online algorithm $\mathcal{A}$ may use the probabilistic information at each stage via sampling. The algorithm samples future developments from this known distribution and then takes a decision optimizing the expected objective value change in this very stage. The additional information can then be used to base the decision on the maximization of the expected profit

$$\mathbb{E}_{\sigma\sim\mathcal{F}_\sigma}[f(\mathcal{A}(\sigma))]. \tag{13}$$

Clearly, in case of limitations on computational time, approximations are necessary which are provided by the consensus and regret algorithms analyzed in [94]. Conceptually, online stochastic optimization integrates stochastic programming into an online algorithm to employ probabilistically motivated anticipatory algorithms; at each stage a multi-stage stochastic program is solved, but only first stage decisions are implemented. Thereby, it inherits the algorithmic structure from online algorithms while at the same time extending them by sampling from the distribution of the elements of σ to generate scenarios of the future at each decision time. Furthermore, online stochastic optimization inherits the property of stochastic programming that uncertainty does not depend on past decisions while moving away from a priori optimization in favour of making decisions as the algorithm executes. Lately, Mercier and Van Hentenryck [95] have generalized the one-step anticipatory algorithms to a multi-step version (based on sample average approximation and heuristic search algorithms for exactly solving Markov decision processes).

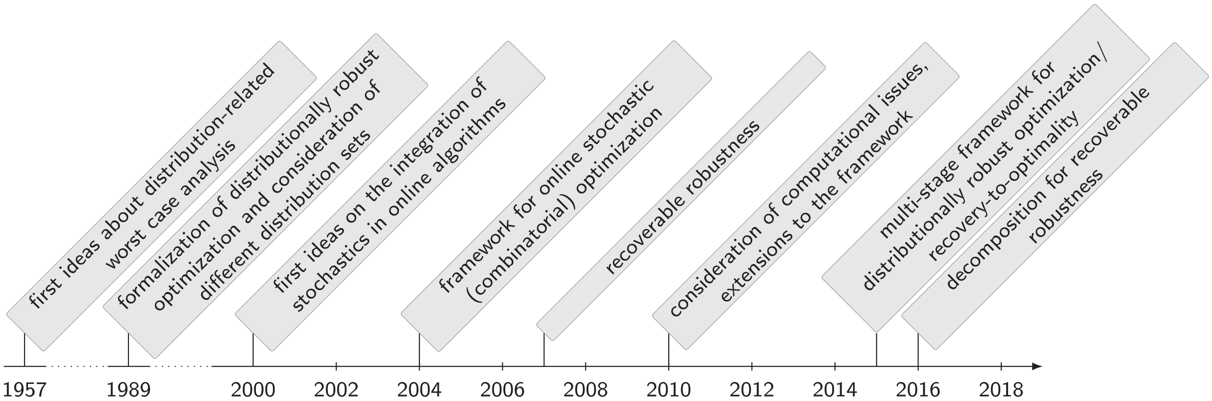

Fig. 9. Evolution of methods and concepts related to Combinations of stochastic programming, robust optimization, and online optimization with regard to multi-stage settings.

2.4.2. Online and robust optimization

2.4.2.1. Recoverable robustness. One approach to combine robust and online optimization is recoverable robust optimization, first presented by Cicerone et al. [96] in the context of creating stable train timetables that are able to cope with disruptions in the operational phase. Its concept is to split the decision process into a strategic planning phase, in which a recoverable robust solution is determined, and an operational phase, during which a sequence of disruptions occur and the solution is modified to retain feasibility by means of an online recovery algorithm. Hence, the concept of robustness has been altered from requiring the solution to be feasible for all realizations of the uncertain data to that of recoverable robustness which only requires that the solution can be recovered to feasibility in every case. The concept is further formalized in [97] and generalized to the multi-stage setting in [98,99] under the requirement that the recovered solution has to be kept feasible when answering to unfolding uncertainty realizations. As opposed to the mathematical programming based concept of adjustable robustness, recoverable robust optimization allows to react upon smaller disruptions with rather fast and simple strategies.

Recoverable robust optimization on the one hand inherits the algorithmic notation of the recovery algorithm $\mathcal{A}_{rec}$ from online optimization. On the other hand, the perception of the uncertain problem as a collection of instances i is inherited from robust optimization. Thereby, random influences (or incoming disruptions) ξ_t shift the problem instance i^t from one to another instance i^{t+1} according to some modification function M: $I \times \Xi \to I$. A sequence of disturbances then results in a sequence of instances. The recoverable robust solution contains two elements. An optimal policy $x^0_{[T]}$ for the nominal problem instance an optimal recovery algorithm $\mathcal{A}_{rec}$ modifying that solution after each disturbance.

The concept of recoverable robustness has been recast in several application-driven contexts: Liebchen et al. [100] have established the same idea of recovery robustness and applied it to delay resistant train timetabling and linear programming. Also [101] introduce recovery robustness in a framework for the so-called robust feasibility recovery applicable to integer programming problems. The framework is instantiated for the repositioning of transportation resources. Goerigk and Schöbel [102] introduce recovery-to-optimality which differs from recovery robustness in that it seeks to recover to an optimal (and not only feasible) solution once uncertainty has revealed and in that it considers a cost function for the recovery operation rather than an algorithmic recovery procedure. The concept can be interpreted as a special case of recovery robustness; the authors present a sampling heuristic to solve the problem approximately and apply this outline to linear programming and aperiodic timetabling. To improve the practical handling of recovery robustness, Van den Akker et al. [103] adapt the branch and price scheme resulting in two decomposition schemes (separate and combined recovery decomposition) for tackling recovery robustness with state-of-the art opportunities for mathematical programming. Lately, recoverable robustness has even been transferred to a bi-objective setting taking into account the original objective value and the recovery costs with the goal of determining the Pareto frontier [104]. Finding a robust solution is interpreted as an analogon to solving a classic center location problem.

2.4.2.2. Holistic adaptive robust optimization. A very recent approach to combining online optimization and robust optimization is presented by Bertsimas et al. [105], who introduce the concept of Holistic Adaptive Robust Optimization (HARO) at the example of the k-server problem. The idea is to combine the ability of online algorithms to base the current decisions on information on the past observations and decisions with the ability of adjustable robust optimization to include assumptions on the future. For this purpose they model future requests via an uncertainty set. At each time step they base the decision of the online algorithm not only on the current greedy term but augment the cost of each decision by the incurred costs of the subsequent time steps as determined via the optimal solution of the associated (affinely) adjustable robust counterpart.

2.4.3. Stochastic programming and robust optimization

2.4.3.1. Distributionally robust stochastic optimization. Distributionally robust stochastic optimization is a paradigm which has been revived rather recently. It combines ideas from robust optimization and stochastic programming, and it acknowledges the fact that the probability distribution $\mathcal{F}$ governing the uncertain data ξ is often subject to uncertainty itself [106]. Scarf [107] proposed a first formulation in which the optimization problem is reformulated with respect to the worst-case costs over a set of probability distributions $\mathcal{D}$ which is assumed to contain the true probability distribution $\mathcal{F}$. Hence, the resulting distributionally robust stochastic program becomes

$$\min_{x\in\mathcal{X}} \left(\max_{\mathcal{F}\in\mathcal{D}} \mathbb{E}_{\mathcal{F}}(f(x,\xi)) \right). \tag{14}$$

Since its introduction different forms of $\mathcal{D}$ have been considered. In this context we would like to point out that if one limits $\mathcal{D}$ to be the set of all distributions that put all their weight at a single point in the parameter's support, then Eq. (14) reduces to the robust optimization problem according to [55]. Delage and Ye [108] propose sets of distributions derived from the confidence regions for the estimated moments of the probability distributions based on a finite set of data samples. A unifying framework to modelling these

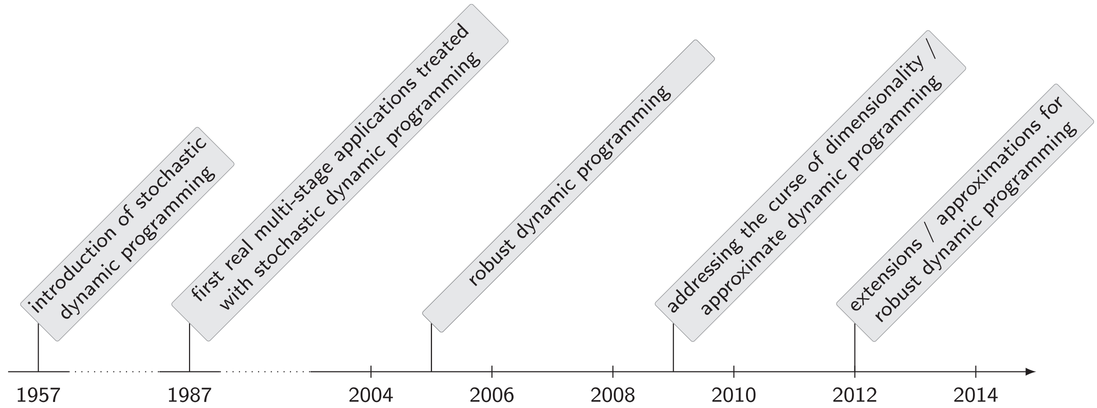

Fig. 10. Evolution of methods and concepts related to dynamic programming with regard to multi-stage settings.

sets is then provided by Wiesemann et al. [106] who introduced a standardized description of what they call ambiguity sets containing many of the sets introduced earlier as special cases. The interested reader shall also be referred to their work for a comprehensive review of distributionally robust stochastic programming. Most work on distributionally robust stochastic programming focused on two-stage stochastic programs. In 2014, however, the framework has been extended to the multi-stage setting by Analui and Pflug [109] who introduced a model for ambiguity sets in the case of multi-stage uncertainty. Given that in multi-stage stochastic programming uncertainty is depicted by a tree they propose what they call "ambiguity neighborhoods" around this tree as alternative models which are close to the baseline model according to the nested distance.

We would like to point out that the use of risk measures (such as the Conditional Value at Risk) in stochastic programming is often seen as a conceptual balance between stochastic programming and robust optimization. However, this stems from the idea that the resulting solutions are more robust towards variations in the uncertain parameters and not from the inclusion of elements from the robust optimization paradigm. Therefore, rather than combining these two approaches we feel that the connection of risk measures and robustness is due to the fact that the term "robustness" is not uniquely defined.

2.5. Dynamic programming

When reviewing solution concepts for sequential decision making problems, the theory of dynamic programming as introduced by [110] cannot be left aside. Multi-stage optimization problems are modelled over a state-action space in which I_t is the set of possible states of the system at stage t and $\mathcal{X}_t$ represents the set of feasible actions (or decisions) that the decision maker may choose from. A decision $x_t \in \mathcal{X}_t$ results in the transition from state $i_t \in I_t$ in stage t to state $i_{t+1} \in I_{t+1}$ in stage $t+1$. In the deterministic set-up the goal is to find a sequence of actions, a so-called policy, which maximizes the reward function $f_t(i_t)$ associated with the final stage state $t = T$. It is found by iteratively solving the Bellman equations which recursively determine the optimal reward obtained through the optimal policy $x_1, \ldots, x_T$ where $g_t(x_t, i_t)$ is the immediate reward in state i_t if decision x_t is implemented:

$$f_t(i_t) = \min_{x_t \in \mathcal{X}_t} \{g_t(x_t, i_t) + f_{t+1}(i_{t+1})\}. \quad (15)$$

2.5.1. Stochastic dynamic programming

Incorporating uncertainty has been addressed in the theory of dynamic programming early on starting with [110] as seen in the timeline in Fig. 10. In stochastic dynamic programming, the transitions between stages occur based on probabilities $p(i_{t+1}|x_t, i_t)$ which may be simultaneously state- and action-dependent. The optimal policy in this case is designed to optimize the expected reward function, thereby drawing a link to traditional stochastic programming approaches. It is found again by recursively solving Bellman's dynamic programming equations with discount factor α

$$f_t(i_t) = \min_{x_t \in \mathcal{X}_t} \left\{ \mathbb{E}[g_t(x_t, i_t)] + \alpha \sum_{i_{t+1}} p(i_{t+1}|x_t, i_t) \cdot f_{t+1}(i_{t+1}) \right\}. \quad (16)$$

Stochastic dynamic programming was linked early with the well-established concept of Markov chains so that problems solved with stochastic dynamic programming are often formalized by Markov decision processes [111]. It requires that the transition probabilities fulfill the Markov property, i.e, the transition probability of one state to another must only depend on the current state and action and not on previous states:

$$p(i_{t+1} = i | i_t, i_{t-1}, \ldots, i_0) = p(i_{t+1} = i | i_t). \quad (17)$$

Despite dynamic programming being a solution method rather than a modelling method, problems are often formulated specifically tailored towards dynamic programming putting emphasis on a state space variable and transition probabilities between states. In addition, there are several other differences between dynamic programming and multi-stage stochastic programming models (which could be solved by dynamic programming in case of a Markovian process structure) such as different point of views on uncertainty description (transitions vs. parameters), solution methods (solution of Bellman equations through policy/value iteration or fixed point methods vs. mathematical programming and decomposition methods), number of stages (large vs. modest number of stages), and objective function (discounted average costs vs. expectation).

However, with large state-spaces problems often become impractically large and suffer from the curse of dimensionality ([58]) which is why until the late 1980s stochastic dynamic programming has not been developed further. Due to the optimal substructure assumed in each setting treated by dynamic programming, the concept can be successfully applied to a large number of stages or an infinite planning horizon with discounted objective. Nonetheless, there are three imminent sources (state space size, action space size, outcome space) for unleashing the curse of dimensionality. Therefore, approximate dynamic programming [112] has been devised to alleviate these issues, i.e., to eliminate a considerable amount of computations. It makes use of different ways to approximate the elements (state space, action space, outcome space, value function) required for solving a dynamic program, e.g., by using Monte Carlo simulation in order to simulate the system in a forward manner for obtaining approximations of the value function that are only based upon a fraction of all states.

2.5.2. Robust dynamic programming

Robust Markov decision processes extend classical Markov decision processes by additionally imposing parameter uncertainty with respect to state transition probabilities $p(i_{t+1}|x_t, i_t)$. Hence, another layer of uncertainty (also called ambiguity) is used to robustify classical Markov decision processes. The goal is then to find a policy which minimizes the maximum expected cost (or maximize the minimum expected reward) given the uncertainty in the transition probabilities and nature playing these probabilities to the decision maker's greatest disadvantage. The setting has been formally proposed in [113,114] with the aim to maximize the worst-case (minimum) expected cost under uncertain state transition probabilities. To foster computational tractability, only rectangular uncertainty sets $\mathcal{P}$ are allowed for each uncertain probability p. In this case, the robust counterpart of the Bellman recursion is shown to be valid by solving

$$f_t(i_t) = \min_{x_t \in \mathcal{X}_t} \left\{ \mathbb{E}[f_t(x_t, i_t)] + \left\{ \max_{p \in \mathcal{P}} \sum_{i_{t+1}} p(i_{t+1}|x_t, i_t) \cdot f_{t+1}(i_{t+1}) \right\} \right\}. \quad (18)$$

In [113], different uncertainty models are analyzed with respect to their effect on the complexity of the robust counterpart. An introduction along with a specification of several uncertainty sets in line with this concept is also provided in the monograph by Ben-Tal et al. [58]. Likewise, for finite horizons the Bellman equation is established. In order to make the concept more applicable, Wiesemann et al. [115] assume that historic data on the Markov decision process under investigation is available. This data is then translated into a confidence region for the uncertain transition probabilities by maximum likelihood estimation. It is also shown which goodness of approximations can be achieved by a policy improvement scheme. With the special class of so-called k-rectangular uncertainty sets, Mannor et al. [116] introduce a case which is computationally tractable and overcomes some of the overconservatism prevalent in the classical robust optimization concept. Sinha and Ghate [117] finally present an approximate policy iteration algorithm for robust non-stationary Markov decision processes. Fig. 10 includes the methodological evolution of robust dynamic programming.

2.6. Performance measures

The following discussion reveals that different methods use substantially different measures and even advocate different scopes to be investigated when evaluating performance. For the sake of simplicity, we refer to minimization problems.

2.6.1. Stochastic programming: Value of stochastic solution and expected value of perfect information

Since stochastic programming encompasses stochastic information about scenarios as the outcome of random events, the inherent question asks about the value of this stochastic information in terms of impact on attainable solution quality [3]. Therefore, the outcome of a stochastic program may in a first step be compared to the outcome of a deterministic model where all stochastic parameters are replaced by their expectation value. To this end, let ξ be the vector of stochastic parameters of a stochastic program, x be the vector of decision variables (addressing all stages), and $f(x, \xi)$ be the objective function. Moreover, let $\bar{\xi} := \mathbb{E}(\xi)$ be the vector of expectation values for the parameters in ξ. Then define EV as the optimal objective in the problem where ξ is replaced by $\bar{\xi}$, i.e., $EV := \min_x f(x, \bar{\xi})$. Additionally, define $\bar{x} = \arg\min_x f(x, \bar{\xi})$ as the expected value solution. Since stochastic programming is based upon the assumption that the assumed stochastic information is true, implementing $\bar{x}$ will lead to an expected objective value of the expected value solution defined as $EEV := \mathbb{E}(f(\bar{x}, \xi))$. On the other hand, when we solve the stochastic program, we obtain SP as the optimal objective, i.e., $SP := \min_x \mathbb{E}(f(x, \xi))$ and it shall be remarked that this value is built upon the recourse possibilities that arise throughout the multiple stages. Additionally, define $x^{sp} := \arg\min_x \mathbb{E}(f(x, \xi))$. Since the assumed distribution of the stochastic programming model is deemed to be true, there is no need to define another expectation, and the value of the stochastic solution x^{sp} compared to the expected value solution $\bar{x}$ is $VSS := EEV - SP$. On the other hand, one could ask what could be achieved if not only stochastic, but perfect information was known. Therefore, define $f^*_\xi := \min_x f(x, \xi)$ as the optimal objective value when ξ is known and $PI := \mathbb{E}(f^*_\xi)$ as the expected objective value in the case of perfect information. The expected value of perfect information is then defined as $EVPI := SP - PI$. Finally, we remark that both VSS and $EVPI$ are evaluation concepts that refer to the value of stochastic information under the requirement that the stochastic assumptions made are perfectly valid.

2.6.2. Robust optimization: Price of robustness

There are two different definitions for the price of robustness. The first one by Bertsimas and Sim [59] refers to robust linear programming and is based upon introducing additional parameters for each row. The price of robustness in this context is the trade-off between feasibility violation probability and impact on the objective value. The parameters in turn can be set so as to specify how "far" deviations can go apart from nominal coefficient values. As the trade-off depends on the problem and the kind of integration of the parameters, there is no formula for this type of price of robustness, but it can be thought of as the change in the objective per change in parameters. Unfortunately, this notion of a price of robustness does not explicitly take into account the recourse possibilities which arise in multi-stage robust optimization problems. Therefore, we turn attention to the second definition from the realms of recovery robustness in [118] where we find an algorithm-related definition similar to the one known from competitive analysis. Let I be the set of all problem instances and let $f(\textsc{Alg}(i))$ be the objective of a recovery robust algorithm Alg on instance $i \in I$, i.e., an algorithm which has to react upon disruptions in a recourse-wise manner as encountered in multi-stage settings. Then the price of robustness of Alg is defined as the largest ratio

$$\max_{i \in I} \frac{f(\textsc{Alg}(i))}{f(\mathrm{OPT}(i))} \quad (19)$$

over all instances $i \in I$, where OPT is an algorithm which entirely knows an instance $i \in I$ and is capable of computing an optimal solution to it. For more detailed guidance on how to apply and interpret the price of robustness it shall be referred to [119].

2.6.3. Online optimization: Competitive ratio

Since there is no information about the future given in online optimization, competitive analysis in pure online optimization cannot relate to the value of information. Focusing on another ideal setting, competitive analysis deals with a comparison to a hypothetical omniscient offline algorithm Opt [2]. Let Σ be the set of all input sequences. The idea of competitive analysis is to directly compare the performance of an online algorithm Alg to that of Opt. Let $\textsc{Alg}(\sigma)$ be the objective attained by algorithm Alg on instance $\sigma \in \Sigma$. Alg is called c-competitive if there is a constant a such that $\textsc{Alg}(\sigma) \le c \cdot \textsc{Opt}(\sigma) + a, \sigma \in \Sigma$. The role of the additive constant a is to facilitate an asymptotic analysis and to make results independent of initial conditions for finite input sequences. In the case $a = 0$, Alg is called strictly c-competitive if

$$\frac{\textsc{Alg}(\sigma)}{\textsc{Opt}(\sigma)} \le c, \qquad \sigma \in \Sigma. \quad (20)$$

Table 1
Overview of performance measures for methods dealing with multi-stage settings.

	Method		
	SP	RO	OO
Performance measure	Value of stochastic solution expected value of perfect information	Price of robustness	Competitive ratio
Evaluated element of optimization problem	Stochastic information	Algorithm quality	Algorithm quality

Table 2
Overview on the uncertainty models deployed by the different methods. Required information must be specified in order to use the model. Optional information can be captured by the model but is not necessary for the application of the method. Entries in parentheses indicate that this type of uncertainty information appears in specific settings, but is unusual in general.

Information on uncertainty		Method							
		SP	RO	OO	OSO	RRO	DRO	SDP	RSDP
Required	Set of possible outcomes	+	+		+		+	+	+
		support of $\mathcal{F}$	uncertainty set $\mathcal{U}$		support of $\mathcal{F}_\sigma$		support of $\mathcal{F} \in \mathcal{D}$		state space I
	Probability of individual outcomes	$+\mathcal{F}$		(+)	$+\mathcal{F}_\sigma$		$(+)\mathcal{D}$	$+p(i_{t+1}\|x_t, i_t)$	$(+)\mathcal{P}$
Optional	Infinite # of outcomes	(+)	+	(+)	(+)	+	+		
	Conditional probabilities	+	(+)			+	+	+	+
	Seperate uncert. and problem	+	+		+		+		

Hence, a c-competitive algorithm is a c-approximation algorithm with the additional restriction that it has to compute online. The competitive ratio c_r of Alg is the greatest lower bound over all c such that Alg is c-competitive, i.e., $c_r = \inf\{c \geq 1 \mid \text{Alg}(\sigma) \leq c \cdot \text{Opt}(\sigma) + a, \sigma \in \Sigma\} = \inf\{c \geq 1 \mid \text{Alg is } c\text{-competitive}\}$. The competitive ratio states how much the performance of Alg degrades with respect to Opt due to the lack of information in the worst-case. In contrast to the measures from stochastic programming, not the value of information is assessed, but an algorithm's performance guarantee.

Table 1 summarizes the established measures. Neither of the concepts of online stochastic optimization, fuzzy optimization, stochastic dynamic programming, or distributionally robust optimization yield a specific performance measure that particularly addresses the impact of uncertainty. In all of these methods the plain objective value is used in order to assess different models or algorithms.

2.7. Conclusion from review on methods and concepts

In order for a decision maker to choose a suitable method from the above, for any given problem two questions must be considered:

1. What problem information is needed to deploy the method? (required input)
2. What solution information does the method yield? (granted output)

Regarding the first question, Table 2 summarizes the features of the respective uncertainty models needed to apply a particular method. The required features in the upper part of the table can be broken down to whether the set of possible outcomes of the uncertain parameters must be specified and whether information on probabilities must be stated. Evidently, stochastic programming and derived concepts require most information as they need both features. On the other hand, online optimization allows to ignore uncertainty and focus on the information available at any particular stage.

As seen in the lower part of the table, some uncertainty models allow to model additional features or information on the uncertain phenomenon. While robust and online optimization can deal with an infinite number of outcomes, stochastic programming generally requires a discretization at some step of the solution procedure. Also, depending on the application it might be beneficial to model uncertainty independently from problem parameters, thereby reducing modelling effort to primitive uncertainties instead of a large number of parameters (e.g., in case of uncertain customer demands one can model the random variable by few economic indicators instead of all individual demands). This approach is by default implemented in robust optimization and it could also be incorporated into models amenable to stochastic programming. In the context of multi-stage problems another interesting aspect concerns the consideration of conditional probabilities of realizations between time stages. While stochastic programming based methods can easily incorporate that information, e.g., through asymmetric scenario trees, robust optimization has left this idea largely unexplored, even though recent advances have been made by Lorca and Sun [120] (dynamic uncertainty sets) and [121] (generalized scenario trees). Notice that in particular with the additional information that can be integrated into the uncertainty model one would expect that solution quality would increase given the information was correct. However, at this point none of the above methods is equipped with appropriate means to really efficiently exploit additional information, even if available. Finally, we recognize that online optimization and its derivatives do not describe uncertainty at all in their methodological outlines.

The second question regards the information a decision maker can expect when deploying a particular method. The overview in Table 3 distinguishes between information that can be derived ex ante, i.e., at the beginning of the planning horizon (particularly relevant for strategic planning), and ex post, i.e., at the end of the planning horizon after uncertainty has been disclosed. Looking at the ex ante information provided on the objective function it becomes evident that the methods yield different insights making a comparison based on these values practically meaningless. Also, it becomes clear that online optimization does not yield insights for strategic planning. If one was to compare the concepts, the comparison can only take place in the ex post evaluation of the objective function, i.e., what objective value was obtained in a particular scenario. This suggests that performance comparisons might best be done based on simulation studies.

Concerning the solution information it must be pointed out that a clear disadvantage of stochastic programming is that in case of a discretized stochastic process the decision maker does not know

Table 3
Overview on the solution information yielded by the different methods on the objective value and the sequence of decisions. Ex ante considers information before the start of the planning horizon, while ex post considers information at the end of the planning horizon, hence after the observation of the random outcome ξ.

	Method					
	SP	RO	OO	OSO	RRO	DRO
Objective value						
- ex ante	$\mathbb{E}[f(x,\xi)]$	$\sup_{\xi} f(x,\xi)$			$f(x^0, i^0)$	$\sup_{\mathcal{F}} f(x,\xi)$
- ex post	$f(x,\bar{\xi})$ *	$f(x,\bar{\xi})$	$f(x,\bar{\xi})$	$f(x,\bar{\xi})$	$f(x^{rec}, i^T)$	$f(x,\bar{\xi})$
Decision sequence						
- ex ante	$x(\xi)$	$x(\xi)$			$\left(x^0, \mathcal{A}^{rec}\right)$	$x(\xi)$
- ex post	$x(\bar{\xi})$	$x(\bar{\xi})$	x^*	x^*	x^{rec}	$x(\bar{\xi})$

*Can only be determined if policy has been determined for the observed realization $\bar{\xi}$

	method	
	SDP	RSDP
Objective value		
- ex ante	$\mathbb{E}[f_T(i_T)]$	$\sup_{\mathcal{P}} \mathbb{E}[f_T(i_T)]$
- ex post	$f_T(i_T)$	$f_T(i_T)$
Decision sequence		
- ex ante	$x_{it},\ i \in I,\ t \in T$	$x_{it},\ i \in I,\ t \in T$
- ex post	x^*	x^*

Table 4
Overview of application focuses for the methods dealing with multi-stage settings. S / T / O = strategic / tactical / operational planning level, $\rightarrow$ = with feedback to, +++ / ++ / + = application very frequently / often / occasionally addressed by method.

Application	Method							Uncertainties
	SP	RO	OO	OSO	RRO	DRO	(R)SDP	
Supply chain, production, inventory	+++	++	+			+	++	Demands, capacities
Scheduling			+++	++				Tasks, dates
Energy, electricity, power	+++	++	+	+		+	++	Demands, prices, resources
Environment, water, air, waste	++							Demands resources emissions inflows
Finance, investment	++		+++			++		Returns
Traffic, transportation	+	+	++	++				Travel times, orders, locations
Packing, loading		+	+++					Sizes, weights
Healthcare	+		+				+	Demands
Staffing, rostering	+							Requirements, demands
Telecommunication	+							Requests, bandwidth
Data structures			+++					Requests
Timetabling		+			++			Arrivals, events
Projects							++	Durations, returns
Chemical engineering	++							Physical data
Planning horizon	S/T	$O \rightarrow S/T$	O	O	$O \rightarrow S/T$	O	T/O	

how to proceed when uncertainty realizes outside one of the scenarios. This disadvantage can be eased to a certain extent when applying stochastic dual dynamic programming which allows to sample scenario tree paths in an on-the-fly manner and to bypass the need to formulate the multi-stage problem in its entirety. However, convergence then becomes an issue and finite convergence can only be shown in the case of multi-stage linear stochastic programming problems with a finite number of scenarios [27]. Finally, only online optimization along and its derivatives allow for an open planning horizon; all other methods derive their output for a closed planning horizon. Hence, to use closed-horizon methods such as stochastic programming or robust optimization in an open-horizon setting, one would have to embed them into a rolling horizon scheme.

3. Applications

This section reports on multi-stage applications that have been dealt with by the different methods and concepts presented in Section 2. We preface the discussion with the overview in Table 4 and remark that only applications with more than two stages have been taken into account capturing the multi-stage feature in the decision making process adequately. From an application perspective it becomes clear that very similar problems have been tackled by several different methods. However, research results derived through different methodologies differ strongly from each other, and a converging perspective taking into account results across different disciplines for the same application is missing. Ultimately, this has made it prohibitive to find out about the method most suitable to a specific problem and it also impedes to build an information pool for application results that would be independent of chosen optimization methodologies.

3.1. Stochastic programming

Applications of multi-stage stochastic programming exist for several domains focusing mainly on strategic and tactical decision making. In most cases, stages coincide with periods. A large share of publications addresses capacitated production planning and lot sizing as well as related problems such as capacity expansion planning, inventory management, or production routing [122–137]. Demand is by far the most prevalent uncertain parameter and it is mainly modelled using scenario trees. Amongst others, yield quality, available capacities as well as lead and processing times are rarely used as uncertain parameters. Typical decisions to be determined for the different stages involve capacity configurations, capacity extensions, technology choices, network designs as well as stage-wise production and transportation quantities. Capacity adaptations are also encountered as part of recourse actions. Apart from modelling and implementation, some works focus on specific aspects such as different approaches for scenario tree generation or specialized algorithmic outlines to deal with several sources of uncertainty. Conceptually, the value of a multi-stage solution compared to a two-stage solution has been introduced.

Another focus of applications for multi-stage stochastic programming encompasses energy management, power generation, and electricity markets [138–149]. Similar to production, decisions concern capacity allocations for energy generation units as well as produced energy quantities for different energy forms such as electricity or renewables. Another frequent decision concerns investments for the installation of new generation plants and lines in the first model stages and dispatching decisions ensuring a reliable supply to energy demands in subsequent stages. Sources of uncertainty comprise energy demands or loads followed by energy prices, greenhouse gas emission quotas, power output quantities, and hydro-inflows in hydro-thermal energy power systems. Advanced topics address scenario tree generation and reduction, specific integration of risk measures, decomposition methods, relaxation schemes and sophisticated solution routines such as stochastic dual dynamic programming.

Multi-stage stochastic programming is also applied for dynamic portfolio optimization and asset/liability management in financial investment models [150–157]. Asset returns and prices are the driving uncertain factor; amongst others interest and exchange rates or wage developments are occasionally used as uncertain parameters. Decisions comprise investment allocations and rebalancing decisions aiming to prescribe a portfolio investment strategy. Authors also concentrate on delimiting scenario tree sizes to accomplish model outputs in realistic market settings, e.g., by scenario decomposition, or state and time aggregation. Alternatively, solution method-related aspects such as decomposition or sample average approximation are discussed to empower solution schemes in the multi-stage setting.

In addition to these application areas, multi-stage stochastic programming is encountered in the following domains: In water resources management [158–162], decision makers have to ensure that – in spite of droughts and limited access to water – people are supplied under uncertain water supplies, demands, flows and availabilities. The health care sector [163–166] yields different types of applications such as clinical trial planning in new drug development with uncertain trial outcomes, optimizing the annual influenza vaccine by determining flu shot designs and production schedules where the uncertain factor comprises strain prevalences and production yields, assigning nurses to patients in hospitals throughout working shifts where patients' required time amounts of care determine the scenario set, and scheduling appointments where service durations and the number of customers are uncertain and may include no-shows. Finally, multi-stage stochastic programming has been employed for planning logistics infrastructure and operations in the oil industry [167,168], workforce capacity planning [169], airport terminal capacity planning, airline network revenue management [170,171] and team deployment in dynamic disaster management [172].

3.2. Robust optimization

Literature on application papers of multi-stage decision making based on robust optimization is sparse. In order to capitalize on uncertainty realizations and to adjust decisions accordingly, adjustable robustness in multi-stage applications has first been considered in the context of robust inventory management [61] where demands are uncertain and recourse decisions are constrained to be affine functions of previous uncertainty realizations. The problem can then be solved using the formulation of an affinely adjustable robust counterpart. Likewise, adjustable robustness has been considered in other supply chain management applications [62,173,174] in two-echelon as well as in multi-echelon supply systems with uncertain demands. Besides affine dependency on uncertainty realizations, conceptual extensions such as a globalized robust counterpart which allows infeasibility for a sufficiently small share of the instances as well as related dynamic programming schemes are presented. Other applications that have been treated in a multi-stage context with adjustable robustness include emergency logistics in [175] with uncertain demands for traffic flows, train load planning in container terminals in [176] with uncertain weights of transportation units, corner castings overhangs, and bug parameters, process scheduling in [177] with uncertain task processing times, utilities consumption, availability, and process yields, and power generation in [178] with uncertain electricity demand and renewable generation. Postek et al. [179] applied k-adjustable robust optimization to the problem of determining optimal dike heights in flood protection. In order to promote the application of robust optimization techniques [119] provide practical guidance for practitioners.

3.3. Online optimization

With the goal of providing a worst-case performance guarantee (competitive analysis) many problem settings deal with online versions of basic discrete optimization problems from computer science or complexity theory. By definition, online optimization is a sequential decision making method where decisions are made in a stage-wise manner. Hence, every application paper on online optimization would be fitting here. Nonetheless, several focal points are clearly perceivable. Most applications appear in an operational planning horizon due to the requirement of repeatedly and frequently processing input elements of the same type. By far the most attention have received packing and scheduling. Packing problems [180–188] mostly allude to bin packing and variants such as the multi-dimensional version, variable-sized bins, bounded space for open bins, or strip packing. Uncertain elements comprise the sizes of the items to be packed. Besides improvements of lower bounds for competitive ratios also conceptual and algorithmic enhancements are considered such as primal dual algorithms, average case analysis, or resource augmentation with augmented bin sizes.

In machine scheduling [189–199], competitive analysis and enhancements as well as alternative measures are considered for a multitude of machine environments and processing characteristics. Uncertainty mainly encompasses job processing times. However, also other characteristics such as release dates may be uncertain. Specialized settings address resource augmentation in the form of different machine speeds, problems with release dates, batch processing, job migration between machines, or job rejections. Whereas in machine scheduling the objective typically considers job-related functions, the objective in the related topic of load balancing [200–202] considers the load of the machines and opts at minimizing the maximum load ever achieved. Also for load balancing several extensions such as related machines or fairness constraints are discussed.

Another spotlight in online optimization is put on metrical task systems ([85,203–207] which allow to represent the operations of an algorithm in a configuration-wise manner where state transitions are assessed by a metric. The k-server problem and paging are two basic problems from computer science which are special metrical task systems. Uncertainty affects tasks that have to be processed in order to form a processing sequence of the tasks. Competitive analysis as well as several refinements and alternatives (loose competitiveness, max-max ratio, comparative ratio, or diffuse adversaries) have been introduced and analyzed in the realms of metrical task systems, *k*-server problems and paging. Likewise, many applications of online optimization can be found in the areas of data management and data transmission [208–214] such as storage allocation, page migration, list update, packet transmission and buffering, web caching, or data routing. Request types depend on the specific application and typically amount to different forms of storage or retrieval requests, or bandwidth for traffic requests. The goal lies in managing different kinds of data operations with minimum costs which are composed of access and administrative (data organization) costs.

Other application papers can be found for routing [215–217] where locations have to be visited by a server, financial planning [218–221] where capital investment strategies such as buy-and-hold approaches, search and trading algorithms, or leasing strategies are evaluated, health care services where treatments have to be scheduled or patients have to be transported [222,223]. Moreover, there exist papers on competitive algorithms in specialized settings such as inventory management, power management, or robotic exploration [224–226].

3.4. Combinations of stochastic programming, robust optimization, and online optimization

The number of application papers on combinations of stochastic programming, robust optimization, and online optimization is rather limited. Moreover, each combination has been strongly coined by a group of few authors.

3.4.1. Online stochastic optimization and recoverable robustness

Starting in 2004 [93,227,228], several prototype applications from different sorts of scheduling were taken into consideration. It was shown that sampling provides meaningful results in order to make scenario-based but still informed decisions when used by the expectation, consensus or regret algorithm. In online packet scheduling for computer networks the packets to be served arise online and the goal is to minimize packet loss under packet distributions that are available for sampling. Also for online vehicle routing problems it is shown that distributions of customer service requests can be learned online or from historical data, and they can be processed in the online stochastic optimization approach. Finally, the seminal paper for theoretical aspects of online stochastic optimization [94] contains an illustration of the framework along with computational results from vehicle routing, vehicle dispatching, and packet scheduling. Afterwards the concept has also been applied to inspecting and repairing power systems [228] and energy scheduling of home automation systems [229]. In the former setting natural disasters may cause power infrastructure to suffer harms and the goal is to send out repair crews to restore functional capability; algorithms use sampled scenarios to guide the damage assessment and restoration based on a so-called power restoration vehicle routing problem. In the latter setting, uncertainties are considered in real-time prices, weather conditions, and occupant behaviour, and the goal is to schedule consumption activities of a set of home automation devices so as to reduce costs while maintaining comfort and convenience.

3.4.2. Recoverable robustness

Only two papers address recoverable robustness in a multi-stage context with more than two stages. Cicerone et al. [99] provides a framework for recovery robust optimization as well as an evaluation concept (price of robustness). These concepts are applied to timetabling in public transportation where an initial timetable with minimal passenger waiting times is recovered during operations in case of unforeseen delays. This task also known as delay management and it is shown how it can be accomplished by different restrictions of a recovery algorithm. The basic ideas for the concept and the application have already been introduced less comprehensively in [118].

3.4.3. Distributionally robust stochastic programming

An introduction to financial decision making under uncertainty including a discussion of multi-stage distributionally robust optimization and its use in asset-liability management is given in [230] with uncertain asset returns. Since the exposition is only conceptual, no details on the ambiguity set specifications (distribution family) of the random variable driving the stochastic process are given. In [108], a framework for distributionally robust optimization under moment uncertainty is first introduced and then applied to portfolio optimization. It allows for uncertainty both in distributions and related moments. Probabilistic arguments justify the use of the framework whenever historical data can be used to describe the stochastic process of the investment returns. A multi-period robust portfolio selection model is considered in [231] with special emphasis on the integration of a multi-period worst-case risk measure that is also presented in the paper. Multi-stage distributionally robust optimization is related to stochastic dual dynamic programming in [232] by imposing that an optimal policy for a multi-stage stochastic program is sought over the worst-case probability distribution in some family of distributions. The devised algorithm is applied to hydrothermal scheduling where water inflow distributions are sampled from historical data. A distributionally robust chance-constrained programming model for the multi-stage distribution expansion planning in power systems is given in [233]. Here distributional robustness is understood as ensuring worst case probabilities with which feasible solutions are obtained when moment-based ambiguity sets are used to model wind generations and loads. Finally, the supply chain management application of dynamic network design under demand uncertainty [234] has also been tackled with a distributionally robust chance constrained model using a worst case conditional value at risk approximation for the chance constraints when only partial distribution information is known such as mean and variance of demand.

3.5. Dynamic programming

We preface the discussion of applications by clarifying that we only consider such applications where a stage really corresponds to a time stage and not to an artificial stage of a decision making process that could be carried out instantaneously.

3.5.1. Stochastic dynamic programming

In the context of energy planning, a stochastic dynamic programming model is considered for scheduling a hydrothermal generating system ([235]) where distributional information on water inflows is part of the required data. Due to the curse of dimensionality resulting from a state space discretization, stochastic dual dynamic programming is proposed to obtain computational results. The same methodological outset is used in [236] for the bidding problem of a price-maker hydropower-based company taking into account several hydro plants, time-coupling and stochastic inflow scenarios. Project management under uncertainty is tackled by stochastic dynamic programming in [237,238] based on the idea of real option values which allows to change the course of actions depending on observed uncertainty realizations in a project. Therefore, performance realizations or market developments are modelled by probability distributions. Several topics from supply chain management are also covered: Capacity reservation, procurement decisions, or competitive bidding strategies are addressed in

[239], [240]. Demands, spot market prices, and costs are modelled as random variables and it is even possible to characterize optimal procurement and reservation policies. In [241], lot sizing is considered for a company which performs successive auctions for revenue generation and inventory clearance. Under stochastic demands the authors provide a structural analysis of optimal policies which solve the specified stochastic dynamic programming model. Although designed for health care operations, the stochastic dynamic program for the capacity allocation for demands of different products and services in [242] opts at answering the rather general question of how to respond to booking requests over several periods. Thus, stochastic customer-type specific demands represent the uncertain part in the Markov decision process which is solved for a radiology department by dynamic programming. In health care, stochastic dynamic programming has also been used in a multi-stage setting for scheduling the operating room utilization [243] with stochastic demands and decisions to be made on each day leading up to the day of surgery. Apart from economic applications also technological aspects such as encountered in sensor scheduling [244] are treated. The goal consists of selecting stagewise which one of a set of noisy measurement devices to select in order to obtain an overall measurement profile.

3.5.2. Robust dynamic programming

Robustness for dynamic programming and Markov decision processes has only been considered for very few multi-stage applications: Scheduling charging operations for electric vehicle charging [245] under uncertain wind supply is modelled as a robust stochastic shortest path problem underlying the robust version of Bellman's equation. Uncertainties affect the available wind power at different stages and translate into state transition probabilities. Multi-stage optimization problems related to production and inventory management under Markovian uncertainty with uncertain customer demands or procurement prices are formulated in [246] and treated with dynamic programming. In particular, a comparison is sought between robust and stochastic problem versions and it is found that the robust approach can outperform the stochastic approach under low risk measured by the value at risk concept. Finally, Dimitrov et al. [247] yields an application of a robust decomposable Markov decision process to the allocation of school funding. The authors model the funding allocation problem of a school district as a Markov decision process where a state is a vector of performance states for the individual schools and schools transition between performance states. Exact transition probabilities are not known, but rather uncertainty sets where probabilities have to be contained in. A robust version of the Bellman recursion provides a solution method in this setting.

3.6. Conclusion from review on applications

Essentially, multi-stage optimization under uncertainty plays a role whenever a system is considered which undergoes time progression and requires decisions throughout the course of time. Thereby, every rolling horizon planning scheme which controls a system stepping from configuration to configuration can be viewed as a multi-stage planning process. Therefore, in applications the above concepts may also be encountered under different names which do not directly point towards one of the discussed methods but rather deal with a “dynamic” version of some problem (see, e.g., [135,248–255]).

Concerning the applications, two observations deserve special mentioning: First, only stochastic programming and online optimization have reached a mature state with respect to the relevance in the overall research community on multi-stage optimization; all other disciplines were either driven by particular researchers, or applications are scattered among different domains. Hence, apart from stochastic programming and online optimization, each of the other approaches has only been investigated in a rather small amount of applications, especially with regards to an explicit consideration of the multi-stage character. Second, the topics of energy planning, environmental planning, and health care planning have experienced increasing popularity in accordance with current societal needs over the past few years.

Apart from methodological incongruities, the discussion of the applications showed that there seem to be favourable applications that are covered by some specific method in a multi-stage setting or that some concepts were even devised more or less from an application-oriented motivation. While there are certainly good reasons for this correspondence between method and application, there may also be several good reasons or at least no obvious hindrances to try some of the other methods for any application under scrutiny. Nonetheless, the current state of analysis for multi-stage optimization problems under uncertainty has to be summarized as shown in the left part of Fig. 11 exhibiting pre-determined combinations of uncertainty model and solution methods to derive rather specific statements in given application contexts. Overall, this leads to an accumulation of non-interrelated knowledge about problem domains, often only applicable to rather specific problem assumptions and settings.

4. Conclusion and outlook

While sequential decision making under uncertainty is a generic task required in a variety of applications, there is no unified understanding on how to address it. Several concepts exist, yet they are not necessarily complementary. Rather have these paradigms been developed largely independently from each other - often on an application-driven basis with researchers having a bias towards a certain concept. The result is the situation depicted in this paper: Concepts exist in parallel, deploy different terminology, and there is a lack of definitions on how they overlap and differ.

Section 2 reviewed the theoretical underpinnings of the different concepts for solving multi-stage optimization problems under uncertainty that can be found in the areas of mathematical programming and computer science. Section 3 shows that a large number of applications has been treated with more than one of these concepts already. Yet few papers consider more than one concept at the same time. Therefore, there are also no statements available why one or the other method might be favourable in a specific application. The results from Section 2 suggest that concepts differ with respect to three main aspects: the required information on the uncertain data – the uncertainty model –, the information they yield for decision makers, – the (prescriptive) solution information –, and the way the quality of the obtained solution is assessed – the performance evaluation. While the first makes it difficult to apply different concepts to a particular problem, the latter makes a comparison between methods cumbersome. Nevertheless, we believe that the results of one specific method can only be understood in full when put into context with results from alternative methods on the same application.

Therefore, besides providing an overview on the different concepts that coexist for solving uncertainty inflicted multi-stage optimization problems, the results from the survey clearly indicate a need for a mutual framework for multi-stage optimization problems. As Fig. 11 (i) illustrates, the current state of the analysis is as follows: When solving a problem in a particular application, a concept is chosen first. Based on the requirements of this concept the uncertain data is modeled and a solution derived. This solution is then again evaluated using the performance measures associated with this concept. We feel that a unifying framework is the first step in order to arrive at the more desirable state of analysis as depicted in Fig. 11 (ii). When given a particular application,

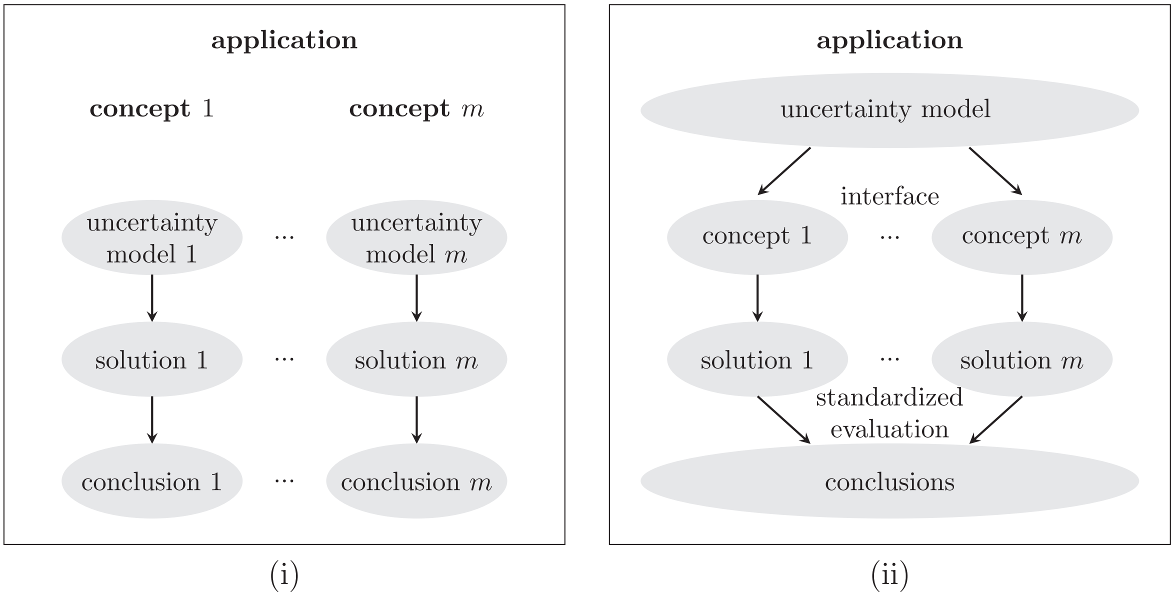

Fig. 11. (i) Current state of analysis where uncertainty model, solution and conclusion depend on the concept chosen in a particular application; (ii) Desirable state of analysis where a problem in a particular application is modeled with a generalised uncertainty model, different solutions are obtained depending on the concept and an overall conclusion is drawn.

uncertainty is modeled comprehensively in such a way that the information required by different concepts can be derived with the help of respective interfaces. From a business perspective, a unified model of uncertainty will represent a significant step towards the integration of big data and analytics as it would allow for a well-defined data retrieval in time-dynamic optimization under uncertainty. Standardized handling of large amounts of data becomes increasingly important and deriving insights on uncertain parameters via analytical tools can yield significant benefits for the subsequent optimization. In this next step, different solutions may be derived with the help of different concepts. These solutions are then to be evaluated by standardized performance measures in order to come to a mutual conclusion. In order to arrive at such a framework much research is still needed. By design of the concepts, the (prescriptive) solution information they provide is very different. Therefore, new standardized performance measures need to be developed. Furthermore, the need for a unified framework does not solely exist in a multi-stage context but can equally be raised in a more general context of optimization under uncertainty. However, it is the multi-stage character of the problem that creates the link from stochastic programming and robust optimization to concepts like online optimization or dynamic programming which makes the potential value of such a framework even greater in the multi-stage setting. The authors have recently launched a research project to establish a unified model for multi-stage decision making under uncertainty [256].

To the best of our knowledge there have only been limited attempts for such a framework which explicitly addressed the multi-stage problem character: Pflug and Pichler [257] formalize multi-stage uncertainty through discrete stochastic processes which can be used for scenario tree generation. In the general problem formulation, different attitudes of decision makers are cast by various forms of risk functionals, and uncertainty is grasped by probability functionals. Multi-stage optimization subject to geometrically described uncertainty sets is considered in [121] where finitely adaptable solutions are analyzed. In this concept, a set of solutions is tentatively computed making every uncertainty realization answerable with a feasible solution. As a generalization of the scenario tree description, multi-stage uncertainty is described by a directed acyclic network. Based on this model of uncertainty, general forms of multi-stage stochastic and multi-stage adaptive optimization problems are introduced and analyzed for specific uncertainty sets.

While models for multi-stage optimization under uncertainty have often been addressed from a specific application-driven point of view (pre-determining the style of uncertainty representation and solution methodology), we believe that the insights and classification possibilities shown in this review can form the basis of a consistent and undistorted model for the analysis of multi-stage uncertainty taking into account potentials of a variety of uncertainty models, solution methods, and evaluation techniques. This review goes into details mainly with respect to different concepts, approaches, and applications. We believe that an additional survey on available solution algorithms (in particular specializing on the multi-stage setting) would complement the contents of this review.

Acknowledgments

This work was partly supported by the German Research Foundation (DFG) [project no. 354864080]. This support is gratefully acknowledged.

References

[1] Dantzig G. Linear programming under uncertainty. Manag. Sci. 1955;1(3/4):197–206.
[2] Borodin A, El-Yaniv R. Online computation and competitive analysis. 1st paperbacked. Cambridge University Press; 2005.
[3] Birge J, Louveaux F. Introduction to Stochastic Programming. 2nd. Springer; 2011.
[4] Powell WB. A unified framework for stochastic optimization. Eur. J. Oper. Res. 2019;275(3):795–821.
[5] Sahinidis N. Optimization under uncertainty: state-of-the-art and opportunities. Comput. Chem. Eng. 2004;28(6–7):971–83.
[6] Herroelen W, Leus R. Project scheduling under uncertainty: survey and research potentials. Eur. J. Oper. Res. 2005;165(2):289–306.
[7] Govindan K, Fattahi M, Keyvanshokooh E. Supply chain network design under uncertainty: a comprehensive review and future research directions. Eur. J. Oper. Res. 2017;263(1):108–41.
[8] Birge J. State-of-the-art-survey-stochastic programming: computation and applications. INFORMS J. Comput. 1997;9(2):111–33.
[9] Gabrel V, Murat C, Thiele A. Recent advances in robust optimization: an overview. Eur J Oper Res 2014;235(3):471–83.

[10] Yanıkoğlu İ, Gorissen B, den Hertog D. A survey of adjustable robust optimization. Eur J Oper Res 2018.
[11] Albers S. Online algorithms: a survey. Math Program 2003;97(1):3–26.
[12] Homem-de Mello T, Bayraksan G. Monte carlo sampling-based methods for stochastic optimization. Surv Oper Res Manag Sci 2014;19(1):56–85.
[13] Rosenhead J, Elton M, Gupta S. Robustness and optimality as criteria for strategic decisions. J Oper Res Soc 1972(23):413–31.
[14] Markowitz H. Portfolio selection: efficient diversification of investments. J Finance 1952;7(1):77–91.
[15] Jorion P. Value at risk: the new benchmark for managing financial risk. McGraw-Hill; 1996.
[16] Rockafellar R, Uryasev S. Optimization of conditional value-at-risk. Journal of Risk 2000;2:21–41.
[17] Louveaux F. A solution method for multistage stochastic programs with recourse with application to an energy investment problem. Oper Res 1980;28(4):889–902.
[18] Lenstra J, Rinnooy Kan A, Stougie L. A framework for the probabilistic analysis of hierarchical planning systems. Annal Oper Res 1984;1(1):23–42.
[19] Dupačová J, Consigli G, Wallace S. Scenarios for multistage stochastic programs. Annal Oper Res 2000;100(1–4):25–53.
[20] Olsen P. Multistage stochastic programming with recourse as mathematical programming in an lp space. SIAM J Control Optim 1976;14(3):528–37.
[21] Olsen P. When is a multistage stochastic programming problem well-defined? SIAM J Control Optim 1976;14(3):518–27.
[22] Heitsch H, Römisch W, Strugarek C. Stability of multistage stochastic programs. SIAM J Optim 2006;17(2):511–25.
[23] Maggioni F, Pflug G. Bounds and approximations for multistage stochastic programs. SIAM J Optim 2016;26(1):831–55.
[24] Shapiro A. On complexity of multistage stochastic programs. Oper Res Lett 2006;34(1):1–8.
[25] Dyer M, Stougie L. Computational complexity of stochastic programming problems. Math Program 2006;106(3):423–32.
[26] Pereira M. Optimal stochastic operations scheduling of large hydroelectric systems. Int J Electr Power Energy Syst 1989;11(3):161–9.
[27] Shapiro A. Analysis of stochastic dual dynamic programming method. Eur J Oper Res 2011;209(1):63–72.
[28] Gassmann HI. MSLIp: a computer code for the multistage stochastic linear programming problem. Math Program Ser B 1990;47(3):407–423.
[29] Messina E, Mitra G. Modelling and analysis of multistage stochastic programming problems: a software environment. Eur J Oper Res 1997;101(2):343–359.
[30] Birge J. Decomposition and partitioning methods for multistage stochastic linear programs. Oper Res 1985;33(5):989–1007.
[31] Ruszczyński A. Decomposition methods. Handbook Oper Res Manag Sci 2003;10(C):141–211.
[32] Ruszczyński A. Parallel decomposition of multistage stochastic programming problems. Math Program 1993;58(1–3):201–28.
[33] Lulli G, Sen S. A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems. Manag Sci 2004;50(6):786–96.
[34] Guan Y, Ahmed S, Nemhauser G. Cutting planes for multistage stochastic integer programs. Oper Res 2009;57(2):287–98.
[35] Singh K, Philpott A, Kevin Wood R. Dantzig-wolfe decomposition for solving multistage stochastic capacity-planning problems. Oper Res 2009;57(5):1271–86.
[36] Høyland K, Wallace SW. Generating scenario trees for multistage decision problems. Manag Sci 2001;47(2):295–307.
[37] Casey M, Sen S. The scenario generation algorithm for multistage stochastic linear programming. Math Oper Res 2005;30(3):615–31.
[38] Heitsch H, Römisch W. Scenario tree modeling for multistage stochastic programs. Math Program 2009;118(2):371–406.
[39] Shapiro A. Monte carlo sampling methods. Handbook Oper Res Manag Sci 2003;10(C):353–425.
[40] Rebennack S. Combining sampling-based and scenario-based nested benders decomposition methods: application to stochastic dual dynamic programming. Math Program 2016;156(1–2):343–89.
[41] Shapiro A. Minimax and risk averse multistage stochastic programming. Eur J Oper Res 2012;219(3):719–26.
[42] Homem-De-Mello T, Pagnoncelli B. Risk aversion in multistage stochastic programming: a modeling and algorithmic perspective. Eur J Oper Res 2016;249(1):188–99.
[43] Shapiro A. On a time consistency concept in risk averse multistage stochastic programming. Oper Res Lett 2009;37(3):143–7.
[44] Shapiro A, Tekaya W, Da Costa J, Soares M. Risk neutral and risk averse stochastic dual dynamic programming method. Eur J Oper Res 2013;224(2):375–91.
[45] Kozmík V, Morton D. Evaluating policies in risk-averse multi-stage stochastic programming. Math Program 2015;152(1–2):275–300.
[46] Collado R, Papp D, Ruszczyński A. Scenario decomposition of risk-averse multistage stochastic programming problems. Annal Oper Res 2012;200(1):147–70.
[47] Shapiro A. Stochastic programming approach to optimization under uncertainty. Math Program 2008;112(1):183–220.
[48] Goel V, Grossmann I. A class of stochastic programs with decision dependent uncertainty. Math Program 2006;108(2–3):355–94.
[49] Gupta V, Grossmann I. Solution strategies for multistage stochastic programming with endogenous uncertainties. Comput Chem Eng 2011;35(11):2235–47.
[50] Gupta V, Grossmann I. A new decomposition algorithm for multistage stochastic programs with endogenous uncertainties. Comput Chem Eng 2014;62:62–79.
[51] Tarhan B, Grossmann I, Goel V. Computational strategies for non-convex multistage MINLP models with decision-dependent uncertainty and gradual uncertainty resolution. Annal Oper Res 2013;203(1):141–66.
[52] Apap R, Grossmann I. Models and computational strategies for multistage stochastic programming under endogenous and exogenous uncertainties. Comput Chem Eng 2017;103:233–74.
[53] King AJ, Wallace SW. Modeling with stochastic programming. Springer; 2012.
[54] Soyster AL. Constraints and applications to inexact linear programming. Oper Res 1973;21(5):1154–7.
[55] Ben-Tal A, Nemirovski A. Robust convex optimization. Math Oper Res 1998;23(4):769–805.
[56] Ben-Tal A, Nemirovski A. Robust solutions of uncertain linear programs. Oper Res Lett 1999;25(1):1–13.
[57] El Ghaoui L, Oustry F, Lebret H. Robust solutions to uncertain semidefinite programs. SIAM J Optim 1998;9(1):33–52.
[58] Ben-Tal A, El Ghaoui L, Nemirovski A. Robust optimization. Princeton University Press; 2009.
[59] Bertsimas D, Sim M. The price of robustness. Oper Res 2004;52(1):35–53.
[60] Goerigk M, Schöbel A. Algorithm engineering in robust optimization. Lect Note Comput Sci 2016;9220 LNCS:245–79.
[61] Ben-Tal A, Goryashko A, Guslitzer E, Nemirovski A. Adjustable robust solutions of uncertain linear programs. Math Program 2004;99(2):351–76.
[62] Ben-Tal A, Golany B, Nemirovski A, Vial J-P. Retailer-supplier flexible commitments contracts: a robust optimization approach. Manuf Serv Oper Manag 2005;7(3):248–71.
[63] Bertsimas D, Bidkhori H. On the performance of affine policies for two-stage adaptive optimization: a geometric perspective. Math Program 2015;153(2):577–94.
[64] Bertsimas D, Iancu DA, Parrilo PA. Optimality of affine policies in multistage robust optimization. Math Oper Res 2010;35(2):363–94.
[65] Bertsimas D, Caramanis C. Finite adaptability in multistage linear optimization. IEEE Trans Autom Control 2010;55(12):2751–66.
[66] Bertsimas D, Dunning I. Multistage robust mixed-integer optimization with adaptive partitions. Oper Res 2016;64(4):980–98.
[67] Postek K, den Hertog D. Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. INFORMS J Comput 2016;28(3):553–74.
[68] Bertsimas D, Gupta V, Kallus N. Data-driven robust optimization. Math Program 2018;167(2):235–92.
[69] Hanasusanto GA, Kuhn D, Wiesemann W. *K*-Adaptability in two-stage robust binary programming. Oper Res 2015;63(4):877–91.
[70] Bertsimas D, Litvinov E, Sun XA, Zhao J, Zheng T. Adaptive robust optimization for the security constrained unit commitment problem. IEEE Trans Power Syst 2013;28(1):52–63.
[71] Gabrel V, Lacroix M, Murat C, Remli N. Robust location transportation problems under uncertain demands. Discr Appl Math 2014;164:100–11.
[72] Zeng B, Zhao L. An exact algorithm for two-stage robust optimization with mixed integer recourse problems. Tech. Rep. University of South Florida; 2012.
[73] Zeng B, Zhao L. Solving two-stage robust optimization problems using a column-and-constraint generation method. Oper Res Lett 2013;41(5):457–61.
[74] Fiat A, Woeginger G. Online algorithms: the state of the art. Springer; 1998.
[75] Dunke F, Heckmann I, Nickel S, Saldanha-da Gama F. Time traps in supply chains: is optimal still good enough? Eur J Oper Res 2018;264(3):813–29.
[76] Graham R. Bounds for certain multiprocessing anomalies. Bell Syst Tech J 1966;45(9):1563–81.
[77] Sleator D, Tarjan R. Amortized efficiency of list update and paging rules. Commun ACM 1985;28(2):202–8.
[78] Dorrigiv R. Alternative measures for the analysis of online algorithms. University of Waterloo; 2010.
[79] Boyar J, Irani S, Larsen K. A comparison of performance measures for online algorithms. Algorithmica 2015;72(4):969–94.
[80] Hiller B, Vredeveld T. Probabilistic alternatives for competitive analysis. Comput Sci- Res Dev 2012;27(3):189–96.
[81] Naaman N, Rom R. Average case analysis of bounded space bin packing algorithms. Algorithmica 2008;50(1):72–97.
[82] Souza A. Average performance analysis. Eidgenössische Technische Hochschule Zürich; 2006.
[83] Ausiello G, Allulli L, Bonifaci V, Laura L. On-line algorithms, real time, the virtue of laziness, and the power of clairvoyance. In: Cai J, Cooper S, Li A, editors. Theory and Applications of Models of Computation. Springer; 2006. p. 1–20.
[84] Albers S, Favrholdt L, Giel O. On paging with locality of reference. J Comput Syst Sci 2005;70(2):145–75.
[85] Koutsoupias E, Papadimitriou C. Beyond competitive analysis. SIAM J Comput 2000;30(1):300–17.
[86] Dunke F, Nickel S. A general modeling approach to online optimization with lookahead. Omega 2016;63:134–53.
[87] Dunke F, Nickel S. Evaluating the quality of online optimization algorithms by discrete event simulation. Central Eur J Oper Res 2017;25(4):831–58.

[88] Mitrović-Minić S, Krishnamurti R, Laporte G. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. Transp Res Part B 2004;38(8):669–85.
[89] Erlebach T. Computing and scheduling with explorable uncertainty. In: Manea F, Miller RG, Nowotka D, editors. Sailing Routes in the World of Computation. Springer; 2018. p. 156–60.
[90] Dürr C, Erlebach T, Megow N, Meißner J. Scheduling with explorable uncertainty, 94; 2018.
[91] Böckenhauer H-J, Komm D, Královic R, Královic R, Mömke T. On the advice complexity of online problems. Lect Note Comput Sci (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 2009;5878 LNCS:331–40.
[92] Böckenhauer H-J, Komm D, Královic R, Královic R. On the advice complexity of the *k*-server problem. Lect Note Comput Sci (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 2011;6755 LNCS(PART 1):207–18.
[93] Van Hentenryck P, Bent R. Online stochastic combinatorial optimization. MIT Press; 2006.
[94] Van Hentenryck P, Bent R, Upfal E. Online stochastic optimization under time constraints. Annal Oper Res 2010;177(1):151–83.
[95] Mercier L, Van Hentenryck P. An anytime multistep anticipatory algorithm for online stochastic combinatorial optimization. Annal Oper Res 2011;184(1):233–71.
[96] Cicerone S, D'Angelo G, Di Stefano G, Frigioni D, Navarra A. Robust algorithms and price of robustness in shunting problems. OpenAccess Ser Inf 2007;7:175–90.
[97] Cicerone S, D'Angelo G, Di Stefano G, Frigioni D, Navarra A. Delay management problem: complexity results and robust algorithms. In: Yang B, Du D, Wang C, editors. Combinatorial optimization and applications. Springer; 2008. p. 458–68.
[98] Cicerone S, D'Angelo G, Di Stefano G, Frigioni D, Navarra A. Recoverable robust timetabling for single delay: complexity and polynomial algorithms for special cases. J Combin Optim 2009;18(3):229.
[99] Cicerone S, Di Stefano G, Schachtebeck M, Schöbel A. Multi-stage recovery robustness for optimization problems: a new concept for planning under disturbances. Inf Sci 2012;190:107–26.
[100] Liebchen C, Lübbecke M, Möhring R, Stiller S. The concept of recoverable robustness, linear programming recovery, and railway applications. In: Ahuja R, Möhring R, Zaroliagis C, editors. Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems. Springer; 2009. p. 1–27.
[101] Erera A, Morales J, Savelsbergh M. Robust optimization for empty repositioning problems. Oper Res 2009;57(2):468–83.
[102] Goerigk M, Schöbel A. Recovery-to-optimality: a new two-stage approach to robustness with an application to aperiodic timetabling. Comput Oper Res 2014;52:1–15.
[103] Van den Akker J, Bouman P, Hoogeveen J, Tönissen D. Decomposition approaches for recoverable robust optimization problems. Eur J Oper Res 2016;251(3):739–50.
[104] Carrizosa E, Goerigk M, Schöbel A. A biobjective approach to recoverable robustness based on location planning. Eur J Oper Res 2017;261(2):421–35.
[105] Bertsimas D, Jaillet P, Korolko N. The *k*-server problem via a modern optimization lens. Eur J Oper Res 2019;276(1):65–78.
[106] Wiesemann W, Kuhn D, Sim M. Distributionally robust convex optimization. Oper Res 2014;62(6):1358–76.
[107] Scarf HE. A min-max solution of an inventory problem. Stud Math Theory Invent Product 1957:201–9.
[108] Delage E, Ye Y. Distributionally robust optimization under moment uncertainty with application to data-driven problems. Oper Res 2010;58(3):595–612.
[109] Analui B, Pflug GC. On distributionally robust multiperiod stochastic optimization. Comput Manag Sci 2014;11(3):197–220.
[110] Bellman R. Dynamic programming. Princeton University Press; 1957.
[111] Howard R. Dynamic programming and markov processes. MIT Press; 1962.
[112] Powell WB. Approximate dynamic programming: Solving the curses of dimensionality. 2nd. Wiley; 2011.
[113] Nilim A, El Ghaoui L. Robust control of markov decision processes with uncertain transition matrices. Oper Res 2005;53(5):780–98.
[114] Iyengar G. Robust dynamic programming. Math Oper Res 2005;30(2):257–80.
[115] Wiesemann W, Kuhn D, Rustem B. Robust markov decision processes. Math Oper Res 2013;38(1):153–83.
[116] Mannor S, Mebel O, Xu H. Robust MDPs with *k*-rectangular uncertainty. Math Oper Res 2016;41(4):1484–509.
[117] Sinha S, Ghate A. Policy iteration for robust nonstationary markov decision processes. Optim Lett 2016;10(8):1613–28.
[118] Cicerone S, D'Angelo G, Di Stefano G, Frigioni D, Navarra A, Schachtebeck M, et al. Recoverable robustness in shunting and timetabling. In: Ahuja R, Möhring R, Zaroliagis C, editors. Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems. Springer; 2009. p. 28–60.
[119] Gorissen BL, Yanıkoğlu İ, den Hertog D. A practical guide to robust optimization. Omega 2015;53:124–37.
[120] Lorca A, Sun XA. Adaptive robust optimization with dynamic uncertainty sets for multi-period economic dispatch under significant wind. IEEE Trans Power Syst 2015;30(4):1702–13.
[121] Bertsimas D, Goyal V, Sun XA. A geometric characterization of the power of finite adaptability in multistage stochastic and adaptive optimization. Math Oper Res 2011;36(1):24–54.
[122] Denizel M, Ferguson M, Souza G. Multiperiod remanufacturing planning with uncertain quality of inputs. IEEE Trans Eng Manag 2010;57(3):394–404.
[123] Körpeoğlu E, Yaman H, Selim Aktürk M. A multi-stage stochastic programming approach in master production scheduling. Eur J Oper Res 2011;213(1):166–79.
[124] Chen Z-L, Li S, Tirupati D. A scenario-based stochastic programming approach for technology and capacity planning. Comput Oper Res 2002;29(7):781–806.
[125] Beraldi P, Ghiani G, Grieco A, Guerriero E. Fix and relax heuristic for a stochastic lot-sizing problem. Comput Optim Appl 2006;33(2–3):303–18.
[126] Huang K, Küçükyavuz S. On stochastic lot-sizing problems with random lead times. Oper Res Lett 2008;36(3):303–8.
[127] Brandimarte P. Multi-item capacitated lot-sizing with demand uncertainty. Int J Prod Res 2006;44(15):2997–3022.
[128] Huang K, Ahmed S. The value of multistage stochastic programming in capacity planning under uncertainty. Oper Res 2009;57(4):893–904.
[129] Karabuk S, Wu S. Coordinating strategic capacity planning in the semiconductor industry. Oper Res 2003;51(6):839–849+1003–1004.
[130] Zanjani M, Nourelfath M, Ait-Kadi D. A multi-stage stochastic programming approach for production planning with uncertainty in the quality of raw materials and demand. Int J Prod Res 2010;48(16):4701–23.
[131] Zeballos L, Méndez C, Barbosa-Povoa A, Novais A. Multi-period design and planning of closed-loop supply chains with uncertain supply and demand. Comput Chem Eng 2014;66:151–64.
[132] Kim G, Wu K, Huang E. Optimal inventory control in a multi-period newsvendor problem with non-stationary demand. Adv Eng Inf 2014;29(1):139–45.
[133] Adulyasak Y, Cordeau J-F, Jans R. Benders decomposition for production routing under demand uncertainty. Oper Res 2015;63(4):851–67.
[134] Nickel S, Saldanha-da Gama F, Ziegler H-P. A multi-stage stochastic supply network design problem with financial decisions and risk management. Omega 2012;40(5):511–24.
[135] Curcio E, Amorim P, Zhang Q, Almada-Lobo B. Adaptation and approximate strategies for solving the lot-sizing and scheduling problem under multistage demand uncertainty. Int J Prod Econ 2018;202:81–96.
[136] Sawik T. Integrated supply, production and distribution scheduling under disruption risks. Omega 2016;62:131–44.
[137] Correia I, Nickel S, Saldanha-da Gama F. A stochastic multi-period capacitated multiple allocation hub location problem: formulation and inequalities. Omega 2018;74:122–34.
[138] Triki C, Beraldi P, Gross G. Optimal capacity allocation in multi-auction electricity markets under uncertainty. Comput Oper Res 2005;32(2):201–17.
[139] Shina T, Birge J. Multistage stochastic programming model for electric power capacity expansion problem. Japan J Ind Appl Math 2003;20(3):379–97.
[140] Pineda S, Conejo A. Managing the financial risks of electricity producers using options. Energy Econ 2012;34(6):2216–27.
[141] Seddighi A, Ahmadi-Javid A. Integrated multiperiod power generation and transmission expansion planning with sustainability aspects in a stochastic environment. Energy 2015;86:9–18.
[142] Fleten S-E, Kristoffersen T. Short-term hydropower production planning by stochastic programming. Comput Oper Res 2008;35(8):2656–71.
[143] Rebennack S, Flach B, Pereira M, Pardalos P. Stochastic hydro-thermal scheduling under CO2 emissions constraints. IEEE Trans Power Syst 2012;27(1):58–68.
[144] Rebennack S. Generation expansion planning under uncertainty with emissions quotas. Electr Power Syst Res 2014;114:78–85.
[145] López J, Contreras J, Muñoz J, Mantovani J. A multi-stage stochastic non-linear model for reactive power planning under contingencies. IEEE Trans Power Syst 2013;28(2):1503–14.
[146] Nowak M, Römisch W. Stochastic lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty. Annal Oper Res 2000;100(1–4):251–72.
[147] Philpott A, De Matos V. Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion. Eur J Oper Res 2012;218(2):470–83.
[148] Qadrdan M, Wu J, Jenkins N, Ekanayake J. Operating strategies for a GB integrated gas and electricity network considering the uncertainty in wind power forecasts. IEEE Trans Sustain Energy 2014;5(1):128–38.
[149] Morton D. An enhanced decomposition algorithm for multistage stochastic hydroelectric scheduling. Annal Oper Res 1996;64:211–35.
[150] Kouwenberg R. Scenario generation and stochastic programming models for asset liability management. Eur J Oper Res 2001;134(2):279–92.
[151] Topaloglou N, Vladimirou H, Zenios S. A dynamic stochastic programming model for international portfolio management. Eur J Oper Res 2008;185(3):1501–24.
[152] Gondzio J, Kouwenberg R. High-performance computing for asset-liability management. Oper Res 2001;49(6):879–91.
[153] Zenios S, Holmer M, McKendall R, Vassiadou-Zeniou C. Dynamic models for fixed-income portfolio management under uncertainty. J Econ Dyn Control 1998;22(10):1517–41.
[154] Klaassen P. Financial asset-pricing theory and stochastic programming models for asset/liability management: a synthesis. Manag Sci 1998;44(1):31–48.
[155] Fang Y, Chen L, Fukushima M. A mixed r&d projects and securities portfolio selection model. Eur J Oper Res 2008;185(2):700–15.
[156] Ferstl R, Weissensteiner A. Asset-liability management under time-varying investment opportunities. J Bank Finance 2011;35(1):182–92.
[157] Blomvall J, Shapiro A. Solving multistage asset investment problems by the sample average approximation method. Math Program 2006;108(2–3):571–95.

Please cite this article as: H. Bakker, F. Dunke and S. Nickel, A structuring review on multi-stage optimization under uncertainty: Aligning concepts from theory and practice, Omega, https://doi.org/10.1016/j.omega.2019.06.006

[158] Liu X, Huang G, Wang S, Fan Y. Water resources management under uncertainty: factorial multi-stage stochastic program with chance constraints. Stoch Environ Res Risk Assess 2016;30(3):945–57.
[159] Higgins A, Archer A, Hajkowicz S. A stochastic non-linear programming model for a multi-period water resource allocation with multiple objectives. Water Resour Manag 2008;22(10):1445–60.
[160] Wang S, Huang G. Identifying optimal water resources allocation strategies through an interactive multi-stage stochastic fuzzy programming approach. Water Resour Manag 2012;26(7):2015–38.
[161] Watkins DWJ, McKinney D, Lasdon L, Nielsen S, Martin Q. A scenario-based stochastic programming model for water supplies from the highland lakes. Int Trans Oper Res 2000;7(3):211–30.
[162] Zhang Q, Grossmann I, Lima R. On the relation between flexibility analysis and robust optimization for linear systems. AIChE J 2016;62(9):3109–23.
[163] Colvin M, Maravelias C. A stochastic programming approach for clinical trial planning in new drug development. Comput Chem Eng 2008;32(11): 2626–2642.
[164] Özaltın O, Prokopyev O, Schaefer A, Roberts M. Optimizing the societal benefits of the annual influenza vaccine: a stochastic programming approach. Oper Res 2011;59(5):1131–43.
[165] Punnakitikashem P, Rosenberger J, Buckley Behan D. Stochastic programming for nurse assignment. Comput Optim Appl 2008;40(3):321–49.
[166] Erdogan S, Denton B. Dynamic appointment scheduling of a stochastic server with uncertain demand. INFORMS J Comput 2013;25(1):116–32.
[167] Gupta V, Grossmann I. Multistage stochastic programming approach for offshore oilfield infrastructure planning under production sharing agreements and endogenous uncertainties. J Petroleum Sci Eng 2014;124:180–97.
[168] Dempster M, Hicks Pedron N, Medova E, Scott J, Sembos A. Planning logistics operations in the oil industry. J Oper Res Soc 2000;51(11):1271–88.
[169] Song H, Huang H-C. A successive convex approximation method for multistage workforce capacity planning problem with turnover. Eur J Oper Res 2008;188(1):29–48.
[170] Solak S, Clarke J-P, Johnson E. Airport terminal capacity planning. Transp Res Part B 2009;43(6):659–76.
[171] Möller A, Römisch W, Weber K. Airline network revenue management by multistage stochastic programming. Comput Manag Sci 2008;5(4):355–77.
[172] Chen L, Miller-Hooks E. Optimal team deployment in urban search and rescue. Transp Res Part B 2012;46(8):984–99.
[173] Ben-Tal A, Golany B, Shtern S. Robust multi-echelon multi-period inventory control. Eur J Oper Res 2009;199(3):922–35.
[174] Shapiro A. A dynamic programming approach to adjustable robust optimization. Oper Res Lett 2011;39(2):83–7.
[175] Ben-Tal A, Chung B, Mandala S, Yao T. Robust optimization for emergency logistics planning: risk mitigation in humanitarian relief supply chains. Transp Res Part B 2011;45(8):1177–89.
[176] Bruns F, Goerigk M, Knust S, Schöbel A. Robust load planning of trains in intermodal transportation. OR Spectrum 2014;36(3):631–68.
[177] Lappas N, Gounaris C. Multi-stage adjustable robust optimization for process scheduling under uncertainty. AIChE J 2016;62(5):1646–67.
[178] Lorca A, Sun X, Litvinov E, Zheng T. Multistage adaptive robust optimization for the unit commitment problem. Oper Res 2016;64(1):32–51.
[179] Postek K, den Hertog D, Kind J, Pustjens C. Adjustable robust strategies for flood protection. Omega 2019;82:142–54.
[180] Seiden S. On the online bin packing problem. J ACM 2002;49(5):640–71.
[181] Van Vliet A. An improved lower bound for on-line bin packing algorithms. Inf Process Lett 1992;43(5):277–84.
[182] Buchbinder N, Naor J. Online primal-dual algorithms for covering and packing. Math Oper Res 2009;34(2):270–86.
[183] Csirik J, van Vliet A. An on-line algorithm for multidimensional bin packing. Oper Res Lett 1993;13(3):149–58.
[184] Shor P. The average-case analysis of some on-line algorithms for bin packing. Combinatorica 1986;6(2):179–200.
[185] Csirik J. An on-line algorithm for variable-sized bin packing. Acta Inf 1989;26(8):697–709.
[186] Csirik J, Woeginger G. Shelf algorithms for on-line strip packing. Inf Process Lett 1997;63(4):171–5.
[187] Csirik J, Johnson D. Bounded space on-line bin packing: best is better than first. Algorithmica 2001;31(2):115–38.
[188] Csirik J, Woeginger G. Resource augmentation for online bounded space bin packing. J Algorithm 2002;44(2):308–20.
[189] Kalyanasundaram B, Pruhs K. Speed is as powerful as clairvoyance. J ACM 2000;47(4):617–43.
[190] Albers S. Better bounds for online scheduling. SIAM J Comput 1999;29(2):459–73.
[191] Phillips C, Stein C, Torng E, Wein J. Optimal time-critical scheduling via resource augmentation. Algorithmica 2002;32(2):163–200.
[192] Goemans M, Queyranne M, Schulz A, Skutella M, Wang Y. Single machine scheduling with release dates. SIAM J Discr Math 2002;15(2):165–92.
[193] Zhang G, Cai X, Wong C. On-line algorithms for minimizing makespan on batch processing machines. Naval Res Logist 2001;48(3):241–58.
[194] Chen B, Vestjens A. Scheduling on identical machines: how good is LPT in an on-line setting? Oper Res Lett 1997;21(4):165–9.
[195] Megow N, Uetz M, Vredeveld T. Models and algorithms for stochastic online scheduling. Math Oper Res 2006;31(3):513–25.
[196] Chen B, van Vliet A, Woeginger G. New lower and upper bounds for on-line scheduling. Oper Res Lett 1994;16(4):221–30.
[197] Sanders P, Sivadasan N, Skutella M. Online scheduling with bounded migration. Math Oper Res 2009;34(2):481–98.
[198] Hoogeveen H, Potts C, Woeginger G. On-line scheduling on a single machine: maximizing the number of early jobs. Oper Res Lett 2000;27(5):193–7.
[199] Bartal Y, Leonardi S, Marchetti-Spaccamela A, Sgall J, Stougie L. Multiprocessor scheduling with rejection. SIAM J Discr Math 2000;13(1):64–78.
[200] Azar Y, Broder A, Karlin A, Upfal E. Balanced allocations. SIAM J Comput 1999;29(1):180–200.
[201] Berman P, Charikar M, Karpinski M. On-line load balancing for related machines. J Algorithm 2000;35(1):108–21.
[202] Andrews M, Goemans M, Zhang L. Improved bounds for on-line load balancing. Algorithmica 1999;23(4):278–301.
[203] Borodin A, Linial N, Saks M. An optimal on-line algorithm for metrical task system. J ACM (JACM) 1992;39(4):745–63.
[204] Manasse M, McGeoch L, Sleator D. Competitive algorithms for server problems. J Algorithm 1990;11(2):208–30.
[205] Irani S, Rubinfeld R. A competitive 2-server algorithm. Inf Process Lett 1991;39(2):85–91.
[206] Young N. The *k*-server dual and loose competitiveness for paging. Algorithmica 1994;11(6):525–41.
[207] Ben-David S, Borodin A. A new measure for the study of on-line algorithms. Algorithmica 1994;11(1):73–91.
[208] Breslauer D. On competitive on-line paging with lookahead. Theor Comput Sci 1998;209(1–2):365–75.
[209] Irani S. Page replacement with multi-size pages and applications to web caching. Algorithmica 2002;33(3):384–409.
[210] Kesselman A, Mansour Y, Van Stee R. Improved competitive guarantees for qos buffering. Algorithmica 2005;43(1–2):63–80.
[211] Adler M, Sitaraman R, Venkataramani H. Algorithms for optimizing the bandwidth cost of content delivery. Comput Netw 2011;55(18):4007–20.
[212] Lund C, Reingold N, Westbrook J, Yan D. Competitive on-line algorithms for distributed data management. SIAM J Comput 1999;28(3):1086–111.
[213] Bar-Noy A, Canetti R, Kutten S, Mansour Y, Schieber B. Bandwidth allocation with preemption. SIAM J Comput 1999;28(5):1806–28.
[214] Albers S, Schmidt M. On the performance of greedy algorithms in packet buffering. SIAM J Comput 2006;35(2):278–304.
[215] Jaillet P, Wagner M. Generalized online routing: new competitive ratios, resource augmentation, and asymptotic analyses. Oper Res 2008;56(3):745–57.
[216] Ausiello G, Feuerstein E, Leonardi S, Stougie L, Talamo M. Algorithms for the on-line travelling salesman. Algorithmica 2001;29(4):560–81.
[217] Kalyanasundaram B, Pruhs K. The online transportation problem. SIAM J Discr Math 2000;13(3):370–83.
[218] Azar Y, Bartal Y, Feuerstein E, Fiat A, Leonardi S, Rósen A. On capital investment. Algorithmica 1999;25(1):22–36.
[219] Lorenz J, Panagiotou K, Steger A. Optimal algorithms for *k*-search with application in option pricing. Algorithmica 2009;55(2):311–28.
[220] El-Yaniv R, Kaniel R, Linial N. Competitive optimal on-line leasing. Algorithmica 1999;25(1):116–40.
[221] El-Yaniv R, Fiat A, Karp R, Turpin G. Optimal search and one-way trading online algorithms. Algorithmica 2001;30(1):101–39.
[222] Hahn-Goldberg S, Carter M, Beck J, Trudeau M, Sousa P, Beattie K. Dynamic optimization of chemotherapy outpatient scheduling with uncertainty. Health Care Manag Sci 2014;17(4):379–92.
[223] Beaudry A, Laporte G, Melo T, Nickel S. Dynamic transportation of patients in hospitals. OR Spectrum 2010;32(1):77–107.
[224] Albers S, Kursawe K, Schuierer S. Exploring unknown environments with obstacles. Algorithmica 2002;32(1):123–43.
[225] Wagner M. Fully distribution-free profit maximization: the inventory management case. Math Oper Res 2010;35(4):728–41.
[226] Irani S, Shukla S, Gupta R. Online strategies for dynamic power management in systems with multiple power-saving states. ACM Trans Embed Comput Syst 2003;2(3):325–46.
[227] Bent R, Van Hentenryck P. Online stochastic and robust optimization. Lect Note Comput Sci 2004;3321:286–300.
[228] Van Hentenryck P, Gillani N, Coffrin C. Joint assessment and restoration of power systems. Frontier Artif Intell Appl 2012;242:792–7.
[229] Scott P, Thiébaux S, Van Den Briel M, Van Hentenryck P. Residential demand response under uncertainty. Lect Note Comput Sci 2013;8124 LNCS:645–60.
[230] Consigli G, Kuhn D, Brandimarte P. Optimal financial decision making under uncertainty. Int Ser Oper Res Manag Sci 2017;245:255–90.
[231] Liu J, Chen Z-P, Hui Y-C. Time consistent multi-period worst-case risk measure in robust portfolio selection. J Oper Res Soc China 2018;6(1):139–58.
[232] Philpott A, de Matos V, Kapelevich L. Distributionally robust SDDP. Comput Manag Sci 2018:1–24.
[233] Zare A, Chung C, Zhan J, Faried S. A distributionally robust chance-constrained MILP model for multistage distribution system planning with uncertain renewables and loads. IEEE Trans Power Syst 2018;33(5):5248–62.
[234] Sun H, Gao Z, Szeto W, Long J, Zhao F. A distributionally robust joint chance constrained optimization model for the dynamic network design problem under demand uncertainty. Netw Spatial Econ 2014;14(3–4):409–33.
[235] Pereira M, Pinto L. Multi-stage stochastic optimization applied to energy planning. Math Program 1991;52(1–3):359–75.
[236] Flach B, Barroso L, Pereira M. Long-term optimal allocation of hydro generation for a price-maker company in a competitive market: latest developments and a stochastic dual dynamic programming approach. IET Gener Transm Distrib 2010;4(2):299–314.

[237] Eckhause J, Hughes D, Gabriel S. Evaluating real options for mitigating technical risk in public sector r&d acquisitions. Int J Project Manag 2009;27(4):365–77.
[238] Huchzermeier A, Loch C. Project management under risk: using the real options approach to evaluate flexibility in r&d. Manag Sci 2001;47(1):85–101.
[239] Inderfurth K, Kelle P, Kleber R. Dual sourcing using capacity reservation and spot market: optimal procurement policy and heuristic parameter determination. Eur J Oper Res 2013;225(2):298–309.
[240] Takano Y, Ishii N, Muraki M. A sequential competitive bidding strategy considering inaccurate cost estimates. Omega 2014;42(1):132–40.
[241] Chen X, Ghate A, Tripathi A. Dynamic lot-sizing in sequential online retail auctions. Eur J Oper Res 2011;215(1):257–67.
[242] Schütz H-J, Kolisch R. Capacity allocation for demand of different customer-product-combinations with cancellations, no-shows, and overbooking when there is a sequential delivery of service. Annal Oper Res 2013;206(1):401–23.
[243] Herring W, Herrmann J. A stochastic dynamic program for the single-day surgery scheduling problem. IIE Trans Healthcare Syst Eng 2011;1(4):213–25.
[244] Krishnamurthy V. Algorithms for optimal scheduling and management of hidden markov model sensors. IEEE Trans Signal Process 2002;50(6):1382–97.
[245] Huang Q, Jia Q-S, Guan X. Robust scheduling of EV charging load with uncertain wind power integration. IEEE Trans Smart Grid 2018;9(2):1043–54.
[246] Minoux M. Robust and stochastic multistage optimisation under markovian uncertainty with applications to production/inventory problems. Int J Prod Res 2018;56(1–2):565–83.
[247] Dimitrov N, Dimitrov S, Chukova S. Robust decomposable markov decision processes motivated by allocating school budgets. Eur J Oper Res 2014;239(1):199–213.
[248] Ovacik I, Uzsoy R. Rolling horizon procedures for dynamic parallel machine scheduling with sequence-dependent setup times. Int J Prod Res 1995;33(11):3173–92.
[249] Chand S, Hsu V, Sethi S. Forecast, solution, and rolling horizons in operations management problems: a classified bibliography. Manuf Serv Oper Manag 2002;4(1):25–43.
[250] Clark A. Rolling horizon heuristics for production planning and set-up scheduling with backlogs and error-prone demand forecasts. Prod Plan Control 2005;16(1):81–97.
[251] Gunnarsson H, Rönnqvist M. Solving a multi-period supply chain problem for a pulp company using heuristics: an application to södra cell AB. Int J Prod Econ 2008;116(1):75–94.
[252] Balakrishnan J, Hung Cheng C. The dynamic plant layout problem: incorporating rolling horizons and forecast uncertainty. Omega 2009;37(1):165–77.
[253] Rakke J, Stalhane M, Moe C, Christiansen M, Andersson H, Fagerholt K, et al. A rolling horizon heuristic for creating a liquefied natural gas annual delivery program. Transp Res Part C 2011;19(5):896–911.
[254] Papadimitriou D, Fortz B. A rolling horizon heuristic for the multiperiod network design and routing problem. Networks 2015;66(4):364–79.
[255] Addis B, Carello G, Grosso A, Tànfani E. Operating room scheduling and rescheduling: a rolling horizon approach. Flexible Serv Manuf J 2016;28(1–2):206–32.
[256] Bindewald V., Dunke F., Nickel S. Research project of the German Research Foundation: Sequential Decision Making under System-inherent Uncertainty: Mathematical Optimization Methods for Time-dynamic Applications. http://dol.ior.kit.edu/english/Projects_DFG.php; 2018.
[257] Pflug G, Pichler A. Multistage stochastic optimization. Springer; 2014.

Please cite this article as: H. Bakker, F. Dunke and S. Nickel, A structuring review on multi-stage optimization under uncertainty: Aligning concepts from theory and practice, Omega, https://doi.org/10.1016/j.omega.2019.06.006