Citation
Yu, H. and Sun, J. and Wang, Y. 2021. A time-consistent Benders decomposition method for multistage distributionally robust stochastic optimization with a scenario tree structure. Computational Optimization and Applications. 79 (1): pp. 67-99. http://doi.org/10.1007/s10589-021-00266-7

Computational Optimization and Applications manuscript No.
(will be inserted by the editor)

A Time-Consistent Benders Decomposition Method for Multistage Distributionally Robust Stochastic Optimization with a Scenario Tree Structure

Haodong Yu · Jie Sun · Yanjun Wang

Received: date / Accepted: date

Abstract A computational method is developed for solving time consistent distributionally robust multistage stochastic linear programs with discrete distribution. The stochastic structure of the uncertain parameters is described by a scenario tree. At each node of this tree, an ambiguity set is defined by conditional moment constraints to guarantee time consistency. This method employs the idea of nested Benders decomposition that incorporates forward and backward steps. The backward steps solve some conic programming problems to approximate the cost-to-go function at each node, while the forward steps are used to generate additional trial points. A new framework of convergence analysis is developed to establish the global convergence of the approximation procedure, which does not depend on the assumption of polyhedral structure of the original problem. Numerical results of a practical inventory model are reported to demonstrate the effectiveness of the proposed method.

Keywords multistage stochastic programming · distributionally robust · scenario tree model · decomposition method

Mathematics Subject Classification (2000) MSC 90C15 · 90C47

This work is partially supported by Grants 11401384, B16002 and 11271243 of National Natural Science Foundation of China.

H. Yu
School of Statistics and Mathematics, Shanghai Lixin University of Accounting and Finance, PRC.
E-mail: nianchuixiao@msn.com

J. Sun (The corresponding author)
School of Science, Hebei University of Technology, PRC and School of Business, National University of Singapore, Singapore
E-mail: jie.sun@curtin.edu.au

Y. Wang
School of Mathematics, Shanghai University of Finance and Economics, PRC
E-mail: wangyj@mail.shufe.edu.cn

1 Introduction

Stochastic programming focuses on optimization models involving random factors. In many cases, the decision is naturally split into several stages in response to the realization of a random process. This leads to the concept of multistage stochastic programming. The general form of a T-stage stochastic programming problem is as follows:

$$\min_{x_1\in C_1} F_1(x_1)+\mathbb{E}\left[\inf_{x_2\in C_2(x_1,\xi_2)} F_2(x_2,\xi_2)+\mathbb{E}\Big[\cdots\mathbb{E}\big[\inf_{x_T\in C_T(x_{[T-1]},\xi_{[T]})} F_T(x_T,\xi_{[T]})\big]\Big]\right], \tag{1}$$

where $x_{[t]}=(x_1,...,x_t)$ and $x_t=x_t(\xi_{[t]})$ depend on the data process $\xi_{[t]}=(\xi_1,...,\xi_t)^\top$ up to time t (note that the first stage data ξ_1 is in fact deterministic, representing the given data before any decision is made). This pattern of solution is due to the nonanticipative nature of human decisions, where $C_1,...,C_T(x_{[T-1]},\xi_{[T]})$ are constraints parameterized by past decision $x_{[t-1]}$ and past random data $\xi_{[t]}, t=1,...,T$. See [1] for details on multistage stochastic programming.

In order to solve a stochastic optimization problem by computer, one often has to make simulation or discretization for the distribution of the involved random variables. It is therefore natural to assume that the random process $\xi_1,\cdots,\xi_T$ has a finite number of realizations, which leads to the scenario tree model, where the value of ξ_1 at stage 1 is the root node, and the realizations of ξ_t correspond to nodes of level t in this tree. For stage t satisfying $1<t<T$, each node has a parent node and some children nodes, corresponding to the realizations of its previous and next stages, respectively. In this way, all the possible realizations of the random process can be organized through a tree structure. The scenario tree formulation is a main model in multistage stochastic programming, see e.g. [2–4] for references.

We are concerned with a special linear case of the multistage programming problem (1) with a scenario tree structure as follows.

$$\min_{\substack{A_1x_1=b_1\\ x_1\geq 0}} c_1^\top x_1+\mathbb{E}\left[\min_{\substack{B_2x_1+A_2x_2=b_2\\ x_2\geq 0}} c_2^\top x_2+\mathbb{E}\Big[\cdots+\mathbb{E}\big[\min_{\substack{B_Tx_{T-1}+A_Tx_T=b_T\\ x_T\geq 0}} c_T^\top x_T\big]\Big]\right], \tag{2}$$

where the matrix-vector pair $(B_t,b_t)=(B_t(\xi_t),b_t(\xi_t))$, $t=2,\cdots,T$, while the vector c_t and the matrix $\{A_t\}$ are assumed to be deterministic.

In stochastic optimization, the distribution of the random variables is traditionally assumed to be known, e.g., see the recent paper [5]. Thus, in an abstract form, problem (1) and (2) can be written as

$$\min_{x\in X}\mathbb{E}_P[F(x,\xi)]$$

for a certain decision vector $x=(x_1,...,x_T)$ and a random vector $\xi=(\xi_1,...,\xi_T)$, where the probability measure P and function F are given. However, in many cases, we can only obtain limited information about the distribution of ξ.

This motivated the study on distributionally robust stochastic programming (DRSP), which is generally expressed as follows.

$$\min_{x\in X} \sup_{P\in\mathcal{P}} \mathbb{E}_P[F(x,\xi)], \tag{3}$$

where $\mathcal{P}$ is a set of probability measures of ξ, called the ambiguity set. The DRSP problem has attracted much interest during the recent two decades. See e.g., [6–9] for some progress of this method.

The mapping $F(x,\xi)$ in (3) is an abstract function. If $F(x,\xi)$ is not explicit and contains the optimal value function of a second-stage problem, then it is called a two-stage DRSP problem, i.e.,

$$\min_{x\in X} f(x) := F_1(x) + \sup_{P\in\mathcal{P}} \mathbb{E}_P[F_2(x,\xi)], \tag{4}$$

where $F_1(x)$ is its explicit part and $F_2(x,\xi)$ is the optimal value of a second-stage optimization problem for given x with ξ being a random vector. During the past several years, the two-stage DRSP has attracted much interests. In [10, 11, 13–15], Sun et al. studied the two-stage DRSP problem. Specifically, in [10, 11] they studied the two-stage linear DRSP problem with moment constraints and in [12–15] they discussed the DRSP problem with non-expectation risk measures, e.g., general coherent measures. On the other hand, many studies focus on the two-stage DRSP with different types of ambiguity sets. In [16], Jiang and Guan studied the risk-averse two-stage stochastic program with L_1 norm based ambiguity set. In [17], Hanasusanto and Kuhn considered the ambiguity set defined by the Wasserstein balls and showed that this model can be reformulated to a conic programming. Other studies considered the two-stage DRSP with binary variables such as [18] and [19].

Compared with the one and two stage models, there is very limited literature on the multistage-DRSP problems. The difficulty lies in two aspects.

The first aspect is the way to introduce the distributional ambiguity of the tree process. In stochastic programming, a traditional way is to use the nested distance, which is based on the nested distribution, to measure the difference between discrete time stochastic processes. There is a rich body of literatures on nested distances, which have been applied in scenario tree reduction techniques and some other areas. As to DRSP problems, Pflug et al. [20, 21] introduced the Wasserstein distance, one of the most popular nested distances, to construct the ambiguity set for the scenario tree model, and proposed a corresponding algorithm for the multistage-DRSP problem.

The nested distance is a powerful tool in scenario tree reduction problems and some other areas. In view of the dynamic nature of the decision procedure, however, the nested distance approach may cause some problems when the issue of "time consistency" comes into play.

Time consistency is a principle postulated on the risk measure or the multistage stochastic optimization problems to avoid the inconsistency in the decision chain. "The solution of the problem at time 0 consists of a complete plan for all future decisions at later times. If it turns out that it is preferable

to change the initial plan at later stages, then the decision problem is called inconsistent in time" ([1], Page 175). For deterministic problems, due to the Bellman's optimality principle, the inconsistency phenomenon will not occur. While for stochastic problems, some requirements may be imposed so that the the risk measures or the problems can be reformulated as a nested form, to which some decomposition methods can be applied. For some problems, such as the risk aversion problems, a natural approach is to introduce the concept of time consistent risk measure, which is based on the conditional measure or dynamical risk measure, see e.g., [22, 23] for reference. A complete review on the time consistent risk measure can be found in [24].

Another approach to time consistency is to directly deal with the so-called time-consistent decision problem, similar to the Bellman's principle for deterministic optimization. Shapiro [25] considered a tree-structured problem and expressed this approach as "At every state of the system, our optimal decisions should not depend on scenarios which we already know cannot happen in the future". In DRSP it means that, at each node, the decision maker only has to consider the distributional uncertainty of the descendent subtree, and should not take into account the scenarios which will not occur in that subtree. Hence the distributional robustness of the scenario tree should be specifically defined at every node. Other discussions on the time-consistent problem can be found in [26].

The second aspect of difficulty lies in the algorithm and its convergence. The nested Benders decomposition is one of the most important methods for multistage problems. The main idea is to use supporting hyperplanes to dynamically approximate the cost-to-go function at each node and to decompose the original problem to a series of smaller optimization problems. A comprehensive introduction on the nested Benders decomposition method is [27]. A recent review is [28]. Discussions on its acceleration techniques can be found in [29].

In stochastic programming, the nested Benders decomposition is further developed into a stochastic dual dynamic programming (SDDP) approach [30] by incorporating a forward step to generate a statistical upper bound. Some recent developments on this method include improvement on the cutting plane methods for large scale problems [31, 32], stochastic decomposition methods [33] and their convergence [34], Markov uncertainty [35], the construction of a deterministic upper bound [36] etc. In robust optimization, a similar method, called robust dual dynamic programming (RDDP) method, has been proposed by Georghiou et al. [37].

To establish the convergence analysis, the SDDP-type algorithms commonly require the objective function and the feasible set to be polyhedral, which guarantees the decomposition algorithms to achieve finite termination and facilitates the use of some vertex enumeration methods. An important exception is the work of [37], which considered the feasible region to be non-polyhedral and showed that the RDDP scheme asymptotically converges to an optimal solution of the generic multistage robust optimization problem.

The multistage DRSP problem is often more complicated than the multistage stochastic optimization problem. In particular, for the distributionally robust problem with moment uncertainty considered in this paper, the related sub-problems in each stage are conic programming problems. Thus, it calls for a new framework to analyze the convergence of multistage optimization problems without using the polyhedral property.

The contribution of this paper is two fold. First, a time-consistent Benders decomposition computational scheme is developed for the linear multistage distributionally robust scenario tree optimization (DRSTO) model. Differently from [20, 21], we do not introduce the nested distances. In view of the time consistency in Shapiro's sense [25], we decompose the distributional robustness to each node and consider the following multistage DRSP problems

$$\min_{\substack{A_1x_1=b_1\\x_1\geq 0}} c_1^\top x_1+\sup_{P_2\in\mathbb{P}_2}\mathbb{E}\Bigg[\min_{\substack{B_2x_1+A_2x_2=b_2\\x_2\geq 0}} c_2^\top x_2+\sup_{P_3\in\mathbb{P}_3}\mathbb{E}\Big[\cdots+\sup_{P_T\in\mathbb{P}_T}\mathbb{E}[\min_{\substack{B_Tx_{T-1}+A_Tx_T=b_T\\x_T\geq 0}} c_T^\top x_T]\Big]\Bigg], \tag{5}$$

where $\mathbb{P}_t, t=1,...,T$ are ambiguity sets defined by moment constraints, which will be made clear in next section.

In contrast to the nested distance approach, model (5) decomposes the distributional robustness into each decision stage. Since the decision process is completed stage by stage, for each node, it is natural to only consider the conditional distributionally robustness for its following scenarios, and exclude the scenarios which we are known not to happen in the future. This will guarantee time consistency for the solution to the multistage DRSP problems.

It is emphasized that the ambiguity sets $\mathbb{P}_t$ is defined by moment uncertainty. As is pointed out in [38], there are two typical ways to construct the uncertainty set. One is based on the moment constraints. Another is to introduce a reference probability measure (or called an "empirical distribution") and define a related distance or divergence with respect to it. The latter approach includes the Wasserstein distance and the $\phi-$divergence (e.g. the K-L divergence). In our opinion, the Wasserstein distance would have a difficulty in implementation of the time consistency principle. As to the $\phi-$divergence, a drawback is that the ambiguity set will exclude some important distributions. For example, if we use the K-L divergence, then the the distributions contained in the ambiguity set must be absolutely continuous with respect to the reference distribution. In particular, if the basis distribution p_0 is discrete, and has probability $p_0(x_0)=0$ at some point x_0, then any distribution p in the ambiguity set have to satisfy $p(x_0)=0$, i.e., x_0 will not be contained in the support of p. It is therefore reasonable to base our work on the moment constraints, which provides a wide range on the possible choice of the distributions.

By using the moment constraints, each sub-problem of (5) will be likely a conic programming problem, which requires substantial change in the convergence analysis. The second contribution of this paper is therefore to develop a new framework for the convergence analysis. The new framework is based on

parametric optimization and is hopefully applicable to more general convex functions.

The rest of this paper is organized as follows. In Section 2, we introduce the nested moment ambiguity and present the algorithm, which incorporates forward and backward steps. In Section 3, we analyze the convergence of the algorithm. We will show that, the approximated cost-to-go functions at every node will finally converge to their real counterparts. Furthermore, under mild assumptions, we prove that any limit point of the iteration sequence is a solution to the problem. In Section 4, we report results of numerical tests on a small inventory control problem to show the effectiveness of the proposed method. The paper is concluded in Section 5.

2 Problem Formulation and the Decomposition Method

In this section, we formulate the multistage linear DRSTO model and discuss the decomposition method.

Let $\mathcal{N}$ be the set of nodes of the scenario tree. For any $i \in \mathcal{N}$, let $t(i)$ be its stage level in the random process. Further let $a(i)$ be its unique ancestor node at stage $t(i)-1$. If i is the root node, then $t(i)=1$ and $a(i)=\emptyset$. Let $\mathcal{T}(i)$ be the set of its children nodes at stage $t(i)+1$. To unify the expression, in the case when $t(i)=T$, i.e., i is a leaf node, let $\mathcal{T}(i)=\{i\}$.

For any given $i \in \mathcal{N}$, let $n_i = |\mathcal{T}(i)|$ be the cardinality of $\mathcal{T}(i)$. Let p_{ij} be the transfer probability from i to $j \in \mathcal{T}(i)$, thus $(p_{ij_1}, \cdots, p_{ij_s})$ $(s = n_i; j_1, \cdots, j_s \in \mathcal{T}(i))$ constitutes a conditional probability distribution from node i to its next stage (For the leaf nodes, trivially, we have $p_{ii}=1$). In what follows, we denote such conditional distribution by $(p_{ij}), j \in \mathcal{T}(i)$ for short.

For each non-leaf node $i \in \mathcal{N}$, we define a conditional random variable $\eta_i = \xi_{t(i)+1} | \xi^{(i)}_{[t(i)]}$ (we use $\xi^{(i)}_{[t(i)]}$ to denote the history process associated to node i) with $\Omega_i = \{\eta_{ij} = \xi^{(j)}_{t(i)+1} | j \in \mathcal{T}(i)\}$ being its sample space, where $\xi^{(j)}_{t(i)+1}$ are the possible realizations of $\xi_{t(i)+1}$ for node i. If i is a leaf node, since $\mathcal{T}(i)=\{i\}$, we can define $\eta_i \equiv \xi^{(i)}_{t(i)}$, with $\Omega_i = \{\eta_{ii} = \xi^{(i)}_{t(i)}\}$.

Let μ_i and Σ_i be the estimator of the conditional expectation $\mathbb{E}[\eta_i]$ and the conditional covariance matrix $\mathbb{E}[(\eta_i - \mathbb{E}(\eta_i))(\eta_i - \mathbb{E}(\eta_i))^\top]$, respectively.

To construct the distributionally robust model, we use the following nested moment uncertainty to describe the ambiguity of distribution information with respect to η_i,

$$\begin{aligned} (\mathbb{E}[\eta_i] - \mu_i)^\top \Sigma_i^{-1} (\mathbb{E}[\eta_i] - \mu_i) &\le \gamma_1 \\ \mathbb{E}[(\eta_i - \mu_i)(\eta_i - \mu_i)^\top] &\preceq \gamma_2 \Sigma_i, \end{aligned} \tag{6}$$

where $\gamma_1 > 0, \gamma_2 \ge 1$, and $\Sigma_i \succ 0$ (as usual, $\succ 0$ and $\succeq 0$ mean positive definiteness and positive semidefiniteness, respectively). For discrete distribution,

the ambiguity set can be equivalently expressed as follows.

$$\mathbb{P}_i(\Omega_i, \mu_i, \Sigma_i, \gamma_1, \gamma_2) = \left\{ \sum_{j\in\mathcal{T}(i)} p_{ij} = 1, p_{ij} \geq 0 \;\middle|\; \begin{array}{l} \sum_{j\in\mathcal{T}(i)} p_{ij}[(\eta_{ij}-\mu_i)(\eta_{ij}-\mu_i)^\top] \preceq \gamma_2\Sigma_i \\ \sum_{j\in\mathcal{T}(i)} p_{ij} \begin{bmatrix} \Sigma_i & \eta_{ij}-\mu_i \\ (\eta_{ij}-\mu_i)^\top & \gamma_1 \end{bmatrix} \succeq 0 \end{array} \right\}. \quad (7)$$

The ambiguity set (7) used here was first introduced in [6]. As is well known, moment uncertainty is a classic approach to construct the ambiguity set because this formulation has clear statistical meaning and it facilitates the reformulation of the distributionally robust problem to conic optimization problems, to which some well-developed algorithms can be applied.

We first consider the extreme case of the distributionally robustness. The next proposition shows that if the parameters γ_1 and γ_2 are sufficiently large, the distributionally robust constraint will cover the entire space of probability distributions.

Proposition 1 *If $\Sigma_i \succ 0$, then for sufficiently large $\gamma_1 > 0$ and $\gamma_2 \geq 1$, the ambiguity set $\mathbb{P}_i(\Omega_i, \mu_i, \Sigma_i, \gamma_1, \gamma_2)$ can be described as*

$$\tilde{\mathbb{P}}_i(\Omega_i) = \left\{ p = (p_{ij}), j \in \mathcal{T}(i) \;\middle|\; \sum_{j\in\mathcal{T}(i)} p_{ij} = 1, p_{ij} \geq 0 \right\}. \quad (8)$$

Proof It is sufficient to prove that, for any γ_1 and γ_2 large enough, any distribution in (8) satisfies the robust constraint in (7). Firstly, for γ_1 sufficiently large,

$$(\eta_{ij}-\mu_i)^\top \Sigma_i^{-1}(\eta_{ij}-\mu_i) \leq \gamma_1 \quad \text{for all } j \in \mathcal{T}(i).$$

Together with the positive definiteness of Σ_i, we have

$$\sum_{j\in\mathcal{T}(i)} p_{ij} \begin{bmatrix} \Sigma_i & \eta_{ij}-\mu_i \\ (\eta_{ij}-\mu_i)^\top & \gamma_1 \end{bmatrix} \succeq 0 \quad (9)$$

holds for any distribution (p_{ij}) satisfying (8).

Furthermore, for any unit vector x satisfying $\|x\|_2 = 1$, it holds that for any $j \in \mathcal{T}(i)$

$$x^\top[(\eta_{ij}-\mu_i)(\eta_{ij}-\mu_i)^\top]x \leq \|\eta_{ij}-\mu_i\|_2^2 \quad (10)$$

and

$$\gamma_2 x^\top \Sigma_i x \geq \gamma_2 \lambda_{\min}(\Sigma_i), \quad (11)$$

where $\lambda_{\min}(\Sigma_i) > 0$ is the least eigenvalue of Σ_i. Consequently, for γ_2 large enough,

$$x^\top \left[\gamma_2\Sigma_i - (\eta_{ij}-\mu_i)(\eta_{ij}-\mu_i)^\top\right] x \geq 0$$

which means

$$\gamma_2\Sigma_i \succeq (\eta_{ij}-\mu_i)(\eta_{ij}-\mu_i)^\top \quad \text{for all } j \in \mathcal{T}(i). \quad (12)$$

It follows immediately that $\sum_{j\in\mathcal{T}(i)} p_{ij}[(\eta_{ij}-\mu_i)(\eta_{ij}-\mu_i)^\top] \preceq \gamma_2\Sigma_i$. □

Next, we specialize some notations in (5) to the ones in a scenario tree model. Let us designate $A_i = A_{t(i)}, c_i = c_{t(i)}$, and (B_i, b_i) be the realization of $(B_{t(i)}, b_{t(i)})$ corresponding to node i, respectively. The multi-stage distributionally robust model (5) can then be reformulated as the next dynamic programming equations

$$Q_i(x_{a(i)}) = \min_{x_i} \left\{ c_i^\top x_i + \mathcal{Q}_i(x_i) : B_i x_{a(i)} + A_i x_i = b_i, x_i \geq 0 \right\}. \tag{13}$$

Here $\mathcal{Q}_i(x_i)$ is called a cost-to-go function whose definition is as follows. If i is not a leaf node, let

$$\mathcal{Q}_i(x_i) := \sup_{P_i \in \mathbb{P}_i} \mathbb{E}\left[Q_j(x_i)\right] = \sup_{P_i \in \mathbb{P}_i} \sum_{j \in \mathcal{T}(i)} p_{ij} Q_j(x_i). \tag{14}$$

If i is a leaf node, then let $\mathcal{Q}_i(\cdot) \equiv 0$. In the case when $a(i) = \emptyset$, i.e., i is the root node, we trivially define $x_\emptyset = 0, B_i = 0$.

Also notice that, as is shown by Proposition 1, in the limit case when γ_1 and γ_2 are large enough, the ambiguity set is the entire space of probability distributions. In this case the cost-to-go function will be

$$\mathcal{Q}_i(x_i) = \max_{j \in \mathcal{T}(i)} Q_j(x_i), \tag{15}$$

which means that the worst case distribution is the Dirac measure δ_j (the measure of mass one at the jth sub-branch from i). This case is equivalent to a multistage robust programming problem without consideration of the scenario distribution. Overall, the decision process will reduce to a single path of the scenario tree. That is, when an optimal solution is decided, there exists a path such that the total cost of all the nodes in this path will be equal to the optimal value of the first stage problem.

To analyze the convexity of $\mathcal{Q}_i(\cdot)$, we first cite the next result.

Lemma 2 *([39]) Let X and U be real linear spaces, and suppose $F : X \times U \to [-\infty, +\infty]$ is jointly convex in (x, u). Then the following optimal value function $\varphi(\cdot)$ is also convex.*

$$\varphi(u) := \inf_{x \in X} F(x, u) \qquad u \in U. \tag{16}$$

Lemma 3 *The function $Q_i(\cdot)$ and $\mathcal{Q}_i(\cdot)$ defined as (13) and (14) respectively are both convex.*

Proof The proof is by induction. Firstly, if i is a leaf node, $\mathcal{Q}_i(\cdot) \equiv 0$ and $Q_i(x_{a(i)})$ is the optimal value of a linear programming problem with $x_{a(i)}$ be its parameter. It follows from the Proposition 2.1 in [40] that $Q_i(\cdot)$ is convex, which implies that $\mathcal{Q}_{a(i)}(\cdot)$ is convex.

Generally, suppose that i is not a leaf node and that for all $j \in \mathcal{T}(i)$, $\mathcal{Q}_j(\cdot)$ are convex functions. Denote $C = \{(x,u)|A_i x + B_i u = b_i, x \geq 0\}$, the indicator ψ_C of C as follows:

$$\psi_C(x,u) = \begin{cases} 0 & \text{if } (x,u) \in C, \\ +\infty & \text{if } (x,u) \notin C. \end{cases} \tag{17}$$

Denote

$$F_j(x,u) = c_j^\top x + \mathcal{Q}_j(x) + \psi_C(x,u),$$

which is convex for (x,u). It follows from Lemma 2 and (13) that $Q_j(\cdot)$ is convex, which implies the convexity of $\mathcal{Q}_i(\cdot)$. □

We now turn to discuss the equivalent form of $\mathcal{Q}_i(x_i)$ by Lagrange duality. For this purpose, we need the next assumption.

Assumption 4 *(Slater's condition) Suppose that for any node $i \in \mathcal{N}$, there exists a distribution $(p_{ij_1}, \cdots, p_{ij_s}) \in \mathbb{P}_i(\Omega_i, \mu_i, \gamma_1, \gamma_2)$ $(s = n_i; j_1, \cdots, j_s \in \mathcal{T}(i))$ such that the probability $p_{ij_1}, \cdots, p_{ij_s}$ are all positive and*

$$\sum_{j \in \mathcal{T}(i)} p_{ij}\eta_{ij} = \mu_i, \qquad \sum_{j \in \mathcal{T}(i)} p_{ij}\left[(\eta_{ij} - \mu_i)(\eta_{ij} - \mu_i)^\top\right] \prec \Sigma_i. \tag{18}$$

Lemma 5 *Suppose $\gamma_1 > 0$, $\gamma_2 \geq 1$, $\Sigma_i \succ 0$ and Assumption 4 holds. Then $\mathcal{Q}_i(x_i)$ equals to the optimal value of the following problem*

$$\begin{aligned} \min_{S,q,r,v} \ & r + v \\ \text{s.t.} \ & r + \eta_{ij}^\top S \eta_{ij} + \eta_{ij}^\top q \geq Q_j(x_i), \quad (j \in \mathcal{T}(i)) \\ & v \geq (\gamma_2 \Sigma_i + \mu_i \mu_i^\top) \bullet S + \mu_i^\top q + \sqrt{\gamma_1}\|\Sigma_i^{1/2}(q + 2S\mu_i)\| \\ & S \succeq 0. \end{aligned} \tag{19}$$

Proof By Lemma 1 of [6], we readily have that (19) is the Lagrange dual problem of (14). Furthermore, Assumption 4 ensures that condition (2.20) required by Proposition 2.8(ii) of [41] holds. Specifically, since (18) holds and $\gamma_2 \geq 1$, one has

$$\sum_{j \in \mathcal{T}(i)} p_{ij}[(\eta_{ij} - \mu_i)(\eta_{ij} - \mu_i)^\top] \prec \Sigma_i \preceq \gamma_2 \Sigma_i \tag{20}$$

and

$$\sum_{j \in \mathcal{T}(i)} p_{ij} \begin{bmatrix} \Sigma_i & \eta_{ij} - \mu_i \\ (\eta_{ij} - \mu_i)^\top & \gamma_1 \end{bmatrix} = \begin{bmatrix} \Sigma_i & 0 \\ 0 & \gamma_1 \end{bmatrix} \succ 0. \tag{21}$$

On the other hand, problem (14) can be reformulated as

$$\min_{p \in \mathcal{C}} \langle c, p \rangle \quad \text{s.t.} \ \ A(p) + b \in K, \tag{22}$$

where $\mathcal{C} := \{p = (p_{ij}), j \in \mathcal{T}(i) \,|p_{ij} \geq 0\}$, $c = (Q_j(x_i)), j \in \mathcal{T}(i)$. A is a linear mapping defined as follows

$$A(p) := \sum_{j \in \mathcal{T}(i)} \begin{pmatrix} p_{ij} \\ -p_{ij}[(\eta_{ij} - \mu_i)(\eta_{ij} - \mu_i)^\top] \\ p_{ij} \begin{bmatrix} \Sigma_i & \eta_{ij} - \mu_i \\ (\eta_{ij} - \mu_i)^\top & \gamma_1 \end{bmatrix} \end{pmatrix}. \tag{23}$$

$b = \{-1\} \times \{\gamma_2 \Sigma_i\} \times \{0_{(m_i+1)\times(m_i+1)}\}$ and $K = \{0\} \times \mathbb{S}_+^{m_i} \times \mathbb{S}_+^{m_i+1}$, where m_i is the length of the random vector η_i, $\mathbb{S}_+^{m_i}$ and $\mathbb{S}_+^{m_i+1}$ are the cones of positive semidefinite matrices of dimension m_i and $m_i + 1$ respectively. By (20) and (21), we readily have that

$$-b \in \text{int}[A(\mathcal{C}) - K]. \tag{24}$$

That is, condition (2.20) in [41] holds. This implies that there is no duality gap between (19) and its primal problem. Hence the lemma holds. □

Based on the convexity of $\mathcal{Q}_i(\cdot)$, the main idea of the nested Benders decomposition method is to use a series of cutting planes to approximate $\mathcal{Q}_i(\cdot)$. The definitions of a cutting plane and a supporting plane of a given function are as follows.

Definition 6 *([42]) Let $\mathcal{Q}(\cdot) : \mathbb{R}^n \to \mathbb{R}$ be a convex function. An affine function $l(x)$ is called a cutting plane of $\mathcal{Q}(\cdot)$ if $\mathcal{Q}(x) \geq l(x)$ for all $x \in \mathbb{R}^n$. Furthermore, if $\mathcal{Q}(x_0) = l(x_0)$ for some x_0, then $l(x)$ is said to be a supporting plane of $\mathcal{Q}(x)$.*

In what follows, we illustrate the approach to constructing cutting planes that approximate the cost-to-go functions $\mathcal{Q}_i(x_i)$, which paves the way to solve problem (5). As mentioned in Section 1, the construction is realized through a forward step and a backward step.

Let $\tilde{\mathcal{Q}}_i(x_i)$ be an approximation of $\mathcal{Q}_i(x_i)$, i.e.,

$$\tilde{\mathcal{Q}}_i(x_i) = \max_{1 \leq k \leq K_i} l_k(x_i) := \max_{1 \leq k \leq K_i} \alpha_k^\top x_i + \beta_k, \tag{25}$$

where $l_k(x_i) = \alpha_k^\top x_i + \beta_k$ $(k = 1, \cdots, K_i)$ are cutting planes of $\mathcal{Q}_i(x_i)$.

Substitute $\mathcal{Q}_i(x_i)$ by $\tilde{\mathcal{Q}}_i(x_i)$ in (13), we can define the following functions (when i is not a leaf node),

$$\begin{aligned} \hat{Q}_i(x_{a(i)}) := \min_{x_i, h_i} \; & c_i^\top x_i + h_i \\ \text{s.t.} \; & B_i x_{a(i)} + A_i x_i = b_i, x_i \geq 0, \\ & h_i \geq \alpha_k^\top x_i + \beta_k \quad (k = 1, \cdots, K_i). \end{aligned} \tag{26}$$

Correspondingly, we define

$$\hat{\mathcal{Q}}_i(x_i) := \sup_{P_i \in \mathbb{P}_i} \mathbb{E}\left[\hat{Q}_j(x_i)\right] = \sup_{P_i \in \mathbb{P}_i} \sum_{j \in \mathcal{T}(i)} p_{ij} \hat{Q}_j(x_i). \tag{27}$$

$\hat{\mathcal{Q}}_i(\cdot)$ is a counterpart of the real $\mathcal{Q}_i(\cdot)$ when each $Q_j(\cdot)$ is replaced by $\hat{Q}_j(\cdot)$ ($j \in \mathcal{T}(i)$). Later, we will see that $\tilde{\mathcal{Q}}_i(x_i)$ and $\hat{\mathcal{Q}}_i(x_i)$ are used in the forward and backward step respectively. Furthermore, since $\tilde{\mathcal{Q}}_i(x_i) \leq \mathcal{Q}_i(x_i)$, one has $\hat{Q}_i(x_{a(i)}) \leq Q_i(x_{a(i)})$ which implies that $\hat{\mathcal{Q}}_i(x_i) \leq \mathcal{Q}_i(x_i)$. Hence, any cutting plane of $\hat{\mathcal{Q}}_i(\cdot)$ is also a cutting plane of $\mathcal{Q}_i(\cdot)$. The strict analysis will be given in Proposition 12.

Proposition 7 *Suppose $\gamma_1 \geq 0$, $\gamma_2 \geq 1$, $\Sigma_i \succ 0$, and Assumption 4 holds. Let $\Omega_i = \{\eta_{ij} | j \in \mathcal{T}(i)\}$ be the sample space of η_i. Then for any given $\bar{x}_i$, $\hat{\mathcal{Q}}_i(\bar{x}_i)$ equals to the optimal value of the following problem*

$$\begin{aligned} \min_{\substack{S,q,r,v \\ x_j,h_j}} \ & r + v \\ \text{s.t.} \ & r + \eta_{ij}^\top S \eta_{ij} + \eta_{ij}^\top q \geq c_j^\top x_j + h_j, \\ & v \geq (\gamma_2 \Sigma_i + \mu_i \mu_i^\top) \bullet S + \mu_i^\top q + \sqrt{\gamma_1} \|\Sigma_i^{1/2}(q + 2S\mu_i)\|, \\ & h_j \geq \max_{k=1,\cdots,K_j} \alpha_{jk}^\top x_j + \beta_{jk}, \\ & B_j \bar{x}_i + A_j x_j = b_j, x_j \geq 0, \\ & S \succeq 0 \qquad (j \in \mathcal{T}(i)). \end{aligned} \tag{28}$$

Proof Similar to the proof of Lemma 5, by (27), $\hat{\mathcal{Q}}_i(\bar{x}_i)$ equals to the optimal value of the following problem

$$\begin{aligned} \min_{S,q,r,v} \ & r + v \\ \text{s.t.} \ & r + \eta_{ij}^\top S \eta_{ij} + \eta_{ij}^\top q \geq \hat{Q}_j(\bar{x}_i), \quad (j \in \mathcal{T}(i)) \\ & v \geq (\gamma_2 \Sigma_i + \mu_i \mu_i^\top) \bullet S + \mu_i^\top q + \sqrt{\gamma_1} \|\Sigma_i^{1/2}(q + 2S\mu_i)\|, \\ & S \succeq 0. \end{aligned} \tag{29}$$

The proposition follows by replacing $\hat{Q}_j(\bar{x}_i)$ with (26) in the above expression. □

Lemma 8 *Suppose $\gamma_1 \geq 0$, $\gamma_2 \geq 1$, $\Sigma_i \succ 0$, and Assumption 4 holds, for given $\bar{x}_i$, further suppose that for any $j \in \mathcal{T}(i)$, there exists $x_j > 0$ such that $B_j \bar{x}_i + A_j x_j = b_j$. Then $\hat{\mathcal{Q}}_i(\bar{x}_i)$ equals to the optimal value of the following problem*

$$\begin{aligned} \max_{l_j, \lambda_j, d_{jk}} \quad & \sum_{j \in \mathcal{T}(i)} \sum_{k=1}^{K_j} d_{jk} \beta_k - \Sigma_j l_j^\top (B_j \bar{x}_i - b_j) \\ \text{s.t.} \ & \sum_{j \in \mathcal{T}(i)} \sum_{k=1}^{K_j} d_{jk} = 1, \sum_{k=1}^{K_j} d_{jk} = \lambda_j, d_{jk} \geq 0, \\ & A_j^\top l_j \leq \lambda_j c_j + \sum_{k=1}^{K_j} d_{jk} \alpha_{jk}, \\ & \sum_{j \in \mathcal{T}(i)} \lambda_j [(\eta_{ij} - \mu_i)(\eta_{ij} - \mu_i)^\top] \preceq \gamma_2 \Sigma_i, \\ & \sum_{j \in \mathcal{T}(i)} \lambda_j \begin{bmatrix} \Sigma_j & \eta_{ij} - \mu_i \\ (\eta_{ij} - \mu_i)^\top & \gamma_1 \end{bmatrix} \succeq 0. \end{aligned} \tag{30}$$

Proof As is shown by Proposition 7 that, under the stated condition, $\hat{\mathcal{Q}}_i(\bar{x}_i)$ equals to the optimal value of (28). On the other hand, by Lemma 1 of [6], it holds that (28) can be rewritten as the following semidefinite programming problem

$$\begin{aligned}
\min_{\substack{r,P,p,S,s\\ x_j,h_j}} \quad & (\gamma_2\Sigma_i - \mu_i\mu_i^\top) \bullet S + r + (\Sigma_i \bullet P) - 2\mu_i^\top p + \gamma_1 s \qquad (31)\\
\text{s.t.} \quad & r + \eta_{ij}^\top S\eta_{ij} - 2\eta_{ij}^\top(p + S\mu_i) \geq c_j^\top x_j + h_j,\\
& h_j \geq \alpha_{jk}^\top x_j + \beta_{jk},\\
& B_j\bar{x}_i + A_j x_j = b_j,\\
& \begin{bmatrix} P & p \\ p^\top & s \end{bmatrix} \succeq 0,\\
& S \succeq 0, x_j \geq 0 \quad (j \in \mathcal{T}(i), k = 1, \cdots, K_j).
\end{aligned}$$

By formulating the Lagrangian of (31), it can be verified that (30) is the dual problem of (31). Since there exists $x_j > 0$ ($j \in \mathcal{T}(i)$) satisfying that $B_j\bar{x}_i + A_j x_j = b_j$, there exists $(r, P, p, S, s, x_j, h_j)$ such that

$$\begin{aligned}
& r + \eta_{ij}^\top S\eta_{ij} - 2\eta_{ij}^\top(p + S\mu_i) > c_j^\top x_j + h_j, \qquad (32)\\
& h_j > \alpha_{jk}^\top x_j + \beta_{jk},\\
& B_j\bar{x}_i + A_j x_j = b_j,\\
& S \succ 0, x_j > 0,\\
& \begin{bmatrix} P & p \\ p^\top & s \end{bmatrix} \succ 0 \quad (j \in \mathcal{T}(i), k = 1, \cdots, K_j),
\end{aligned}$$

which means $(r, P, p, S, s, x_j, h_j)$ satisfies the Slater condition of (31). Therefore, there is no duality gap between (30) and (31). □

Remarks.
(1) The above proof points out the strong duality relationship between (30) and (31), which is important in the analysis of the subgradient of $\hat{\mathcal{Q}}_i(\cdot)$.
(2) An alternative approach to verifying Lemma 8 is that, for given $\bar{x}_i$ and $P_i \in \mathbb{P}_i$, we can rewrite $\mathbb{E}\left[\hat{Q}_j(\bar{x}_i)\right]$ as follows.

$$\begin{aligned}
\min_{x_j,h_j} \quad & \sum_{j\in\mathcal{T}(i)} p_{ij}(c_j^\top x_j + h_j) \qquad (33)\\
\text{s.t.} \quad & B_j\bar{x}_i + A_j x_j = b_j\\
& h_j \geq \alpha_{jk}^\top x_j + \beta_{jk}\\
& x_j \geq 0 \quad (j \in \mathcal{T}(i), k = 1, \cdots, K_j),
\end{aligned}$$

which has the following dual problem

$$\begin{aligned} \max_{l_j,d_{jk}} & \sum_{j\in\mathcal{T}(i)}\sum_{k=1}^{K_j} d_{jk}\beta_{jk} - \sum_{j\in\mathcal{T}(i)} l_j^\top(B_j\bar{x}_i - b_j) \\ \text{s.t.} & \sum_{k=1}^{K_j} d_{jk} = p_{ij}, d_{jk} \geq 0 \\ & A_j^\top l_j \leq p_{ij}c_j + \sum_{k=1}^{K_j} d_{jk}\alpha_{jk} \quad (j\in\mathcal{T}(i), k=1,\cdots,K_j). \end{aligned} \tag{34}$$

By the assumption of Lemma 8, (33) has a strictly feasible solution, hence there is no duality gap. Thus, $\mathbb{E}\left[\hat{Q}_j(x_i)\right]$ is the optimal value of (34). Together with the definition of $\mathbb{P}_i$, we obtain that (30) is the equivalent form of (27).

Theorem 9 *For given $i \in \mathcal{N}$ such that neither $\{i\}$ nor $\mathcal{T}(i)$ contains leaf nodes, and for fixed $\bar{x}_i$, let l_j, λ_j, d_{jk}, $(j\in\mathcal{T}(i), k=1,\cdots,K_j)$ be any solution of (30). Then*

$$-\sum_{j\in\mathcal{T}(i)} B_j^\top l_j \in \partial\hat{\mathcal{Q}}_i(\bar{x}_i). \tag{35}$$

Proof Denote $Z := (W, Y, v)$, where $W \in \mathbb{R}^{|\mathcal{T}(i)|}$ consists of scalars w_j, $v \in \mathbb{R}^{|\mathcal{T}(i)|}$ denotes the set of v_j, and Y consists of scalars y_{jk}, where $j\in\mathcal{T}(i), k=1,\cdots,K_j$. Consider the following parameterized problem of (31)

$$\begin{aligned} \min_{\substack{r,P,p,S,s\\ x_j,h_j}} & (\gamma_2\Sigma_i - \mu_i\mu_i^\top)\bullet S + r + (\Sigma_i\bullet P) - 2\mu_i^\top p + \gamma_1 s \\ \text{s.t.} & \ r + \eta_{ij}^\top S\eta_{ij} - 2\eta_{ij}^\top(p + S\mu_i) \geq c_j^\top x_j + h_j + w_j \\ & h_j \geq \alpha_{jk}^\top x_j + y_{jk} \\ & A_j x_j = v_j \\ & \begin{bmatrix} P & p \\ p^\top & s \end{bmatrix} \succeq 0 \\ & S\succeq 0, x_j \geq 0 \quad (j\in\mathcal{T}(i), k=1,\cdots,K_j). \end{aligned} \tag{36}$$

Note that if $w_j = 0$, $y_{jk} = \beta_k$ and $v_j = b_j - B_j\bar{x}_i$, then (36) is the original problem (31). Let $\varphi(Z) := \varphi(W, Y, v)$ be the optimal value of (36).

Denote $L := (\lambda, D, l)$, where $\lambda \in \mathbb{R}^{|\mathcal{T}(i)|}$ consists of scalars λ_j, D contains d_{jk}, and $L \in \mathbb{R}^{|\mathcal{T}(i)|}$ consists of scalars l_j, with $j\in\mathcal{T}(i), k=1,\cdots,K_j$. The Lagrangian dual of (36) can be written as

$$\begin{aligned} \varphi(Z) := \max_{l_j,\lambda_j,d_{jk}} & \langle Z, L\rangle = \textstyle\sum_j \lambda_j w_j + \sum_j\sum_k d_{jk}y_{jk} + \Sigma_j l_j^\top v_j \\ \text{s.t.} & \quad L = (\lambda, D, l) \in \Pi, \end{aligned} \tag{37}$$

where

$$\Pi = \left\{ (\lambda, D, l) \left| \begin{array}{c} \sum_j \sum_k d_{jk} = 1, \sum_k d_{jk} = \lambda_j, d_{jk} \geq 0 \\ A_j^\top l_j \leq \lambda_j c_j + \sum_k d_{jk} \alpha_{jk} \\ \Sigma_{j \in \mathcal{T}(i)} \lambda_j [(\eta_{ij} - \mu_i)(\eta_{ij} - \mu_i)^\top] \preceq \gamma_2 \Sigma_i \\ \Sigma_{j \in \mathcal{T}(i)} \lambda_j \begin{bmatrix} \Sigma_j & \eta_{ij} - \mu_i \\ (\eta_{ij} - \mu_i)^\top & \gamma_1 \end{bmatrix} \succeq 0 \end{array} \right. \right\}. \tag{38}$$

Therefore,

$$\varphi(Z) = \sup_L \{\langle Z, L \rangle - I_\Pi(L)\}, \tag{39}$$

where $I_\Pi(\cdot)$ is the indicator function

$$I_\Pi(L) = \begin{cases} 0 & \text{if } L \in \Pi, \\ +\infty & \text{if } L \notin \Pi. \end{cases} \tag{40}$$

This means $\varphi(\cdot)$ is the conjugate of $I_\Pi(\cdot)$. Since $I_\Pi(\cdot)$ is a convex and lower semicontinuous function, $I_\Pi(\cdot)$ is also the conjugate of $\varphi(\cdot)$, and

$$\partial\varphi(Z) = \arg\max_L \langle Z, L \rangle - I_\Pi(L) = \arg\max_{L \in \Pi} \langle Z, L \rangle. \tag{41}$$

Let $Z_0 := (W_0, Y_0, v_0)$, where $W_0 = 0$, $y_{jk}^0 = \beta_{jk}$ and $v_j^0 = b_j - B_j \bar{x}_i$, $L_0 = (\lambda_0, D_0, l_0) = \arg\max_{L \in \Pi} < Z_0, L >$. By the chain rule of subdifferentiation, we have that

$$\partial \hat{\mathcal{Q}}_i(\bar{x}_i) = \nabla(Z_0(\bar{x}_i))^\top \partial\varphi(Z_0) = \{-\sum_j B_j^\top l_j\}. \qquad \square$$

For the case when $t(i) = T - 1$, i.e., i is the ancestor of some leaf nodes, we have the following result.

Theorem 10 *For given $i \in \mathcal{N}$ such that $t(i) = T - 1$ (i.e., the successors of node i are all leaf nodes) and given $\bar{x}_i$, let l_j, λ_j $(j \in \mathcal{T}(i))$ be any solution of the next problem*

$$\begin{aligned} \max_{l_j, \lambda_j} \; & -\sum_j l_j^\top (B_j \bar{x}_i - b_j) \\ \text{s.t.} \quad & A_j^\top l_j \leq \lambda_j c_j, \\ & \sum_{j \in \mathcal{T}(i)} \lambda_j [(\eta_{ij} - \mu_i)(\eta_{ij} - \mu_i)^\top] \preceq \gamma_2 \Sigma_i, \\ & \sum_{j \in \mathcal{T}(i)} \lambda_j \begin{bmatrix} \Sigma_j & \eta_{ij} - \mu_i \\ (\eta_{ij} - \mu_i)^\top & \gamma_1 \end{bmatrix} \succeq 0. \end{aligned} \tag{42}$$

Then

$$-\sum_j B_j^\top l_j \in \partial \mathcal{Q}_i(\bar{x}_i). \tag{43}$$

Proof Similar to Proposition 7, it holds that for given $\bar{x}_i$, $\mathcal{Q}_i(\bar{x}_i)$ equals to the optimal value of the following problem

$$\begin{aligned} \min_{\substack{S,q,r,v \\ x_j}} \ & r+v \qquad (44)\\ \text{s.t.} \ & r+\eta_{ij}^\top S\eta_{ij}+\eta_{ij}^\top q \geq c_j^\top x_j, \\ & v \geq (\gamma_2\Sigma_i+\mu_i\mu_i^\top)\bullet S+\mu_i^\top q+\sqrt{\gamma_1}\|\Sigma_i^{1/2}(q+2S\mu_i)\|, \\ & S\succeq 0, \\ & B_j\bar{x}_i+A_jx_j=b_j, x_j\geq 0 \quad (j\in\mathcal{T}(i)). \end{aligned}$$

It can be verified that (42) is the dual of (44). The rest of the proof is similar to that of Theorem 9. □

We are now ready to present the proposed decomposition method.

Algorithm 1 The Robust Decomposition Algorithm

Initialization: *Let $k=1$, for each node $i\in\mathcal{N}$, set $K_i=0$ and $B_{i_0}=0$. Denote i_0 be the root node and let $x_\emptyset=0$. Choose the tolerance $\epsilon>0$.*

Forward Step: *Starts from the root node. For each node $i\in\mathcal{N}$, retrieve $x_{a(i)}^k$ from its ancestor node. If $k=1$, find x_i^k be any feasible point of (13). If $k>1$, for each non-leaf node i, solve (26) to obtain current approximate solution x_i^k, and for each leaf node i, solve (13) to get current solution x_i^k.*

Stop Criteria: *If $\|x_i^k-x_i^{k-1}\|\leq\epsilon$ for each node i, then terminates. Otherwise, go on to carry out the backward step.*

Backward Step: *First choose $i\in\mathcal{N}$ whose successors are all leaf nodes, compute $\mathcal{Q}_i(x_i^k)$ by solving (44) and compute a g_i^k of $\partial\mathcal{Q}_i(x_i^k)$ by (42) and (43). Update $\tilde{\mathcal{Q}}_i^{k+1}(\cdot)$ and $\hat{Q}_i^{k+1}(\cdot)$ by adding the cutting plane $l_k(x)=\mathcal{Q}_i(x_i^k)+(g_i^k)^\top(x-x_i^k)$ into (25) and (26), respectively.*

By recursion, for $i\in\mathcal{N}$ whose successors have all been renewed in current iteration, use current x_i^k to compute $\hat{\mathcal{Q}}_i^{k+1}(x_i^k)$ by solving (30) and also compute a g_i^k of $\partial\hat{\mathcal{Q}}_i^{k+1}(x_i^k)$ by (35). Update $\tilde{\mathcal{Q}}_i^{k+1}(\cdot)$ and $\hat{Q}_i^{k+1}(\cdot)$ by adding the cutting plane $l_k(x)=\hat{\mathcal{Q}}_i^{k+1}(x_i^k)+(g_i^k)^\top(x-x_i)$ into (25) and (26), respectively.

Update: *Set $k=k+1$, and for each node i, $K_i=K_i+1$. Return to the Forward Step. (k is the number of iteration. K_i is the number of cutting planes for node i.)*

The outline of the iteration process is shown by Figure 1 and Figure 2. At each iteration, a forward step, which starts from the root node, is first carried out to generate new trial points for each node. The trial points are generated by solving the approximated problems. Then, a backward step, which starts from the leaf nodes and ends at the root node, is conducted to compute new cutting planes to update the cost-to-go functions and the approximated problems for each node.

Fig. 1 The Forward Step

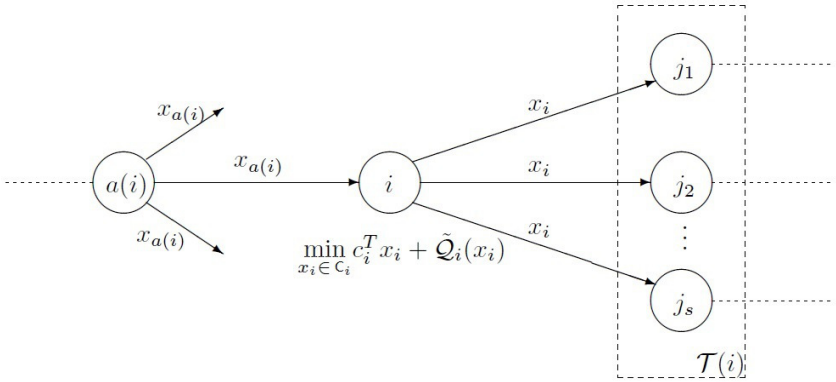
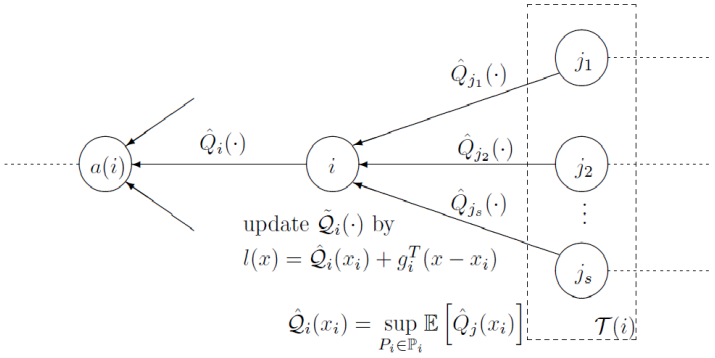


Fig. 2 The Backward Step



From the perspective of computation, there are some potential approaches to improve the numerical efficiency of the decomposition algorithm. For instance, in SDDP, a statistical upper bound is incorporated in the forward step, which enables the algorithm to be applicable for large scale problems. These techniques can not be directly applied to the distributionally robust problems, the main reason is: the computation of the statistical upper bound relies on the exact knowledge of the distribution of the parameters. A possible approach to overcome this difficulty is to construct a deterministic upper bound by using the convex hull of sample points. Literature on this topic includes [36], [43] etc.

Another problem is, with the increase of the number of the cutting planes, the number of the constraints will also increase. This will finally lead to a large scale conic program for each node. An approach to deal with this is to test

and delete the redundant cutting planes at each iteration. See e.g., [44] and [45] for references.

The usage of the above techniques needs specific discussions, and the convergence analysis will be much more complicated as well. Hence in the following sections, we still focus on the convergence properties of Algorithm 1, which are established through a new framework.

Assumption 11 *Suppose that for each node $i \in \mathcal{N}$, and any $x_{a(i)} \geq 0$ it holds that*
(i) *the feasible set $C_i(x_{a(i)}) = \{x_i \geq 0 | B_i x_{a(i)} + A_i x_i = b_i\}$ of (13) has nonempty relative interior;*
(ii) *$Q_i(x_{a(i)})$ and its approximation $\hat{Q}_i(x_{a(i)})$ are proper functions; and*
(iii) *there exists $\alpha \in \mathbb{R}$ and a bounded set C such that for every $\tilde{x}_{a(i)} \geq 0$ in a neighborhood of $x_{a(i)}$, the level set*

$$\mathrm{lev}_\alpha \hat{Q}_j^k := \left\{ x_i \in C_i(\tilde{x}_{a(i)}) : \hat{Q}_j^k(x_i) \leq \alpha \right\}, \quad j \in \mathcal{T}(i)$$

are all contained in C.

Remarks.

(i) The assumption on the feasible set guarantees that the proposed algorithm is always well defined. In practice, in the case where the feasible sets $C_i(x_{a(i)})$ are empty, a feasibility cut can be used to ensure the well-definedness of the algorithm, see [27] for details of the feasibility cut.

(ii) The properness of $Q_i(x_{a(i)})$ and $\hat{Q}_i(x_{a(i)})$ implies that the problem defined in (13) and (26) must have optimal solutions, otherwise $Q_i(x_{a(i)})$ and $\hat{Q}_i(x_{a(i)})$ may take the value of $-\infty$. Furthermore, the properness of $Q_j(x_i)$ and $\hat{Q}_j(x_i)$, $(j \in \mathcal{T}(i))$ also ensures the properness of $\mathcal{Q}_i(x_i)$ and $\hat{\mathcal{Q}}_i(x_i)$.

(iii) Assumption (iii) is called the inf-compactness condition, which is important in the stability analysis of an optimization problem. See e.g. Section 4.1 in [46] for references. □

The next result shows the relationship between $Q_i(x_{a(i)})$, $\mathcal{Q}_i(x_i)$ and their approximations. This also verifies that the cutting planes contained in $\tilde{\mathcal{Q}}_i^k(x_i)$ are also cutting planes of the real cost-to-go function $\mathcal{Q}_i(x_i)$.

Proposition 12 *For all $i \in \mathcal{N}$, and $k = 1, \cdots$, it holds that $\hat{Q}_i^k(x_{a(i)}) \leq Q_i(x_{a(i)})$, $\tilde{\mathcal{Q}}_i^k(x_i) \leq \hat{\mathcal{Q}}_i^k(x_i) \leq \mathcal{Q}_i(x_i)$.*

Proof The proof is by induction. If i is a leaf node, we have $\hat{Q}_i^k \equiv Q_i$, and $\tilde{\mathcal{Q}}_i \equiv \hat{\mathcal{Q}}_i \equiv \mathcal{Q}_i \equiv 0$, the statement trivially holds. Generally, for any node $i \in \mathcal{N}$, suppose that for all $j \in \mathcal{T}(i)$, the statement holds. Hence, $\hat{Q}_j^k(x_i) \leq Q_j(x_i)$, which implies that $\hat{\mathcal{Q}}_i^k(x_i) \leq \mathcal{Q}_i(x_i)$. Since $\tilde{\mathcal{Q}}_i^k(x_i)$ is constructed by a series of cutting planes of $\hat{\mathcal{Q}}_i^k(x_i)$, then, $\tilde{\mathcal{Q}}_i^k(x_i) \leq \hat{\mathcal{Q}}_i^k(x_i)$. Finally, by the definition of $\hat{Q}_i^k(\cdot)$, we have $\hat{Q}_i^k(x_{a(i)}) \leq Q_i(x_{a(i)})$. Consequently, the statement holds for all $i \in \mathcal{N}$. □

Together with Assumption 11 (iii), this proposition implies that, if the inf-compactness condition holds for $\hat{Q}_j^k$, then it also holds for $\hat{\mathcal{Q}}_i^k$ and $\mathcal{Q}_i$. Specifically, we have the next result.

Corollary 13 *Suppose Assumption 11 holds, and denote*

$$\begin{aligned}\mathrm{lev}_\alpha \hat{\mathcal{Q}}_i^k &:= \left\{ x_i \in C_i(\tilde{x}_{a(i)}) : \hat{\mathcal{Q}}_i^k(x_i) \le \alpha \right\} \text{ and} \\ \mathrm{lev}_\alpha \mathcal{Q}_i &:= \left\{ x_i \in C_i(\tilde{x}_{a(i)}) : \mathcal{Q}_i(x_i) \le \alpha \right\}.\end{aligned}$$

Then $\mathrm{lev}_\alpha \mathcal{Q}_i \subseteq \mathrm{lev}_\alpha \hat{\mathcal{Q}}_i^k \subseteq C$ *(*C *is defined in Assumption 11).*

Proof Firstly, by $\hat{\mathcal{Q}}_i^k(x_i) \le \mathcal{Q}_i(x_i)$ it readily holds that $\mathrm{lev}_\alpha \mathcal{Q}_i \subseteq \mathrm{lev}_\alpha \hat{\mathcal{Q}}_i^k$. Secondly, choose any $x_i \in \mathrm{lev}_\alpha \hat{\mathcal{Q}}_i^k$, then $\hat{\mathcal{Q}}_i^k(x_i) \le \alpha$. By (27), this means that there exists $j_0 \in \mathcal{T}(i)$ such that $\hat{Q}_{j_0}^k(x_i) \le \alpha$. Thus, by Assumption 11 (iii), we have $x_i \in C$. This implies $\mathrm{lev}_\alpha \hat{\mathcal{Q}}_i^k \subseteq C$. □

3 Convergence of the Decomposition Method

As is mentioned in Section 1, for multistage distributionally robust problem with moment constraints, the convergence analysis will be much more complicated than that in most current literature. Since the cost-to-go function $\mathcal{Q}_i(\cdot)$ will no longer be polyhedral, we can not take use of the finiteness of the cutting planes any more. In this section, we establish a new framework, by using the results in parametric optimization, to analyze the convergence properties of the presented decomposition algorithm. We first cite the next proposition, which is a general result on the stability of the optimal value function and optimal solutions of a general optimal problem.

Proposition 14 *([46] Proposition 4.4) Consider the parameterized optimization problems of the form*

$$\min_{x \in X} f(x,u) \quad s.t. \quad G(x,u) \in K. \tag{45}$$

Denote

$$\Phi(u) := \{x \in X : G(x,u) \in K\}$$

and the optimal solution set

$$\mathcal{S}(u) := \mathrm{argmin}_{x \in \Phi(u)} f(x,u).$$

Let u_0 *be a given point in the parameter space* U*. Suppose that* (i) *the function* $f(x,u)$ *is continuous on* $X \times U$*;* (ii) *the multifunction* $\Phi(\cdot)$ *is closed;* (iii) *there exists* $\alpha \in \mathbb{R}$ *and a compact set* $C \in X$ *such that for every* u *in a neighborhood of* u_0*, the level set*

$$\mathrm{lev}_\alpha f(\cdot,u) := \{x \in \Phi(u) : f(x,u) \le \alpha\}$$

is nonempty and contained in C; and (iv) *for any neighborhood $\mathcal{V}_X$ of the set $\mathcal{S}(u_0)$ there exists a neighborhood $\mathcal{V}_U$ of u_0 such that $\mathcal{V}_X \cap \Phi(u) \neq \emptyset$ for all $u \in \mathcal{V}_U$. Then* (a) *the optimal value function $\phi(u) := \inf_{x\in\Phi(u)} f(x,u)$ is continuous at $u = u_0$ and* (b) *the multifunction $u \mapsto \mathcal{S}(u)$ is upper semicontinuous at u_0.*

For optimization problem in finite-dimensional spaces, we can further obtain the next result.

Lemma 15 *Define the function $\phi : U \to [-\infty, +\infty]$ as*

$$\begin{aligned} \phi(u) := \min\ & f(x,u) \\ \text{s.t. } & x \in \Phi(u), \end{aligned} \tag{46}$$

where $f : \mathbb{R}^n \times U \to \mathbb{R}$, and the point-to-set mapping $\Phi : U \to \mathcal{P}(\mathbb{R}^n)$ is called the constraint mapping. Suppose that Φ is continuous at $\bar{u} \in U$ with $\Phi(\bar{u}) \neq \emptyset$ and f is continuous at $\Phi(\bar{u}) \times \{\bar{u}\}$ with $\phi(\bar{u}) > -\infty$. Furthermore, suppose condition (iii) *in Proposition 14 holds. Then* (a) *ϕ is continuous at $\bar{u}$; and* (b) *under the additional assumption of the optimal solution set $\mathcal{S}(\bar{u})$ being a singleton, the multifunction $u \mapsto \mathcal{S}(u)$ is continuous at $\bar{u}$.*

Proof It suffices to verify that the conditions in Proposition 14 are satisfied in a neighbour of $\Phi(\bar{u}) \times \{\bar{u}\}$. Notice that (i) is just the continuity of f with respect to jointly x and u. (iii) is guaranteed by the Lemma's assumptions. On the other hand, for given $\Phi(u)$, it is natural to define $G(x,u) := d(x, \Phi(u))$ and $K := \{0\}$ to reformulate (46) as the form of (45). Thus the continuity of the mapping Φ implies that $G(x,u)$ is continuous. Hence (ii) and (iv) hold (See e.g. Page 264 in [46] for details). Consequently, the statement (a) holds. Furthermore, the multifunction $u \mapsto \mathcal{S}(u)$ is upper semicontinuous at $\bar{u}$. Since $\mathcal{S}(\bar{u})$ is a singleton, one must have that $\mathcal{S}(u)$ converges to $\mathcal{S}(\bar{u})$ with $u \to \bar{u}$, so the statement (b) holds. □

Proposition 16 *Suppose that Assumption 11 holds. Then both $Q_i(x_{a(i)})$ and $\mathcal{Q}_i(x_i)$ are continuous when $x_{a(i)} \geq 0$ and $x_i \geq 0$.*

Proof For each node i, by Assumption 11, it is easy to verify that, at any $x_{a(i)} \geq 0$, the constraint mapping $C_i(x_{a(i)}) := \{x_i \geq 0 | B_i x_{a(i)} + A_i x_i = b_i\}$ is continuous and $C_i(x_{a(i)}) \neq \emptyset$.

The proof is by induction. Firstly, if i is a leaf node, then we have $\mathcal{Q}_i(x_i) \equiv 0$, and by Lemma 15, $Q_i(x_{a(i)})$ is continuous when $x_{a(i)} \geq 0$. In general, let i be a non-leaf node, and suppose that for all $j \in \mathcal{T}(i)$, $Q_j(x_i)$ is continuous at $x_i \geq 0$. Then, $f(p, x_i) := \sum_{j\in\mathcal{T}(i)} p_{ij} Q_j(x_i)$ is continuous at $(p_{ij})_{j\in\mathcal{T}(i)} \times x_i$. Hence, by Lemma 15, it holds that $\mathcal{Q}_i(x_i)$ is continuous at $x_i \geq 0$, which implies that $Q_i(x_{a(i)})$ is also continuous at $x_{a(i)}$. □

Following the same path of proof, we have the next result.

Proposition 17 *Suppose that Assumption 11 holds. Then for each node $i \in \mathcal{N}$, both $\hat{Q}_i(x_{a(i)})$ and $\hat{\mathcal{Q}}_i(x_i)$ are continuous when $x_{a(i)} \geq 0$ and $x_i \geq 0$.*

For simplicity, we denote $x^k = (x_i^k), i \in \mathcal{N}$ for the entire scenario tree. Recall that for sets $A, B \subseteq \mathbb{R}^n$, $\text{dist}\,(x, A) := \inf_{x' \in A} \|x - x'\|$ and $\mathbb{D}(A, B) := \sup_{x \in A} \text{dist}\,(x, B)$.

Proposition 18 *([47],Proposition 2.1.5(b)) Let X be a Banach space with X^* being the dual space of continuous linear functionals on X. Let f be Lipschitzian near a given point x and let x_i and ξ_i be repectively the sequences in X and X^* such that $\xi_i \in \partial f(x_i)$, where $\partial f(x)$ is the generalized subdifferential. Suppose that x_i converge to x, and that ξ is a cluster point of ξ_i in the weak* topology. Then one has $\xi \in \partial f(x)$.*

Notice that the subgradient for a proper convex function is a special case of the generalized gradient, we readily obtain the next result.

Lemma 19 *Let $f : \mathbb{R}^n \to (-\infty, +\infty]$ be a closed proper convex function, and $\{z^k\} \subseteq \mathbb{R}^n$ be a sequence converging to z^*. If $\zeta^k \in \partial f(z^k)$ and $\{\zeta^k\} \to \zeta^*$, then $\zeta^* \in \partial f(z^*)$.*

Assumption 20 *There is a common bound C such that for all nodes $i \in \mathcal{N}$ and any $x_i \in \mathcal{C}_i$, where $\mathcal{C}_i = \text{dom}\mathcal{Q}_i$, it holds that $\|g\| \leq C$ for all $g \in \partial \mathcal{Q}_i(x_i)$.*

Proposition 21 *Let $f : \mathbb{R}^n \to (-\infty, +\infty]$ be a closed proper convex function with its domain denoted by X, and suppose there exists a constant C such that $\|g\| \leq C$ for any $x \in X$ and all $g \in \partial f(x)$. Denote $\{x^k\} \subseteq X$ be a sequence converging to x^*, $V_k = \partial f(x^k)$ and $V_* = \partial f(x^*)$. Then $\lim_{k\to\infty} \mathbb{D}(V_k, V_*) = 0$.*

Proof The proof is by contradiction. Suppose that there exists $\epsilon > 0$ and a subsequence $\{x^k\}_K, K \subseteq \{1, 2, \cdots\}$ with some $g_k \in \partial f(x^k)$, such that $\text{dist}\,(g_k, V_*) \geq \epsilon$ for all $k \in K$. Since $\|g_k\| \leq C$, without loss of generality, we can assume that $\{g_k\}_K$ converges to some point g_*. It follows that $\text{dist}\,(g_*, V_*) \geq \epsilon$. However, by Lemma 19 we have $g_* \in V_*$, which leads to a contradiction. □

Lemma 22 *Suppose that there exists an iteration subsequence $\{x^k\}_K$ which converges to a limit point x^*. For any $i \in \mathcal{N}$ fixed, suppose that $\hat{\mathcal{Q}}_i^k(x_i^k)$ converges to $\mathcal{Q}_i(x_i^*)$ and there exists a constant C_i such that $\|g_i^k\| \leq C_i$ for any $g_i^k \in \partial \hat{\mathcal{Q}}_i^k(x_i^k)$ and $k \in K$ sufficiently large. Then $\lim_{k\to\infty, k\in K} \hat{Q}_i^k(x_{a(i)}^k) = Q_i(x_{a(i)}^*)$ and $\lim_{k\to\infty, k\in K} \tilde{\mathcal{Q}}_i^k(x_i^k) = \mathcal{Q}_i(x_i^*)$.*

Proof By Proposition 12, we have $\hat{Q}_i^k(x_{a(i)}^k) \leq Q_i(x_{a(i)}^k)$ and $\tilde{\mathcal{Q}}_i^k(x_i^k) \leq \mathcal{Q}_i(x_i^k)$. Since $Q_i(\cdot)$ and $\mathcal{Q}_i(\cdot)$ are continuous, it holds that

$$\limsup_{k\to\infty, k\in K} \hat{Q}_i^k(x_{a(i)}^k) \leq Q_i(x_{a(i)}^*) \tag{47}$$

and

$$\limsup_{k\to\infty, k\in K} \tilde{\mathcal{Q}}_i^k(x_i^k) \leq \mathcal{Q}_i(x_i^*). \tag{48}$$

On the other hand, it follows from the definition of $\hat{Q}_i^k(\cdot)$ that $\hat{Q}_i^k(x_{a(i)}^k) = c_i^\top x_i^k + \tilde{\mathcal{Q}}_i^k(x_i^k)$. Denote $K = \{k_1, k_2, \cdots, k_l, \cdots\}$, for any $k_l \in K$, we have that

$$\tilde{\mathcal{Q}}_i^{k_l}(x_i^{k_l}) \geq \hat{\mathcal{Q}}_i^{k_{l-1}}(x_i^{k_{l-1}}) + (g_i^{k_{l-1}})^\top (x_i^{k_l} - x_i^{k_{l-1}}), \tag{49}$$

where $g_i^{k_{l-1}} \in \partial \hat{\mathcal{Q}}_i^{k_{l-1}}(x_i^{k_{l-1}})$. Consequently,

$$\begin{aligned} \hat{Q}_i^{k_l}(x_{a(i)}^{k_l}) &= c_i^\top x_i^{k_l} + \tilde{\mathcal{Q}}_i^{k_l}(x_i^{k_l}) \\ &\geq c_i^\top x_i^{k_l} + \hat{\mathcal{Q}}_i^{k_{l-1}}(x_i^{k_{l-1}}) + (g_i^{k_{l-1}})^\top (x_i^{k_l} - x_i^{k_{l-1}}). \end{aligned} \tag{50}$$

Since $\{x^k\}_K \to x^*$, $\|g_i^k\| \leq C$, and $\hat{\mathcal{Q}}_i^{k_{l-1}}(x_i^{k_{l-1}})$ converges to $\mathcal{Q}_i(x_i^*)$, we have that

$$\liminf_{k\to\infty, k\in K} \hat{Q}_i^k(x_{a(i)}^k) \geq c_i^\top x_i^* + \mathcal{Q}_i(x_i^*) = Q_i(x_{a(i)}^*) \tag{51}$$

and

$$\liminf_{k\to\infty, k\in K} \tilde{\mathcal{Q}}_i^k(x_i^k) \geq \mathcal{Q}_i(x_i^*). \tag{52}$$

Together with (47) and (51), we get $\lim_{k\to\infty, k\in K} \hat{Q}_i^k(x_{a(i)}^k) = Q_i(x_{a(i)}^*)$. Similarly, from (48) and (52), we have $\lim_{k\to\infty, k\in K} \tilde{\mathcal{Q}}_i^k(x_i^k) = \mathcal{Q}_i(x_i^*)$. □

Corollary 23 *Suppose that there exists a subsequence $\{x^k\}_K$ that converges to a limit point x^*. For any fixed $i \in \mathcal{N}$, suppose that the assumptions of Lemma 22 hold. Then x_i^* is the optimal solution of the sub-problem defined by $Q_i(x_{a(i)}^*)$.*

Proof Recall that x_i^k $(k \in K)$ is the optimal solution of the next problem

$$\hat{Q}_i^k(x_{a(i)}^k) = \min_{x_i} \left\{ c_i^\top x_i + \tilde{\mathcal{Q}}_i^k(x_i) : B_i x_{a(i)}^k + A_i x_i = b_i, x_i \geq 0 \right\}. \tag{53}$$

Since $x_{a(i)}^k$ and x_i^k $(k \in K)$ converges to $x_{a(i)}^*$ and x_i^* respectively, obviously x_i^* satisfies the constraints $B_i x_{a(i)}^* + A_i x_i^* = b_i, x_i^* \geq 0$.

Moreover, by Lemma 22, we have

$$\begin{aligned} Q_i(x_{a(i)}^*) &= \lim_{k\to\infty, k\in K} \hat{Q}_i^k(x_{a(i)}^k) \\ &= \lim_{k\to\infty, k\in K} c_i^\top x_i^k + \tilde{\mathcal{Q}}_i^k(x_i^k) \\ &= c_i^\top x_i^* + \mathcal{Q}_i(x_i^*). \end{aligned} \tag{54}$$

The first equality follows from the first conclusion of Lemma 22, the second equality follows from the definition of $\hat{Q}_i^k(\cdot)$, while the last equality follows from the second conclusion of Lemma 22, which implies that the sub-problem in $Q_i(x_{a(i)}^*)$ attains its optimal value at x_i^*. □

Lemma 24 *Suppose that there is an iteration subsequence $\{x^k\}_K$ which converges to a limit point x^*. Let $i \in \mathcal{N}$ be any fixed node. Suppose that for all $j \in \mathcal{T}(i)$, $\lim_{k\to\infty, k\in K} \hat{Q}_j^k(x_i^k) = Q_j(x_i^*)$. Then $\lim_{k\to\infty, k\in K} \hat{\mathcal{Q}}_i^k(x_i^k) = \mathcal{Q}_i(x_i^*)$.*

Proof For any given $i \in \mathcal{N}$, denote $\phi(\cdot) : \mathbb{R}^{|\mathcal{T}(i)|} \to \mathbb{R}$

$$\phi(q) := \sup_{P_i \in \mathbb{P}_i} \sum_{j \in \mathcal{T}(i)} p_{ij} q_j. \tag{55}$$

Denote $q^k = (\hat{Q}_j^k(x_i^k))$, $q^* = (Q_j(x_i^*))$ $(j \in \mathcal{T}(i))$, then $\phi(q^k) = \hat{\mathcal{Q}}_i^k(x_i^k)$ and $\phi(q^*) = \mathcal{Q}_i(x_i^*)$. By the Lemma's assumption, we have $q^k \to q^* (k \in K)$. It follows immediately from Lemma 15 that $\phi(q^k)$ converges to $\phi(q^*)$. □

Lemma 25 *([40], Theorem 7.4) Let f_t: $\mathbb{R}^n \to \bar{\mathbb{R}}$, $t = 1, \cdots, m$ be proper convex functions, x_0 be a point such that $f_t(x_0)$ are all finite, denote $f(\cdot) = f_1(\cdot) + \cdots + f_m(\cdot)$. Then*

$$\partial f_1(x_0) + \cdots + \partial f_m(x_0) \subset \partial f(x_0). \tag{56}$$

Moreover,

$$\partial f_1(x_0) + \cdots + \partial f_m(x_0) = \partial f(x_0) \tag{57}$$

if any one of the following conditions holds: (i) the set $\cap_{i=1}^m \mathrm{ri}(\mathrm{dom} f_i)$ is nonempty; (ii) the functions $f_1, \cdots, f_k, k \leq m$, are polyhedral and the intersection of the sets $\cap_{i=k+1}^m \mathrm{ri}(\mathrm{dom} f_i)$ is nonempty.

Lemma 26 *Suppose Assumption 11 and 20 hold, and there is an iteration subsequence $\{x^k\}_K$ converging to its limit point x^*. For any fixed $i \in \mathcal{N}$, suppose that the assumptions of Lemma 22 holds. Further denote $V_i^k = \partial \hat{\mathcal{Q}}_i^k(x_i^k)$, $V_i^* = \partial \mathcal{Q}_i(x_i^*)$ and suppose $\lim_{k \to \infty, k \in K} \mathbb{D}(V_i^k, V_i^*) = 0$. Then it holds that*

$$\lim_{k \to \infty, k \in K} \mathbb{D}(\partial \hat{Q}_i^k(x_{a(i)}^k), \partial Q_i(x_{a(i)}^*)) = 0. \tag{58}$$

Proof For node $i \in \mathcal{N}$, denote $X_i := \{x_i | x_i \geq 0\}$, and N_{X_i} be its normal cone. Further denote δ_{X_i} be the indicator function of X_i, that is,

$$\delta_{X_i}(x) = 0 \text{ if } x \in X_i; \quad \delta_{X_i}(x) = +\infty \text{ if } x \notin X_i.$$

Moreover, we can define

$$\psi(x_{a(i)}, x_i) = \begin{cases} 0 & \text{if } B_i x_{a(i)} + A_i x_i = b_i, \\ +\infty & \text{if } B_i x_{a(i)} + A_i x_i \neq b_i. \end{cases} \tag{59}$$

Consequently, it can be verified that

$$\hat{Q}_i^k(x_{a(i)}^k) = \inf_{x_i} f_i^k \left(x_{a(i)}^k, x_i \right), \tag{60}$$

where $f_i^k(x_{a(i)}, x_i) = c_i^\top x_i + \tilde{\mathcal{Q}}_i^k(x_i) + \psi(x_{a(i)}, x_i) + \delta_{X_i}(x_i)$. Similarly, it holds that

$$Q_i^*(x_{a(i)}^*) = \inf_{x_i} f_i^* \left(x_{a(i)}^*, x_i \right), \tag{61}$$

where $f_i^*(x_{a(i)}, x_i) = c_i^\top x_i + \mathcal{Q}_i^*(x_i) + \psi(x_{a(i)}, x_i) + \delta_{X_i}(x_i)$.

Based on Theorem 10.13 in [48], it can be verified that

$$\partial \hat{Q}_i^k(x_{a(i)}^k) = \left\{ y | (0, y) \in \partial f_i^k \left(x_{a(i)}^k, x_i^k \right) \right\} \tag{62}$$

and

$$\partial Q_i^*(x_{a(i)}^*) = \left\{ y | (0, y) \in \partial f_i^* \left(x_{a(i)}^*, x_i^* \right) \right\}. \tag{63}$$

It follows from Assumption 11 that, the feasible set $C_i(x_{a(i)})$ of (13) has nonempty relative interior, hence the condition needed by (57) holds. Therefore, we have that

$$\partial f_i^k \left(x_{a(i)}^k, x_i^k \right) = (0, c_i^\top)^\top + \partial \tilde{\mathcal{Q}}_i^k(x_i^k) + \partial \psi \left(x_{a(i)}^k, x_i^k \right) + \partial \delta_{X_i}(x_i^k) \tag{64}$$

and

$$\partial f_i^* \left(x_{a(i)}^*, x_i^* \right) = (0, c_i^\top)^\top + \partial \mathcal{Q}_i^*(x_i^*) + \partial \psi \left(x_{a(i)}^*, x_i^* \right) + \partial \delta_{X_i}(x_i^*). \tag{65}$$

Since $x_{a(i)}^k$ and x_i^k $(k \in K)$ converges to $x_{a(i)}^*$ and x_i^* respectively, together with the assumption that $\lim_{k \to \infty, k \in K} \mathbb{D}(V_i^k, V_i^*) = 0$, it is easy to verify that (58) holds. □

Proposition 27 *([49], Proposition 4.2.5) Let $f : \mathbb{R} \to \mathbb{R}$ be a continuously differentiable and monotonically nondecreasing function and let $g : \mathbb{R}^n \to \mathbb{R}$ be a convex function. Suppose the composite function $F(\cdot) := f(g(\cdot))$ is convex. Then the subgradient of $F(\cdot)$ is given by*

$$\partial F(x) = \nabla f(g(x)) \partial g(x). \tag{66}$$

Corollary 28 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function with its gradient $\nabla f(\cdot) \geq 0$, and $g : \mathbb{R}^m \to \mathbb{R}^n$ be a mapping with each component $g_i(\cdot)$ $(i = 1 \cdots, n)$ be a convex function, if the composite function $F(\cdot) := f(g(\cdot))$ are convex, and the set $\cap_{i=1}^n$ ri(dom g_i) is nonempty, then the subgradient of $F(\cdot)$ is given by*

$$\partial F(x) = \sum_{i=1}^{n} f_i'(g(x)) \partial g_i(x), \tag{67}$$

where $f_i'(\cdot)$ is the ith component of $\nabla f(\cdot)$.

Proof Similar to the proof of Proposition 27, it is easy to verify that the directional derivative of F is

$$F'(x; y) = \sum_{i=1}^{n} f_i'(g(x)) g_i'(x; y), \quad x, y \in \mathbb{R}^n, \tag{68}$$

where $g_i'(x; y)$ is the directional derivative of $g_i(x)$ along direction y. Furthermore, it follows from the relationship between the subgradient and directional derivative that

$$d \in \partial F(x) \Leftrightarrow \forall y \in \mathbb{R}^n \quad y'd \leq F'(x; y). \tag{69}$$

Together with (68), we have that $d \in \partial F(x)$ is equivalent to

$$\forall y \in \mathbb{R}^n \quad y'd \leq \sum_{i=1}^{n} f_i'(g(x))g_i'(x;y). \tag{70}$$

Denote $f_i = f_i'(g(x))$, by the corollary's assumption, we have $f_i \geq 0$. Consequently, (70) is further equivalent with

$$d \in \partial \left[\sum_{i=1}^{n}(f_i \cdot g_i(x))\right] = \sum_{i=1}^{n} f_i'(g(x))\partial g_i(x). \tag{71}$$

Consequently, we have that $d \in \partial F(x)$ is equivalent with (71), which shows that (67) is correct. □

Lemma 29 *Denote by $\{x^k\}_K$ an iteration subsequence which converges to a limit point x^*. Let $i \in \mathcal{N}$ be any fixed node, suppose that for any $q^k = (\hat{Q}_j^k(x_i^k))$ and $q^* = (Q_j(x_i^*))$ $(j \in \mathcal{T}(i))$, problem (55) has a unique optimal solution, and the set $\cap_{j\in\mathcal{T}(i)}\mathrm{ri}(\mathrm{dom}\hat{Q}_j^k)$ and $\cap_{j\in\mathcal{T}(i)}\mathrm{ri}(\mathrm{dom}Q_j)$ are nonempty. Further suppose that $q^k \to q^*$. Denote $V_i^k = \partial\hat{\mathcal{Q}}_i^k(x_i^k)$ and suppose that for all $j \in \mathcal{T}(i)$, $\lim_{k\to\infty,k\in K}\mathbb{D}(\partial\hat{Q}_j^k(x_i^k), \partial Q_j(x_i^*)) = 0$. Then it holds that*

$$\lim_{k\to\infty,k\in K}\mathbb{D}(V_i^k, \partial\mathcal{Q}_i(x_i^*)) = 0. \tag{72}$$

Proof Denote the optimal solution set $\mathcal{S}(q^k) = \{p^k\}$ and $\mathcal{S}(q^*) = \{p^*\}$. It can be verified that condition (i)-(iv) in Proposition 4.12 in [46] holds for problem (55). Since (55) has a unique optimal solution, we have $\nabla\phi(q^k) = p^k$ and $\nabla\phi(q^*) = p^*$. It follows from the assumption $q^k \to q^*$ and Lemma 15 that $p^k \to p^*$.

Furthermore, by Corollary 28, it holds that

$$\partial\hat{\mathcal{Q}}_i^k(x_i^k) = \sum_{j\in\mathcal{T}(i)}\phi_i'(q^k)\partial\hat{Q}_j^k(x_i^k) = \sum_{j\in\mathcal{T}(i)}p_j^k\partial\hat{Q}_j^k(x_i^k). \tag{73}$$

Similarly, we have that

$$\partial\mathcal{Q}_i(x_i^*) \supseteq \sum_{j\in\mathcal{T}(i)}\phi_i'(q^*)\partial Q_j(x_i^*) = \sum_{j\in\mathcal{T}(i)}p_j^*\partial Q_j(x_i^*). \tag{74}$$

The lemma follows immediately. □

Lemmae 22-29 constitute a complete chain of induction. We can now establish the convergence of the proposed algorithm.

Theorem 30 *Suppose that Assumption 11 and 20 hold, and there exists an iteration subsequence $\{x^k\}_K$ which converges to a limit point x^*, for each node $i \in \mathcal{N}$. Further suppose that for all $q^k = (\hat{Q}_j^k(x_i^k))$ and $q^* = (Q_j(x_i^*))$ $(j \in \mathcal{T}(i))$, problem (27) has a unique optimal solution, and the set $\cap_{j\in\mathcal{T}(i)}\mathrm{ri}(\mathrm{dom}\hat{Q}_j^k)$ and $\cap_{j\in\mathcal{T}(i)}\mathrm{ri}(\mathrm{dom}Q_j)$ are nonempty. Then*

(i) $\lim_{k\to\infty,k\in K} \hat{Q}_i^k(x_{a(i)}^k) = Q_i(x_{a(i)}^*)$;
(ii) $\lim_{k\to\infty,k\in K} \hat{\mathcal{Q}}_i^k(x_i^k) = \mathcal{Q}_i(x_i^*)$;
(iii) $\lim_{k\to\infty,k\in K} \tilde{\mathcal{Q}}_i^k(x_i^k) = \mathcal{Q}_i(x_i^*)$; *and*
(iv) *for any* $i \in \mathcal{N}$, x_i^* *is the optimal solution of the right side problem of (13). That is,* $x^* = (x_i^*), i \in \mathcal{N}$ *is the optimal solution of the entire multistage problem.*

Proof Under the assumptions of the theorem, it follows from Proposition 16 and 17 that the functions $Q_i(x_{a(i)})$, $\mathcal{Q}_i(x_i)$, $\hat{Q}_i(x_{a(i)})$ and $\hat{\mathcal{Q}}_i(x_i)$ are all continuous in their feasible sets.

The proof of (i)-(iii) is by induction. If i is a leaf node, we have $\hat{Q}_i^k \equiv Q_i$, and $\tilde{\mathcal{Q}}_i^k \equiv \hat{\mathcal{Q}}_i^k \equiv \mathcal{Q}_i \equiv 0$, then (i)-(iii) holds for i immediately from the convergence of $\{x^k\}_K$ and the continuousness of Q_i at x^*. Furthermore, by Lemma 26, we have

$$\lim_{k\to\infty,k\in K} \mathbb{D}(\partial\hat{Q}_i^k(x_{a(i)}^k), \partial Q_i(x_{a(i)}^*)) = 0. \tag{75}$$

Now for any $i \in \mathcal{N}$ fixed, suppose that for any $j \in \mathcal{T}(i)$, (i)-(iii) and (75) hold. By Lemma 24, we have $\lim_{k\to\infty,k\in K} \hat{\mathcal{Q}}_i^k(x_i^k) = \mathcal{Q}_i(x_i^*)$, i.e., (ii) holds. On the other hand, by Lemma 29 and (75), we have

$$\lim_{k\to\infty,k\in K} \mathbb{D}(\partial\hat{\mathcal{Q}}_i^k(x_i^k), \partial\mathcal{Q}_i(x_i^*)) = 0. \tag{76}$$

Consequently, the conditions of Lemma 22 hold, which implies (i) and (iii) holds for i. Furthermore, by Lemma 26 and (76), we obtain that (75) hold for i. Hence, the induction holds, that is, (i)–(iii) and (75) hold for all $i \in \mathcal{N}$.

Finally, since the conditions of Lemma 22 hold for all $i \in \mathcal{N}$, (iv) holds by Corollary 23. □

4 Numerical Test on a Inventory Control Problem

In this section, we test our algorithm on a classical multistage inventory control problem. The original form of this problem is described in the AIMMS optimization modeling book ([50], Chapter 17). Here we test a more complicated version that introduces the distributional robustness.

The multi-period inventory control model aims to maximize the total expected profit by deciding the production, inventory and external supply volume of two products at each stage (or each node of the scenario trees). All the parameters and variables are listed in Table 1. In particular, at each node i, the random variables are the demand vector d_i of the two products. Among the deterministic parameters, the selling price ps, the production cost cp, inventory ci, and external supply ce are all vectors in $\mathbb{R}^2$. As to the decision variables, denote $x_i = (x_i^1, x_i^2)$, of which the elements x_i^1 and x_i^2 are the volume of the two types of production respectively. The meanings of the elements of $y_i = (y_i^1, y_i^2)$ and $z_i = (z_i^1, z_i^2)$ are likewise.

Table 1 Symbols and Notations

Indices	
i	node
Parameters	
ps	selling price vector
cp	production cost vector
ci	inventory cost vector
ce	external supply cost vector
c	overall capacity
$\bar{y}$	maximum inventory
Random factors	
d_i	demand vector of node i
Decision Variables	
x_i	production volume of node i
y_i	inventory volume of node i
z_i	external supply of node i
v_i	profit of node i

Consider the following dynamic programming form of this model.

$$
\begin{aligned}
Q_i(y_{a(i)}) := \min_{v_i, x_i, y_i, z_i} \quad & -v_i + \mathcal{Q}_i(y_i) && \text{(77a)}\\
\text{s.t.} \quad & x_i^1 + x_i^2 \le c, && \text{(77b)}\\
& x_i + y_{a(i)} + z_i = d_i + y_i, && \text{(77c)}\\
& y_i^1 + y_i^2 \le \bar{y}, && \text{(77d)}\\
& y_{a(i)} + z_i \ge d_i, && \text{(77e)}\\
& v_i = ps^\top d_i - (cp^\top x_i + ci^\top y_i + ce^\top z_i), && \text{(77f)}\\
& x_i, y_i, z_i \in \mathbb{R}_+^2, v_i \in \mathbb{R}, && \text{(77g)}
\end{aligned}
$$

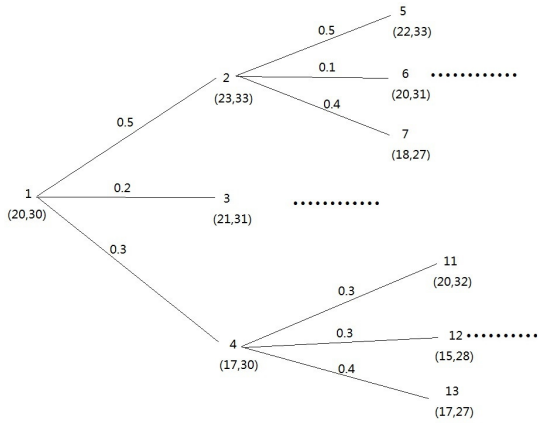
where $\mathcal{Q}_i(\cdot)$ is defined as follows, in which the ambiguity set $\mathbb{P}_i$ is defined by (7),

$$
\mathcal{Q}_i(y_i) = \sup_{P_i \in \mathbb{P}_i} \sum_{j \in \mathcal{T}(i)} p_{ij} Q_j(y_i). \tag{78}
$$

If i is a leaf node, $\mathcal{Q}_i(y_i) = 0$. The symbols used in model (77) are listed as follows.

- Constraint (77b) states that the total production of the two products is restricted by the overall capacity c.
- (77c) is the inventory decision constraint, which means that at each node, the summation of the production, the external supply of the current node and the inventory from the predecessor node equals to the summation of demand and inventory at the current node.
- Constraint (77d) ensures that the total inventory volume is bounded by the maximum inventory capacity.
- (77e) shows that the stochastic demand should be met by the inventory of the predecessor node and the external supply at current node. This is due to the fact that the production of the current node is prepared for the later stage.

Fig. 3 Part of the Scenario Tree



– Finally, constraint (77f) is the expression of the profit at each node.

Moreover, $-\mathcal{Q}_i(\cdot)$ is the worst case expected cumulative profit of the subtree rooted at node i. If i is the root node, $y_{a(i)}$ is actually the initial inventory, and the objective function $Q_i(y_{a(i)})$ is the total expected profit under the worst case.

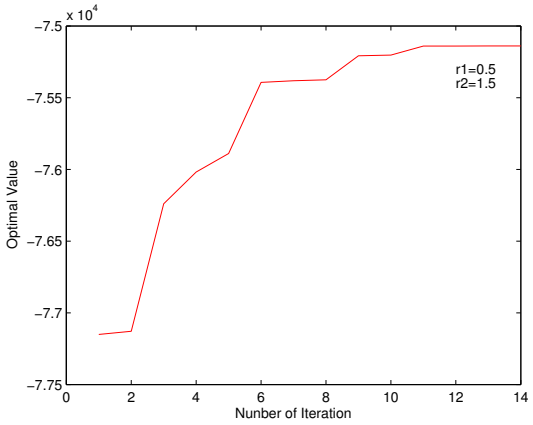
The parameters are set as follows: $ps = (300, 400)^\top$, $cp = (12, 10)^\top$, $ci = (5, 5)^\top$, $ce = (195, 200)^\top$, the overall production capacity c is 46, the maximum inventory $\bar{y}$ is 52, and the initial inventory level is $(17, 35)^\top$.

We now describe the scenario tree of the tested example. In the example, each node, except the leaf nodes, has three children nodes. The test problem has 5 stages, including 81 scenarios and 121 nodes ($121 = 1 + 3^1 + \cdots + 3^4$). Suppose that, for each node, we have obtained its demand and the empirical transition probability to each of its children nodes. Thus we obtain an estimation of the expectation and the covariance matrix of its predecessor. Part of this tree is shown in Figure 3.

Since the empirical distribution may not be an accurate estimation of the real one (this is very likely to occur especially for the leaf nodes), we use the empirical probability to estimate the moments of the demand and then solve the distributionally robust problem. For instance, at the root node 1, the demand is $d_1 = (20, 30)^\top$, the transition probability is $p_{12} = 0.5, p_{13} = 0.2, p_{14} = 0.3$, hence the moment statistics is calculated as follows:

$$\begin{aligned}\hat{\mu}_1 &= \sum_{j=2}^{4} p_{1j} d_j = (20.8, 31.7)^\top, \\ \hat{\Sigma}_1 &= \sum_{j=2}^{4} p_{1j} (d_j - \hat{\mu}_1)(d_j - \hat{\mu}_1)^\top = \begin{pmatrix} 6.76 & 3.34 \\ 3.34 & 1.81 \end{pmatrix}.\end{aligned}$$

Fig. 4 Numerical Result for $\gamma_1 = 0.5$ and $\gamma_2 = 1.5$



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
γ_1	0	0.25	0.5	0.75	1	1.25	1.5	1.6	1.75	1.85	2	2.25	2.5	4	6	8
γ_2	1	1.25	1.5	1.75	2	2.25	2.5	2.6	2.75	2.85	3	3.25	3.5	5	7	9

Table 2 Tested Parameters

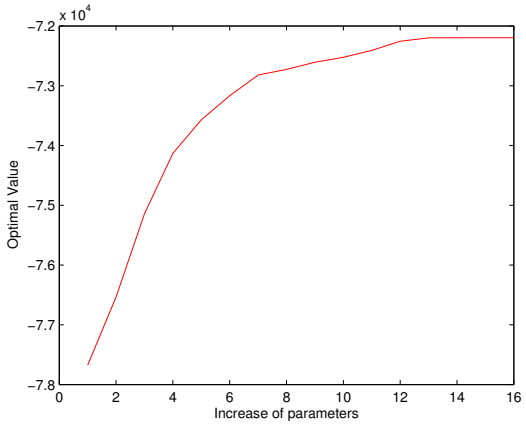
All numerical tests are coded in MATLAB 7.6 and run on a PC with an Intel Quad Core 3.4 GHz CPU and 8 GB of RAM under the Windows 7 operating system. We use the Matlab built-in solver *linprog* in the forward steps and use CVX with SeDuMi to solve the semidefinite programs in the backward steps.

Figure 4 illustrates the numerical result when $\gamma_1 = 0.5$ and $\gamma_2 = 1.5$. The algorithm converges after 12 iterations (each iteration consists of all the forward step and backward step for every node). The total runtime is 62.4 seconds. This means it spends about 5 seconds on average for each circulation (i.e., traverse all the nodes and compute the forward and backward steps.) The worst case expected profit $-\mathcal{Q}_{i_0}$ should be 71539.64. Also note that the optimal value is increasing during the additions of the cutting planes. Figure 5 shows the trend of the optimal value as the parameters γ_1 and γ_2 increase. The curve starts from the optimal value of the multistage stochastic programming with $\gamma_1 = 0$ and $\gamma_2 = 1$, and progressively converges to an upper bound. This coincides with the analysis of Proposition 1 and (15), which predicts that, when these parameters are large enough, the DRSTO problem will finally become the worst case single path problem of the scenario tree.

5 Conclusion

We proposed a time-consistent Benders decomposition method for solving multistage distributionally robust stochastic linear programs with a scenario tree structure. The distributional robustness is incorporated into each node of the

Fig. 5 Trend of the Optimal Value



scenario tree, therefore reflects the time consistency and facilitates the decomposition of the original problem into small ones with respect to each node. A new and complicated framework of convergence analysis is developed to establish the global convergence of the method, which does not depend on the assumption of polyhedral structure of the original problem. Numerical results of a practical inventory model are reported to demonstrate the effectiveness of the proposed method.

There have been abundant studies on various ambiguity sets for DRSTO problems. It is our belief that the proposed method, together with its convergence framework, is general enough to cope with most of them, for example, the ambiguity sets considered in Ling et al. [13] and some major ambiguity sets in Wiesemann et al. [51], etc.

For simplicity, in the proposed algorithm we unify the value of γ_1 and γ_2 for all nodes. In practice, however, it is better to set different values for different nodes. Generally speaking, we may have sufficient information on the root node, and have less information on its successors. This means, we should set small γ_1 and γ_2 for the first stage, while setting larger values for the later stage in order to avoid over-conservativeness. This might be another advantage of the proposed decomposition method, which allows dynamic adjustment of the parameters in the ambiguity sets at different nodes.

In the numerical experiments, we carried out the proposed algorithm on a small scale test problem. In practice, to deal with larger scale practical problems, the Benders decomposition method should be further incorporated into a duality dynamic programming framework and some acceleration techniques should be introduced as well. To construct a SDDP-type algorithm, the main obstacle is to design an upper bound for distributionally robust problems. As is mentioned in Section 2, one possible approach is to use a deterministic upper bound. As to the acceleration techniques for larger scale problems, there exist several alternative approaches to reducing the amount of the computation

related with the cutting planes. If the random process has the Markovain structure, we can use the cutting sharing techniques. Another type of technique is to delete redundant cutting planes, which means the scale of the constraints will be reduced. Other possible schemes include introduction of a quadratic regularized term to speed up the computation, a recent work of this type is [52]. These are possible topics in future study.

References

1. Pflug, G,C., Pichler, A.: Multistage Stochastic Optimization. Springer International Publishing, Switzerland. (2016)
2. Casey, M.S., Sen, S.: The Scenario Generation Algorithm for Multistage Stochastic Linear Programming. Math. Oper. Res. 30(3), 615-631 (2005)
3. Hyland, K., Kaut, M., Wallace, S.W.: A Heuristic for Moment-Matching Scenario Generation. Comput. Optim. Appl. 24(2-3), 169-185 (2003)
4. Kaut, M., Wallace, S.W.: Evaluation of scenario-generation methods for stochastic programming. Pac. J. Optim. 3(2), 257-271 (2003)
5. Rockafellar, R.T., Sun, J.: Solving monotone stochastic variational inequalities and complementarity problems by progressive hedging. Math. Program. 174(1), 453-471 (2019)
6. Delage, E., Ye, Y.: Distributionally robust optimization under moment uncertainty with application to data-driven problems. Oper. Res. 58(3), 595-612 (2010)
7. Wiesemann, W., Kuhn, D., Sim, M.: Distributionally robust convex optimization. Oper. Res. 62(6), 1358-1376 (2014)
8. Xu, H., Liu, Y., Sun, H.: Distributionally robust optimization with matrix moment constraints: Lagrange duality and cutting plane methods. Math. Program. 169(2), 1-41 (2017)
9. Shang, C., You, F.: Distributionally robust optimization for planning and scheduling under uncertainty. Comput. Chem. Eng. 110, 53-68 (2018)
10. Gao, S.Y., Kong, L., Sun, J.: Robust two-stage stochastic linear programs with moment constraints. Optimization. 63(6), 829-837 (2014)
11. Ang, J., Meng, F., Sun, J.: Two-stage stochastic linear programs with incomplete information on uncertainty. Eur. J. Oper. Res. 233, 16-22 (2014)
12. Ang, M., Sun, J., Yao, Q.: On the dual representation of coherent risk measures. Ann. Oper. Res. 262, 29-46 (2018)
13. Ling, A., Sun, J., Yang, X.: Robust tracking error portfolio selection with worst-case downside risk measures. J. Econ. Dyn. Control. 39, 178-207 (2014)
14. Ling, A., Sun, J., Xiu, N., Yang, X.G.: Robust Two-stage Stochastic Linear Optimization with Risk Aversion. Eur. J. Oper. Res. 256(1), 215-229 (2016)
15. Sun, J., Liao, L.Z., Rodrigues, B.: Quadratic two-stage stochastic optimization with coherent measures of risk. Math. Program. 168, 599-613 (2018)
16. Jiang, R., Guan, Y.: Risk-averse two-stage stochastic program with distributional ambiguity. Oper. Res. 66(5), 1390-1405 (2018)
17. Hanasusanto, G.A., Kuhn, D.: Conic programming reformulations of two-stage distributionally robust linear programs over wasserstein balls. Oper. Res. 66(3), 849-869 (2018)
18. Bansal, M., Huang, K.L., Mehrotra, S.: Decomposition algorithms for two-stage distributionally robust mixed binary programs. SIAM J. Optim. 28(3), 2360-2383 (2018)
19. Bansal, M., Mehrotra, S.: On solving two-stage distributionally robust disjunctive programs with a general ambiguity set. Eur. J. Oper. Res. 279(2), 296-307 (2019)
20. Analui, B., Pflug, G.C.: On distributionally robust multiperiod stochastic optimization. Comput. Manag. Sci. 11(3), 197-220 (2014)
21. Pflug, G.C., Pichler, A.: A distance for multistage stochastic optimization models. SIAM J. Optim. 22(1), 1-23 (2012)

22. Artzner, P., Delbaen, F., Eber, J.M., Heath, D., Ku, H.: Coherent multiperiod risk adjusted values and Bellman's principle. Ann. Oper. Res. 152, 5-22 (2007)
23. Ruszczyński, A.: Risk-averse dynamic programming for Markov decision processes. Math. Program. 125(2), 235-261 (2010)
24. Bielecki, T.R., Cialenco, I., Pitera, M.: A survey of time consistency of dynamic risk measures and dynamic performance measures in discrete time: LM-measure perspective. Probab. Uncertain. Quant. Risk. 2(1), 3-54 (2017)
25. Shapiro, A.: On a time consistency concept in risk averse multistage stochastic programming. Oper. Res. Lett. 37(3), 143-147 (2009)
26. Homem-de-Mello, T., Pagnoncelli, B. K.: Risk aversion in multi -stage stochastic programming: A modeling and algorithmic perspective. Eur. J. Oper. Res. 249(1), 188-199 (2016)
27. Ruszczyński, A.: Decomposition methods. in: Shapiro, A., Ruszczyński, A.: Stochastic Programming, vol. 10 of Handbooks in operations research and management science. pp.141-211. Elsevier (2003)
28. Rahmaniani, R., Crainic, T.G., Gendreau, M., Rei, W.: The Benders decomposition algorithm: A literature review. Eur. J. Oper. Res. 259(3), 801-817 (2017)
29. Wolf, C.: Advanced acceleration techniques for nested Benders decomposition in stochastic programming. Doctoral dissertation, University of Paderborn. (2014)
30. Pereira, M.V., Pinto, L.M.: Multi-stage stochastic optimization applied to energy planning. Math. Program. 52, 359-375 (1991)
31. Guigues, V.: Dual dynamic programing with cut selection: Convergence proof and numerical experiments. Eur. J. Oper. Res. 258(1), 47-57 (2017)
32. Guigues, V.: Inexact cuts in Stochastic Dual Dynamic Programming. SIAM J. Optim. 30, 407-438 (2020)
33. Rebennack, S.: Combining sampling-based and scenario-based nested Benders decomposition methods: application to stochastic dual dynamic programming. Math. Program. 156(1-2), 343-389 (2016)
34. Girardeau, P., Leclere, V., Philpott, A. B.: On the convergence of decomposition methods for multistage stochastic convex programs. Math. Oper. Res. 40(1), 130-145 (2015)
35. Baucke, R.: An algorithm for solving infinite horizon Markov dynamic programs. Optimization online, 2018. http://www.optimization-online.org/DB_HTML/2018/04/6565.html
36. Baucke, R., Downward, A., Zakeri, G.: A deterministic algorithm for solving multistage stochastic programming problems. Optimization Online, 2017. http://www.optimization-online.org/DB_FILE/2017/07/6138.html
37. Georghiou, A., Tsoukalas, A., Wiesemann, W.: Robust dual dynamic programming. Oper. Res. 67(3), 813-830 (2019)
38. Shapiro, A.: Distributionally robust stochastic programming. SIAM J. Optim. 27(4), 2258-2275 (2017)
39. Rockafellar, R.T.: Conjugate duality and optimization. SIAM, Philadelphia. (1974)
40. Shapiro, A., Dentcheva,D., Ruszczyński, A.: Lectures on Stochastic Programming: Modeling and Theory. SIAM, Philadelphia. (2009)
41. Shapiro, A.: On duality theory of conic linear problems. In Semi-infinite programming, pp. 135-165. Springer, Boston, MA. (2001)
42. Shapiro, A.: Topics in stochastic programming. CORE Lecture Series, Universite Catholique de Louvain. (2011)
43. Philpott, A., de Matos, V., Finardi, E.: On solving multistage stochastic programs with coherent risk measures. Oper. Res. 61(4), 957-970 (2013)
44. de Matos, V., Philpott. A.B., Finardi, E.C.: Improving the performance of stochastic dual dynamic programming. J. Comput. Appl. Math. 290(25),196-208 (2015)
45. Shapiro, A., Tekaya, W., Soares, M.P., da Costa, J.P.: Worst-case-expectation approach to optimization under uncertainty. Oper. Res. 61(6), 1435-1449 (2013)
46. Bonnans, J.F., Shapiro, A.: Perturbation analysis of optimization problems. Springer Science and Business Media. (2013)
47. Clarke, F.H.: Optimization and nonsmooth analysis. SIAM, Philadelphia. (1990)

48. Rockafellar, R.T., Wets, R.J.B.: Variational analysis. Springer Science and Business Media. (2009)
49. Bertsekas, D.P., Nedi, A., Ozdaglar, A.: Convex analysis and optimization. Athena Scientific. (2003)
50. Bisschop, J.: AIMMS-Optimization Modeling. AIMMS B.V. (2020)
51. Wiesemann, W., Kuhn, D., Sim, M.: Distributionally robust convex optimization. Oper. Res. 62, 1358-1376 (2014)
52. Asamov, T., Powell, W.B.: Regularized decomposition of high-dimensional multistage stochastic programs with markov uncertainty. SIAM J. Optim. 28(1), 575-595 (2018)