Computers & Operations Research 153 (2023) 106149

Contents lists available at ScienceDirect

Computers and Operations Research

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

# Scenario-dominance to multi-stage stochastic lot-sizing and knapsack problems

İ. Esra Büyüktahtakın

*Grado Department of Industrial and Systems Engineering, Virginia Tech, United States of America*

ARTICLE INFO

*Keywords:*
Scenario dominance
Partial ordering of scenarios
Multi-stage stochastic mixed-integer programs
Scenario tree
Cutting planes
Bounds
Stochastic
Dynamic knapsack problem
Lot-sizing
Inventory-production management
MIP test problem library

ABSTRACT

This paper presents strong *scenario dominance* cuts for effectively solving the multi-stage stochastic mixed-integer programs (M-SMIPs), specifically focusing on the two most well-known M-SMIPs: stochastic capacitated multi-item lot-sizing (S-MCLSP) and the stochastic dynamic multi-dimensional knapsack (S-MKP) problems. Scenario dominance is characterized by a partial ordering of scenarios based on the pairwise comparisons of random variable realizations in a scenario tree of a stochastic program. In this paper, we study the implications of scenario-dominance relations and inferences obtained by solving scenario sub-problems to drive new strong cutting planes to solve S-MCLSP and S-MKP instances faster. Computational experiments demonstrate that our strong scenario dominance cuts can significantly reduce the solution time for such M-SMIP problems with an average of 0.06% deviation from the optimal solution. The results with up to 81 random variables for S-MKP show that strong dominance cuts improve the state-of-the-art solver solution of two hours by 0.13% in five minutes. The proposed framework can also be applied to other scenario-based optimization problems.

## 1. Introduction

Practical decision problems involve a sequence of decisions over multiple time periods, high levels of uncertainty in the data, and discrete decision variables. Multi-stage stochastic mixed-integer programs (M-SMIPs) are widely used as a framework for formulating such sequential decision-making problems under uncertainty, including applications in finance (Mulvey and Vladimirou, 1992; Dantzig and Infanger, 1993), production and capacity acquisition (Ahmed and Garcia, 2003; Lulli and Sen, 2004), energy (Cerisola et al., 2009; Bruno et al., 2016; Cobuloglu and Büyüktahtakın, 2017), health-care (Yin and Büyüktahtakın, 2021, 2022; Yin et al., 2023b; Kıbış and Büyüktahtakın, 2019), and environment (Alonso-Ayuso et al., 2018; Kıbış et al., 2021; Bushaj et al., 2021), among others (Birge and Louveaux, 2011).

One particular challenge regarding the multi-stage stochastic mixed-integer programs is the non-convexity and discontinuity of the expected recourse function. M-SMIPs can be cast into an extensive mixed-integer programming (MIP) form where realizations of uncertain parameters are represented by a scenario tree; however, the size of the MIP grows exponentially in the number of decision periods and uncertainty outcomes in each period. As a result, M-SMIPs are notoriously difficult, requiring large-scale operations to solve practical size problems with billions of variables and constraints.

Recently, Büyüktahtakın (2022) has introduced the concept of scenario dominance to derive cuts based on the solution of a new scenario sub-problem for speeding up the solution of general risk-averse M-SMIPs. Scenario dominance is characterized by a partial ordering of scenarios based on the pairwise comparisons of random variable realizations. Using this partial ordering and the solution of a new scenario sub-problem, scenario dominance cuts are presented to effectively solve the risk-averse M-SMIPs by a cut-and-branch algorithm. The concept of scenario dominance, as defined in Büyüktahtakın (2022), is different than stochastic dominance as considered in the literature (Müller and Stoyan, 2002). In stochastic dominance, two random variables are compared based on their probability distribution function, while in scenario dominance, two scenarios are compared based on the discrete realization of random variables at each time stage with respect to the possible outcomes of one, two, or more variables. Later, Bushaj et al. (2022a) have extended the definition of scenario dominance presented by Büyüktahtakın (2022) under the case of endogenous uncertainty. The authors have incorporated the surveillance pattern in defining the scenario dominance and the associated cutting planes to solve the surveillance and control of a non-native forest insect, the emerald ash borer (EAB), while maximizing social benefits from healthy ash trees. Yin et al. (2023b) have modified the scenario-based sub-problems defined in Büyüktahtakın (2022) to create region-based sub-problems for deriving lower and upper bounds to reduce the optimality gap of a risk-averse M-SMIP. The authors particularly have focused on solving a multi-stage stochastic epidemics-ventilator-logistics formulation to address the resource allocation challenges of mitigating COVID-19.

*E-mail address:* esratoy@vt.edu.

https://doi.org/10.1016/j.cor.2023.106149
Received 14 April 2022; Received in revised form 5 January 2023; Accepted 9 January 2023
Available online 16 January 2023

In this paper, we present scenario dominance cutting planes and bounds for effectively solving the risk-neutral multi-stage stochastic mixed-integer programs (M-SMIPs). We first strengthen the scenario dominance cuts driven by Büyüktahtakın (2022). Different than the study of Büyüktahtakın (2022), we have driven and proved new and specified strong scenario dominance cuts for both the stochastic capacitated multi-item lot-sizing problems (S-MCLSP) and for the risk-neutral stochastic dynamic multi-dimensional knapsack (S-MKP). We introduce strong scenario dominance cuts for both the S-MCLSP and S-MKP and demonstrate the feasibility of the scenario dominance approach for solving those M-SMIPs efficiently. Our computational results demonstrate that the strong scenario dominance cuts are quite effective in reducing the solution time of those M-SMIP problems. Yin et al. (2023b) solve resource allocation problems in controlling epidemic diseases, and Bushaj et al. (2022a) tackle a forest insect surveillance and control planning issue utilizing region-based bounds and dominance cuts, respectively. Different than the works of Yin et al. (2023b) and Bushaj et al. (2022a), in this paper, we focus on solving more general multi-stage stochastic MIPs, such as the knapsack and lot-sizing problems. In Yin et al. (2023b), the authors derive the lower and upper bounds based on regional decomposition, while we use a scenario decomposition to generate those bounds. Also distinct from Bushaj et al. (2022a) and Büyüktahtakın (2022), we have generated more aggressive forms of scenario dominance cuts specified for S-MCLSP and S-MKP in this study.

### 1.1. Literature review and key contributions

The majority of recent research focuses on developing tailored algorithms for solving two-stage stochastic MIPs or multi-stage stochastic programs without discrete variables (for reviews, see Birge and Louveaux, 2011, Römisch and Schultz, 2001, Sen, 2005). The progress on general multi-stage stochastic mixed-integer programs is relatively limited. Due to the large-scale nature of M-SMIPs, one commonly used approach is to relax the coupling constraints (e.g., non-anticipativity) in order to decompose the M-SMIP problem into scenario-based sub-problems. Most solution algorithms for stochastic mixed-integer programs include some form of decomposition, including Lagrangian relaxation and scenario decomposition (CarøE and Schultz, 1999; Chen et al., 2002; Escudero et al., 2016), column generation (Lulli and Sen, 2004; Alonso-Ayuso et al., 2003), and decomposition-based heuristics, such as progressive hedging (Watson and Woodruff, 2011; Gade et al., 2016).

Other approaches, such as nested decomposition and Stochastic Dual Dynamic Programming (SDDP), focus on the convex relaxation of expected recourse function (Löhndorf et al., 2013; Heitsch and Römisch, 2003; Shapiro et al., 2013). Locally valid cuts are introduced and integrated into an SDDP framework (Abgottspon et al., 2014). McCormick cuts are studied to approximate the expected recourse function (Cerisola et al., 2012). This approach is later improved by Thome et al. (2013) by optimizing the Lagrangian multipliers, resulting in tighter cuts. Zou et al. (2019) propose Stochastic Dual Dynamic integer Programming (SDDiP) and Lagrangian cuts for solving multi-stage stochastic programs with binary state variables. Guan et al. (2009) present cutting planes for M-SMIPs based on combining inequalities that are valid for the individual scenarios and apply those cuts to solving stochastic multi-stage knapsack problems and stochastic dynamic lot-sizing problems. Hartman et al. (2010) present inequalities based on the iterative solutions of a dynamic program (Büyüktahtakın, 2011) for solving the capacitated lot-sizing (CLSP) problem. Büyüktahtakın et al. (2018b) then derive valid bounds on the partial objective function of the MCLSP formulation using dynamic programming and integer programming techniques. Ruszczyński (2002) has considered the partial ordering of the scenarios to study the two-stage stochastic programming problems with probabilistic constraints and reformulate them as a large-scale knapsack problem. The author has defined induced covers (Park and Park, 1997; Boyd, 1993) for the precedence-constrained knapsack polyhedra based on the partial ordering of scenarios. Other studies focus on developing problem-driven scenario reduction methods based on scenario similarities, effectiveness, or distances (Arpón et al., 2018; Rahimian et al., 2019; Bertsimas and Mundru, 2022). Despite recent progress, M-SMIPs remain challenging and require the development of advanced cutting-plane and scenario-based algorithms that take advantage of decomposition.

In this paper, we address the computational difficulty of the extensive formulation of M-SMIPs based on the scenario dominance concept that has been recently introduced in Büyüktahtakın (2022). Different from than former work, we derive new strong scenario dominance cuts to solve the stochastic capacitated multi-item lot-sizing problems (S-MCLSP) and stochastic dynamic multi-dimensional knapsack (S-MKP) and prove that those aggressive cuts satisfy at least one feasible solution. Our computational experiments show that the strong scenario dominance cuts can reduce the solution time for M-SMIPs, such as stochastic dynamic knapsack and lot-sizing problems, by one to two orders of magnitude with an average of 0.06% deviation from the optimal solution. We implement heuristics similar to a separation algorithm to improve the quality of the scenario dominance cuts. Our heuristic is different than the classical cutting plane separation by searching the aggressive cuts that will potentially preserve the optimal solution. The strong dominance cuts are also shown to improve the state-of-the-art solver solution quality by 0.13% in five minutes for the S-MKP instances with 81 random variables that cannot be solved in two hours.

The remainder of this paper is organized as follows. In Section 2, we review the multi-stage mixed-integer program and its extensive formulation and present the scenario sub-problem, bounds, and scenario dominance cuts for the risk-neutral M-SMIPs. Sections 3 and 4 present bounds and scenario dominance and the strong scenario dominance cuts specifically driven for the multi-stage stochastic lot-sizing and the multi-stage stochastic knapsack problems, respectively. Section 5 presents computational results, and Section 6 gives conclusions and directions for future research.

## 2. Scenario dominance to M-SMIPs

This section gives the general and extensive formulations of M-SMIP. We then present the scenario sub-problem and the associated bounds, the scenario dominance concept, and the scenario dominance cutting plane algorithm for the risk-neutral M-SMIPs.

### 2.1. Multi-stage stochastic mixed-integer programs

Here, we present the M-SMIP formulation, which represents multi-period discrete optimization models with dynamic stochastic data over time (Birge and Louveaux, 2011). We consider a finite multi-stage stochastic sequential decision-making process. Each decision is made at a discrete time period or stage $t \in \mathcal{T} = \{1, \dots, T\}$ based on the available information up to that stage. Let $\tilde{\xi}_1$ be known, or deterministic, and $\tilde{\xi}_2, \dots, \tilde{\xi}_T$ denote a sequence of random vectors that are observed as $\hat{\xi}_2, \dots, \hat{\xi}_T$, where $\xi_{[t]} := \{\hat{\xi}_2, \dots, \hat{\xi}_t\}$ is the information observed by stage $t$. Define also $\tilde{\xi}_{[t]} := (\tilde{\xi}_2, \dots, \tilde{\xi}_t)$ for $t = 2, \dots, T$. We assume that $\tilde{\xi}_{[2]}, \dots, \tilde{\xi}_{[T]}$ is a discrete-time stochastic process with a finite probability space $(\Xi_t, F_t, P_t)$, where $\Xi_t$ represents a finite sample space for $t = 2, \dots, T$, and $\Xi = \Xi_T$.

Let $n_t$, $q_t$, and $m_t$ represent the number of decision variables, the number of integer variables, and the number of constraints, respectively, at time $t$. Let $\mathbb{R}$ denote the set of real numbers, and $\mathbb{R}_+$ and $\mathbb{Z}_+$ denote the set of positive real numbers and positive integers, respectively. The general multi-stage stochastic mixed-integer program with recourse (M-SMIP) can then be expressed as follows:

$$\min\ f_1(x_1) + \mathbb{E}_{\tilde{\xi}_2}\left[\min_{x_2} f_2(x_2,\tilde{\xi}_2) + \mathbb{E}_{\tilde{\xi}_3|\xi_{[2]}}\left[\cdots + \mathbb{E}_{\tilde{\xi}_T|\xi_{[T-1]}}\left[\min_{x_T} f_T(x_T,\tilde{\xi}_T)\right]\right]\right] \quad (1a)$$

$$\text{s.t. } A_1 x_1 \ge b_1, \quad (1b)$$

$$A_2\left(\tilde{\xi}_1\right)x_1 + H_2\left(\tilde{\xi}_{[2]}\right)x_2\left(\tilde{\xi}_{[2]}\right) \ge b_2\left(\tilde{\xi}_{[2]}\right), \quad (1c)$$

$$A_t\left(\tilde{\xi}_{[t-1]}\right)x_{t-1}\left(\tilde{\xi}_{[t-1]}\right) + H_t\left(\tilde{\xi}_{[t]}\right)x_t\left(\tilde{\xi}_{[t]}\right) \ge b_t\left(\tilde{\xi}_{[t]}\right) \quad \forall t \in \mathcal{T}\backslash\{1,2\}, \quad (1d)$$

$$x_1 \in \mathbb{R}_+^{n_1-q_1}\times\mathbb{Z}_+^{q_1};\, x_t\left(\tilde{\xi}_{[t]}\right) \in \mathbb{R}_+^{n_t-q_t}\times\mathbb{Z}_+^{q_t} \quad \forall t \in \mathcal{T}\backslash\{1\}. \quad (1e)$$

where $x := (x_1, x_2, \ldots, x_T)$ is the decision vector with $x_t \in \mathbb{R}_+^{n_t-q_t}\times\mathbb{Z}_+^{q_t}$; $A_1 \in \mathbb{R}^{m_t\times n_t}$ and $b_1 \in \mathbb{R}^{m_t}$ are known; and the uncertain parameters realizing as the stochastic process $\tilde{\xi}$ evolves are given by $A_t(\tilde{\xi}) \in \mathbb{R}^{m_t\times n_t}$, $H_t(\tilde{\xi}) \in \mathbb{R}^{m_t\times n_t}$, and $b_t(\tilde{\xi}) \in \mathbb{R}^{m_t}$; $f_t(\cdot) : \mathbb{R}^{n_t} \to \mathbb{R}^{m_t}$ represents a linear finite-valued function for positive integers $m_t$ and $n_t$, i.e., $f(x) = c^T x$ with $c_t(\tilde{\xi}) \in \mathbb{R}_+^{n_t}$; and $\mathbb{E}_{\tilde{\xi}_t|\xi_{[t-1]}}[\bullet]$ is the expectation with respect to $\tilde{\xi}_t$ conditioned on the realization $\xi_{[t-1]}$, for $t = 3, \ldots, T$. Note that for $T = 2$, the M-SMIP (1a)–(1e) is a two-stage stochastic mixed-integer program.

To obtain a mathematical formulation that is more amenable to numerical optimization, we consider a finite number of realizations of a discrete stochastic process $\tilde{\xi} = \left(\tilde{\xi}_1, \ldots, \tilde{\xi}_T\right)$, which has a finite support $\Xi = \left\{\xi^1, \ldots, \xi^S\right\}$ such that $|\Xi| = S$ for some positive integer $S$. A realization of a sequence of random parameters in stages $1, \ldots, T$, $\xi^\omega = \left(\xi_1^\omega, \xi_2^\omega, \ldots, \xi_T^\omega\right) \in \Xi$, is referred to as a scenario indexed by $\omega \in \Omega := \{1, 2, \ldots, S\}$.

The uncertainty in the decision process and the gradual realization of random parameters can be represented using a scenario tree (illustrated in Fig. 1). Each layer of the scenario tree represents the stage-$t$ of the stochastic decision process, and each node of the scenario tree represents a specific realization of the random parameter(s). The unique path from the root node to a leaf (terminal) node corresponds to a scenario. Thus the number of nodes at level $T$ is equal to the number of scenarios $S$. Each scenario $\xi^\omega \in \Xi$ is associated with probability $p^\omega$, which is computed as the product of the conditional probabilities of all nodes that belong to the scenario path $\xi^\omega$, such that $\sum_{\omega\in\Omega} p^\omega = 1$. The realizations of a scenario $\xi^\omega$ up to stage $t$ are denoted by $\xi_{[t]}^\omega$. As depicted in Fig. 1, two scenarios that share the same scenario realization up to stage $t$ (e.g., $\xi_{[3]}^1 = \xi_{[3]}^4$ for scenarios $\xi^1$ and $\xi^4$) share the same decisions up to that stage, which is defined as non-anticipativity constraints.

Denoting $x_t^\omega := x_t\left(\xi_{[t]}^\omega\right)$ as the decision variable for the scenario realization $\xi_{[t]}^\omega$ at stage $t = 1, \ldots, T$, and setting $f(x) = c^T x$, the multi-stage stochastic MIP given in (1) can be equivalently formulated as a large-scale deterministic MIP, denoted as the extensive form of the multi-stage stochastic MIP, as follows:

$$(P)\quad Z = \min \sum_{\omega\in\Omega} p^\omega \sum_{t=1}^{T} c_t^\omega x_t^\omega \quad (2a)$$

$$\text{s.t. } A_1 x_1^\omega \ge b_1 \quad \forall \omega \in \Omega \quad (2b)$$

$$A_t^\omega x_{t-1}^\omega + H_t^\omega x_t^\omega \ge b_t^\omega \quad \forall t \in \mathcal{T}\backslash\{1\},\ \forall \omega \in \Omega \quad (2c)$$

$$x_t^\omega = x_t^{\omega'} \quad \forall t \in \mathcal{T}, \quad \forall \omega, \omega' \in \Omega \ \ s.t.\ \ \xi_{[t]}^\omega = \xi_{[t]}^{\omega'} \quad (2d)$$

$$x_t^\omega \in \mathbb{R}_+^{n_t-q_t}\times\mathbb{Z}_+^{q_t} \quad \forall t \in \mathcal{T} \quad \forall \omega \in \Omega, \quad (2e)$$

where the random parameters become known with $\xi_t^\omega = \left(c_t^\omega, b_t^\omega, A_t^\omega, H_t^\omega\right)$ for each $t \in \mathcal{T}$ and $\omega \in \Omega$. Note that constraints (2d) represent the nonanticipativity constraints, which ensure that, for any stage $t$, decision vectors that correspond to two scenarios $\xi^\omega$ and $\xi^{\omega'}$ should be equal if they are indistinguishable up to stage $t$ (i.e., $\xi_{[t]}^\omega$ and $\xi_{[t]}^{\omega'}$). Since the extensive formulation of the M-SMIP explicitly considers the nonanticipativity constraints, it is called an explicit formulation.

Any multi-stage stochastic MIP defined over a scenario tree can be modeled as in formulation (2). Most practical decision problems involve a large set of scenarios to reflect the uncertainties that occur at successive time stages. The number of scenarios $S$ grows exponentially in the time horizon $T$. Even for relatively small $T$, the dimension of the problem given in (2) can be so large that the whole problem could be impractical to be directly solved by commercial solvers. In the next sections, we discuss the scenario sub-problem that exploits the special structure of explicit formulation given above in decomposing it into scenario sub-problems for developing efficient solution algorithms to solve M-SMIP problems. We then present bounds based on this scenario sub-problem, and the scenario dominance concept based on the dominance relation of any two scenarios and the associated cuts for the general M-SMIPs.

### 2.2. Scenario sub-problem and bounds

In this section, we present a scenario sub-problem, which optimizes a single scenario objective while assigning feasible solutions for all the other variables in the original formulation. Using the solution of the single-scenario sub-problems, we derive lower and upper bounds on M-SMIP, which could be used to strengthen the original formulation's relaxations.

**Definition 2.1** (*Scenario-$\omega$ Problem, Büyüktahtakın, 2022*)**.** The scenario-$\omega$ problem ($P^\omega$) is formulated as follows:

$$(P^\omega)\ Z^\omega = \min \quad p^\omega \sum_{t=1}^{T} c_t^\omega x_t^\omega \quad \text{s.t. (2b), (2c), (2d), (2e).} \quad (3)$$

In Propositions 1 and 2 below, we assume that $f^\omega(x^\omega) = p^\omega \sum_{t=1}^{T} c_t^\omega x_t^\omega$ is a non-negative function for each $\omega \in \Omega$. Since $x^\omega$ are non-negative in the feasible region of M-SMIP (2), this assumption ($f^\omega(x^\omega) \ge 0$) implies that all $c_t^\omega$ are also assumed to be non-negative.

**Proposition 1.** *$P^\omega$ is a relaxation of $P$; $Z \ge Z^\omega$.*

**Proof.** The result follows from the fact that both $P$ and $P^\omega$ have the same feasible region and $\sum_{\omega\in\Omega} p^\omega \sum_{t=1}^{T} c_t^\omega x_t^\omega \ge p^\omega \sum_{t=1}^{T} c_t^\omega x_t^\omega$ for each $\omega \in \Omega$ and for each $x$ that is satisfied by (2b), (2c), (2d), and (2e). □

**Remark 1.** The scenario-$\omega$ problem ($P^\omega$) is an MIP, involving all the variables and all the constraints of formulation (2), while its objective function includes only one specific scenario $\omega$. Thus, the $P^\omega$ is different than the conventional single scenario sub-problem $P_R^\omega$, which has been widely studied in the literature (see, e.g., Madansky, 1960 and Ahmed, 2013). Since the objective function of $P^\omega$ is defined for only one scenario $\omega$, $P^\omega$ is a relaxation of $P$. Our computational results for the lot-sizing and dynamic knapsack problems presented in Section 5 show that the solution time of $P^\omega$ is faster compared to the original problem (2), $P$. However, for some instances for which finding a feasible solution is hard, solving the single scenario sub-problems could take a long time. Thus for such instances, we suggest solving the relaxed version of $P^\omega$, $P_R^\omega$, as shown in Proposition 2 below.

**Definition 2.2** (*Relaxed Scenario-$\omega$ Problem*)**.** The relaxed scenario-$\omega$ problem ($P_R^\omega$) is obtained by removing the non-anticipativity constraints (2d) from the scenario-$\omega$ problem ($P^\omega$) and formulated as follows:

$$(P_R^\omega)\ Z_R^\omega = \min\ p^\omega \sum_{t=1}^{T} c_t^\omega x_t^\omega \quad (4a)$$

$$\text{s.t. } A_1 x_1^\omega \ge b_1, \quad (4b)$$

$$A_t^\omega x_{t-1}^\omega + H_t^\omega x_t^\omega \ge b_t^\omega \quad \forall t \in \mathcal{T}\backslash\{1\}, \quad (4c)$$

$$x_t^\omega \in \mathbb{R}_+^{n_t-q_t}\times\mathbb{Z}_+^{q_t} \quad \forall t \in \mathcal{T}. \quad (4d)$$

**Proposition 2.** *$P_R^\omega$ is a relaxation of $P^\omega$; $Z^\omega \ge Z_R^\omega$. Thus, we have $Z \ge Z_R^\omega$ by Proposition 1.*

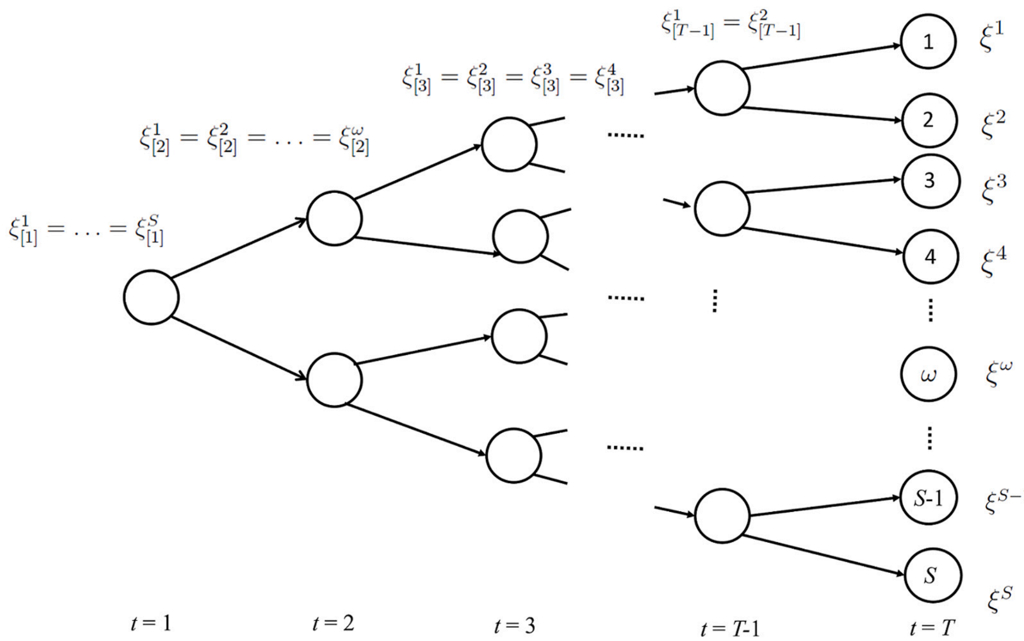


**Fig. 1.** A multi-stage scenario tree and non-anticipativity relations.

**Proposition 3** (*Lower Bound*). *The sum of optimal scenario-$\omega$ problem objective values over all $\omega \in \Omega$, $\sum_{\omega\in\Omega} Z^\omega$, provides a lower bound on the optimal objective function value of the original problem $P$ (2) as in the following inequality:*

$$Z(x^*) \geq \sum_{\omega\in\Omega} Z^\omega. \tag{5}$$

**Proof.** The proof follows from

$$Z(x^*) = Z(x^{1*}, x^{2*}, \ldots, x^{S*}) = \sum_{\omega\in\Omega} Z^\omega(x^{\omega*}) \geq \sum_{\omega\in\Omega} Z^\omega, \tag{6}$$

where $x^* := (x^{1*}, x^{2*}, \ldots, x^{S*})$ is an optimal solution of (2), and the last inequality holds because of the optimality of $Z^\omega$. □

**Proposition 4** (*Upper Bound*). *Let $\dot{x}^\omega$ be the optimal solution for the scenario-$\omega$ problem $P^\omega$ and $Z(\dot{x}^\omega)$ be the objective value of the original problem $P$ where $\dot{x}^\omega$ is substituted in the original problem objective function (1a). Then, the following inequality holds:*

$$Z(x^*) \leq \min_{\omega\in\Omega} Z(\dot{x}^\omega). \tag{7}$$

### *2.3. Scenario dominance*

Below, we describe the concept of scenario dominance, which gives the partial order relations of two scenarios, $\xi^k, \xi^l \in \Xi$, as defined in M-SMIP problem $P$ (2), by pairwise comparing the realization of scenario $\xi^k \in \Xi$ at time $t \in \mathcal{T}$, $\left(\xi^k_2, \xi^k_3, \ldots, \xi^k_T\right)$ with the realization of scenario $\xi^l \in \Xi$, $\left(\xi^l_2, \xi^l_3, \ldots, \xi^l_T\right)$, for each $t \in \mathcal{T}$. Our goal is to derive new bounds and cutting planes to improve the computational solvability of M-SMIPs, using the implications from the partial ordering of scenarios.

**Definition 2.3** (*Scenario Dominance, Büyüktahtakın, 2022*). Define a scenario realization at time $t \in \mathcal{T}$ as $\xi^\omega_t := \left(c^\omega_t, b^\omega_t, A^\omega_t, H^\omega_t\right)$. Given two scenarios $\xi^k$ and $\xi^l$, scenario $\xi^k$ dominates scenario $\xi^l$, denoted by $\xi^l \preceq \xi^k$, with respect to problem $P$ (2) if

$$\left(p^k \geq p^l\right) \wedge \left(c^k_t \geq c^l_t\right) \wedge \left(b^k_t \geq b^l_t\right) \wedge \left(A^k_t \leq A^l_t\right) \wedge \left(H^k_t \leq H^l_t\right) \quad \forall t \in \mathcal{T},$$

where $\wedge$ denotes conjunction or 'and' operator. In other words, scenario domination is determined based on an element-wise comparison of the realizations of $c^\omega_t$, $b^\omega_t$, $A^\omega_t$, and $H^\omega_t$ for $k, l \in \Omega$ at each period $t \in \mathcal{T}$. If, for example, $c^\omega_t := \left(c^\omega_{1,t}, \ldots, c^\omega_{n_t,t}\right)$ and $A^\omega_t := \left(A^\omega_{1,1,t}, \ldots, A^\omega_{m_t,n_t,t}\right)$ are only random variables, then the scenario domination is determined based on comparing the realizations of $c^\omega_t$ and $A^\omega_t$ for $\omega = k, l$, as follows:

$$c^k_{j,t} \geq c^l_{j,t} \quad \forall j = 1, \ldots, n_t, \quad \forall t = 1, \ldots, T,$$

$$A^k_{i,j,t} \leq A^l_{i,j,t} \quad \forall i = 1, \ldots, m_t, \quad \forall j = 1, \ldots, n_t, \quad \forall t = 1, \ldots, T.$$

Using the scenario dominance definition above, we define the dominance sets as follows:

**Definition 2.4** (*Dominance Sets*). The index set of scenarios, which are dominated by scenario $\xi^k \in \Xi$ ($\beta^+_{\xi^k}$), the index set of scenarios, which dominate scenario $\xi^k$ ($\beta^-_{\xi^k}$), and the index set of scenarios, which neither dominate nor are dominated by $\xi^k$ ($N_{\xi^k}$) are defined as follows:

$$\beta^+_{\xi^k} = \left\{l \in \Omega : \xi^l \preceq \xi^k\right\},$$

$$\beta^-_{\xi^k} = \left\{l \in \Omega, l \neq k : \xi^k \preceq \xi^l\right\},$$

$$N_{\xi^k} = \left\{l \in \Omega : \xi^k \npreceq \xi^l \text{ and } \xi^l \npreceq \xi^k\right\}.$$

The definition of scenario dominance above is general as it is based on the uncertainty of all the parameters of the problem: uncertainty in the right-hand side $b^\omega_t$, left-hand side $(A^\omega_t, H^\omega_t)$, and the objective coefficients $c^\omega_t$. We note that the comparison of scenarios for parameter values $(b_t, A_t, H_t, c_t)$ is only relevant if the considered parameter is uncertain. One can derive a large number of dominance relations among scenarios of a stochastic program using Definition 2.3 by considering uncertainty only in the most important parameters of the problem.

In Fig. 2, we present an example realization of the $c_t$ parameter in terms of 12 different scenario realizations. Here each black point $\omega$ represents a realization of $c_t$ in stages 1 and 2, $[c^\omega_2, c^\omega_3]$. The scenario $\xi^6$ realization is $[40, 44]$. The index set of scenarios, which are dominated by $\xi^6$ is defined as $\beta^+_{\xi^6} = \{1, 2, 6\}$, while the index set of scenarios, which dominate scenario $\xi^6$ is given by $\beta^-_{\xi^6} = \{7, 8, 9, 10, 11, 12\}$. The index set for all other scenarios that neither dominate nor are dominated by $\xi^6$ is given by $N_{\xi^6} = \{3, 4, 5\}$.

**Remark 2.** The uncertain data in the above example follows a discrete distribution, and there are only twelve different scenarios. Our scenario

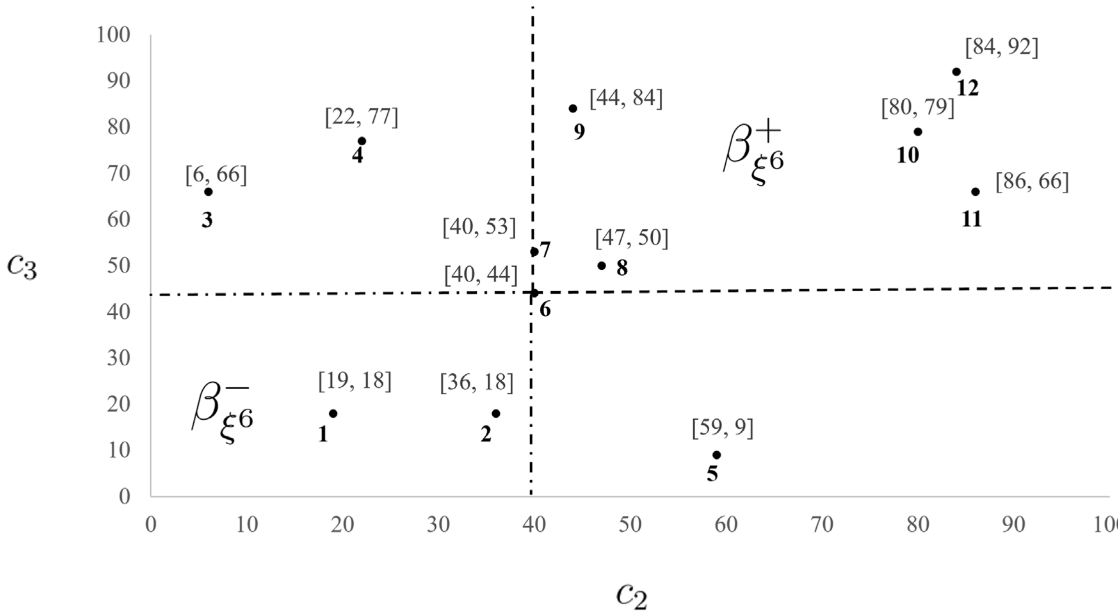


**Fig. 2.** Example of 12 scenario realizations of the $c_t$ variable in three stages and scenarios that dominate and are dominated by scenario $\xi^6$. Each black point $\omega$ represents a scenario realization $\xi^\omega := [c_2^\omega, c_3^\omega]$ for $\omega = 1, \dots, 12$.

dominance approach could also be applied to stochastic programs with random variables that follow continuous distributions by approximating them to scenario-based stochastic programs with discrete distributions through various sampling methods (Shapiro et al., 2009).

### 2.4. Scenario dominance cutting plane algorithm

In this section, we present two lemmas and the main theorem regarding the scenario dominance cutting planes introduced by Büyüktahtakın (2022), focusing on the case for the general risk-neutral M-SMIP problem. Lemma 1 below states that the portion of the optimal objective function value for the original problem (1) that corresponds to the scenario-$\xi^k$ is bounded below by the optimal objective value of the scenario-$\xi^k$ problem, while Lemma 2 below defines the relations of the optimal objective function values of the two scenario subproblems with their dominance relation. Using those lemmas, we present the main theorem on the scenario dominance cut that provides a lower bound on the scenario-$\xi^k$ portion of the original problem's optimal objective value. In the corollaries following the theorem, we first strengthen the scenario dominance cut and then show its feasibility based on a relaxed scenario sub-problem.

**Definition 2.5.** Let $\dot{x}^\omega$ be the optimal solution for the scenario-$\omega$ problem $P^\omega$ (3) and $Z^\omega(\dot{x}^\omega)$ be the corresponding optimal objective value, i.e., $Z^\omega(\dot{x}^\omega) = p^\omega \sum_{t\in\mathcal{T}} c_t^\omega \dot{x}_t^\omega$ and $Z^\omega(\dot{x}^\omega) \cong Z^\omega$.

**Lemma 1.** *The portion of the optimal solution for the original problem (2) that corresponds to scenario-$\xi^k$, $x^{k*}$, satisfies the following inequality:*

$$p^k \sum_{t\in\mathcal{T}} c_t{}^k x_t^k \geq Z^k(\dot{x}^k) \quad \forall k \in \Omega.$$

**Lemma 2.** *Let $\xi^k$ be a scenario such that $k \in \beta^-_{\xi^l} \subset \Omega$ for a given scenario $\xi^l$. Let $\dot{x}^k$ and $\dot{x}^l$ be the optimal solution to the problems $P^k$ and $P^l$, respectively. Then, the optimal objective value of the problem $P^l$ and the optimal objective value of the problem $P^k$ have the following relation:*

$$Z^k(\dot{x}^k) \geq Z^l(\dot{x}^l) \quad \forall k \in \beta^-_{\xi^l}. \tag{8}$$

**Theorem 1.** *Given the pairs of scenarios $\xi^k$ and $\xi^l$ with $\xi^l \preceq \xi^k$, the portion of the optimal solution for the original problem (2) that corresponds to scenario-$\xi^k$, $x^{k*}$, satisfies the following inequality:*

$$p^k \sum_{t\in\mathcal{T}} c_t{}^k x_t^k \geq Z^l(\dot{x}^l) \quad \forall k \in \beta^-_{\xi^l}. \tag{9}$$

**Proof.** The first inequality below follows from Lemma 1, while the second follows from Lemma 2.

$$p^k \sum_{t\in\mathcal{T}} c_t{}^k x_t^{k*} \geq Z^k(\dot{x}^k) \geq Z^l(\dot{x}^l). \quad \square \tag{10}$$

**Remark 3.** In problem $P$ (2), we define $c$ values as non-negative. However, the inequalities (8) and (9) hold even if $c$ values are unrestricted in sign (positive, negative or zero) since $x^\omega \geq 0$ in problem $P$. Suppose that $x^\omega \leq 0$ in $P$ for all $\omega \in \Omega$. Then for any $c$ values (positive, negative or zero), we would have $Z^k(\dot{x}^k) \leq Z^l(\dot{x}^l)$. When both $c$ and $x$ are unrestricted in sign, we may have either $Z^k(\dot{x}^k) \geq Z^l(\dot{x}^l)$ or $Z^k(\dot{x}^k) \leq Z^l(\dot{x}^l)$. Thus, in the latter two cases, the inequalities (8) and (9) may not hold. However, this problem could be remedied by defining additional $y$-variables such that $y = -x \geq 0$, converting the problem into one with a maximum objective, and redefining the inequalities for this maximization problem, as considered in Bushaj et al. (2022a).

In the below corollary of Theorem 1, we present a tighter version of the scenario dominance cuts (9) for the M-SMIP:

**Corollary 1.1.** *The inequality (9) can be strengthened as in the following inequality:*

$$p^l \sum_{t\in\mathcal{T}} c_t{}^l x_t^k \geq Z^l(\dot{x}^l) = p^l \sum_{t\in\mathcal{T}} c_t{}^l \dot{x}_t^l \quad \forall k \in \beta^-_{\xi^l}. \tag{11}$$

**Proof.** Since the solution $x_t^{k*}$ is a part of a feasible solution to the full problem $P$, $\forall\, l \in \Omega$ such that $\xi^l \preceq \xi^k$

$$p^l \sum_{t\in\mathcal{T}} c_t{}^l x_t^{k*} \geq p^l \sum_{t\in\mathcal{T}} c_t{}^l \dot{x}_t^l.$$

But $c_t^k \geq c_t^l$ and $p^k \geq p^l$ $\forall t \in \mathcal{T}$ implies that

$$\bar{Z}(x^{k*}) = p^k \sum_{t\in\mathcal{T}} c_t{}^k x_t^{k*} \geq p^l \sum_{t\in\mathcal{T}} c_t{}^l x_t^{k*}. \quad \square$$

**Remark 4.** The inequalities (9) and their tighter version (11) do not cut off the optimal solution as proven in Theorem 1 and Corollary 1.1, while they may cut off a feasible solution, i.e., they are not valid inequalities for the feasible region of (2). These types of inequalities, e.g., supervalid inequalities (Israeli and Wood, 2002) and Bender's optimality cuts (Benders, 2005), are useful because they help to reduce the solution space while ensuring that the optimal solutions are not cut off. Also, Theorem 1 considers the generalized uncertainty in Problem (2), that is, the uncertainty in the objective function, right-hand side vector, as well as left-hand side matrices simultaneously. Uncertainty in one or a subset of those parameters is a specific case, and thus, the same findings apply to it.

**Corollary 1.2.**

$$p^l \sum_{t\in\mathcal{T}} {c_t}^l x_t^k \geq Z_R^l(\dot{x}^l)$$

**Proof.** The proof follows from Theorem 1 and Corollary 1.1 and the fact that $P_R^l$ is a relaxation of $P^l$ for $l \in \Omega$, i.e., $Z^l(\dot{x}^l) \geq Z_R^l(\dot{x}^l)$. □

In Algorithm 1 below, we provide a formal procedure for generating the scenario dominance cuts (11).

**Algorithm 1** Cutting plane algorithm for scenario dominance cuts (11)

**Procedure: Define Scenario Dominance Sets**
Define: $\bar{\Omega} \subseteq \Omega$; $\beta^-_{\xi^l} = \emptyset$ **for** $l \in \bar{\Omega}$
**for** $l \in \bar{\Omega}$ **do**
  **for** $t \in \mathcal{T}$ **do**
    **for each** $k \in \Omega$, **do**
      **if** $p^k \geq p^l$ and $c_t^k \geq c_t^l$ and $b_t^k \geq b_t^l$ and $A_t^k \leq A_t^l$ and $H_t^k \leq H_t^l$ **then**
        **append** $k$ **to** $\beta^-_{\xi^l}$
      **end if**
    **end for**
  **end for**
**end for**
**Procedure: Add Scenario Dominance Cut**
Solve $P^l$ and obtain $\dot{x}^l$ and $Z^l(\dot{x}^l)$ {*$P^l$: sub-problem for scenario $\xi^l$; $\dot{x}^l$: optimal solution for scenario $\xi^l$; $Z^l(\dot{x}^l)$: objective value of $P^l$*}
Define $x^k$ {*$x^k$: decision variables corresponding to scenario $\xi^l$*}
**while** $\beta^-_{\xi^l}$ is not $\emptyset$, for each $k \in \beta^-_{\xi^l}$ **do**
  select scenario index $k$
  **add cut:** $p^l \sum_{t\in\mathcal{T}} {c_t}^l x_t^k \geq Z^l(\dot{x}^l)$
**end while**

We demonstrate the use of our cuts on the two most popular combinatorial optimization problems: the stochastic capacitated multi-item lot-sizing problem (S-MCLSP) and the stochastic dynamic multi-dimensional knapsack problem (S-MKP). The lot-sizing problem is a typical example of a dynamic MIP, and thus the stochastic version of it is a perfect application of a general multi-stage stochastic MIP program (Guan et al., 2009). On the other hand, the multi-dimensional knapsack problem, which involves multiple knapsack constraints, is strongly NP-Hard (Kellerer et al., 2004) and is also a special case of the general mixed-integer programs. Consequently, we consider the stochastic multi-stage version of the multi-dimensional knapsack to demonstrate the generality of our approach to tackling stochastic MIP problems. In the next two sections, we explore the implementation of the scenario dominance cuts and derive strong scenario dominance cuts to solve S-MCLSP and S-MKP problems, respectively.

## 3. Scenario dominance cuts for S-MCLSP

### *3.1. Stochastic multi-stage lot-sizing problem*

We consider a multi-stage stochastic integer programming formulation of the multi-item capacitated lot-sizing problem (S-MCLSP). Guan et al. (2009) have considered the single item version of the S-MCLSP. The S-MCLSP involves $T$ time periods, where the stochastic demand for each item $i \in \mathcal{I} = \{1,\ldots,I\}$ during time period $t \in \mathcal{T} = \{1,\ldots,T\}$ is given by $d_{it}^{\omega}$ under each scenario $\omega \in \Omega$. An S-MCLSP solution must satisfy periodic demands for each item $i$ without any backlogging subject to capacity constraints at each period $t$, i.e., the total number of items produced to be no more than $c_t$. Excess production of an item $i$ can be stored as inventory.

The costs associated with the S-MCLSP include per-unit item-$i$ production costs ($g_{it}$), fixed setup costs ($f_{it}$) if the production of item $i$ occurs during period $t$, and per-unit inventory costs ($h_{it}$), for each item $i \in \mathcal{I}$ and period $t \in \mathcal{T}$. The objective of the S-MCLSP is to find production levels for each period that satisfy all demands at a minimum cost.

To formulate the S-MCLSP as a stochastic mixed-integer program (MIP), we define the following decision variables for item $i \in \mathcal{I}$, period $t \in \mathcal{T}$, and each scenario $\omega \in \Omega$. Let $x_{it}^{\omega}$ be the amount of item-$i$ production in period $t$, and let $z_{it}^{\omega}$ be the item-$i$ inventory level after period $t$ under scenario $\omega \in \Omega$. Additionally, define binary variables $y_{it}^{\omega}$, which equal 1 if $x_{it}^{\omega} > 0$. The S-MCLSP formulation (12) can be presented as follows:

$$\min \sum_{\omega\in\Omega} p^{\omega} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \left(g_{it}x_{it}^{\omega} + f_{it}y_{it}^{\omega} + h_{it}z_{it}^{\omega}\right) \tag{12a}$$

$$\text{s.t. } z_{i,t-1}^{\omega} + x_{it}^{\omega} - d_{it}^{\omega} = z_{it}^{\omega} \qquad \forall i \in \mathcal{I},\ \forall t \in \mathcal{T},\ \forall \omega \in \Omega \tag{12b}$$

$$\sum_{i\in\mathcal{I}} x_{it}^{\omega} \leq c_t \qquad \forall t \in \mathcal{T},\ \forall \omega \in \Omega \tag{12c}$$

$$x_{it}^{\omega} \leq \min\left\{c_t, \sum_{j=t}^{T} d_{ij}^{\omega}\right\} y_{it}^{\omega} \qquad \forall i \in \mathcal{I},\ \forall t \in \mathcal{T},\ \forall \omega \in \Omega \tag{12d}$$

$$x_{it}^{\omega} = x_{it}^{\omega'} \quad \forall i \in \mathcal{I},\ \forall t \in \mathcal{T},\ \forall \omega,\omega' \in W \ s.t.\ \xi_{[t]}^{\omega} = \xi_{[t]}^{\omega'} \tag{12e}$$

$$z_{it}^{\omega} = z_{it}^{\omega'} \quad \forall i \in \mathcal{I},\ \forall t \in \mathcal{T},\ \forall \omega,\omega' \in W \ s.t.\ \xi_{[t]}^{\omega} = \xi_{[t]}^{\omega'} \tag{12f}$$

$$y_{it}^{\omega} = y_{it}^{\omega'} \quad \forall i \in \mathcal{I},\ \forall t \in \mathcal{T},\ \forall \omega,\omega' \in W \ s.t.\ \xi_{[t]}^{\omega} = \xi_{[t]}^{\omega'} \tag{12g}$$

$$x_{it}^{\omega}, z_{it}^{\omega} \geq 0 \qquad \forall i \in \mathcal{I},\ \forall t \in \mathcal{T},\ \forall \omega \in \Omega \tag{12h}$$

$$y_{it}^{\omega} \in \{0,1\} \qquad \forall i \in \mathcal{I},\ \forall t \in \mathcal{T},\ \forall \omega \in \Omega. \tag{12i}$$

The objective function (12a) minimizes the expected sum of production, setup, and inventory costs over all periods $t \in \mathcal{T}$ and items $i \in \mathcal{I}$. Constraints (12b) define the inventory balance constraints and the inventory remaining after each time period for each item. Constraints (12c) enforce the periodic capacity over the production of all items. Constraints (12d) represent the capacity constraints by enforcing the binary setup variable $y_{it}^{\omega}$ to equal 1 whenever $x_{it}^{\omega}$ is positive, and the maximum production for each item in each period can be, at most, the minimum of the capacity and the total remaining demand for that item. Constraints (12e), (12f), and (12g) represent the nonanticipativity constraints for production, inventory, and setup variables, respectively. Finally, constraints (12h) and (12i) represent lower bounds on the production and inventory variables and binary integer restrictions, respectively. Without loss of generality, we assume zero initial inventory.

### *3.2. Bounds for S-MCLSP*

In this sub-section, we adapt the lower (5) and upper bounds (7) proposed for the general M-SMIPs to the S-MCLSP problem. Our goal is to improve the solvability of the S-MCLSP by implementing those bounds in a branch and bound solver.

**Proposition 5.** *Let $\bar{\Omega} \subseteq \Omega$ be a subset of scenarios. For a given $s \in \bar{\Omega}$, let $(\dot{x}_{it}^s, \dot{y}_{it}^s, \dot{z}_{it}^s)$ and $Z^s(\dot{x}_{it}^s, \dot{y}_{it}^s, \dot{z}_{it}^s)$ be the optimal solution and the corresponding objective function value of the scenario-$\xi^s$ problem, respectively. Let $Z(\dot{x}_{it}^s, \dot{y}_{it}^s, \dot{z}_{it}^s)$ be the objective function value of (12a) at the optimal solution of the scenario-$\xi^s$ problem, $(\dot{x}_{it}^s, \dot{y}_{it}^s, \dot{z}_{it}^s)$. Then the optimal solution to the S-MCLSP, $(x^*, y^*, z^*)$, satisfies*

(a) *the following lower bound on the optimal objective function value of the S-MCLSP problem (12):*

$$\sum_{\omega\in\Omega} p^{\omega} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \left(g_{it}x_{it}^{\omega} + f_{it}y_{it}^{\omega} + h_{it}z_{it}^{\omega}\right) \geq \sum_{s\in\bar{\Omega}} Z^{s}(\dot{x}_{it}^{s}, \dot{y}_{it}^{s}, \dot{z}_{it}^{s}), \tag{13}$$

(b) *and the following upper bound on the optimal objective function value of the S-MCLSP problem (12):*

$$\sum_{\omega\in\Omega} p^{\omega} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \left(g_{it}x_{it}^{\omega} + f_{it}y_{it}^{\omega} + h_{it}z_{it}^{\omega}\right) \leq \min_{s\in\bar{\Omega}} Z(\dot{x}_{it}^{s}, \dot{y}_{it}^{s}, \dot{z}_{it}^{s}). \tag{14}$$

**Proof.** The result follows immediately by applying inequality (5) in Proposition 3 for a subset of scenarios $\bar{\Omega} \subseteq \Omega$ and inequality (7) in Proposition 4, respectively, for the stochastic multi-stage multi-item lot-sizing problem. □

### 3.3. Scenario-dominance cuts for S-MCLSP

Under this sub-section, we modify the scenario dominance cutting planes (9) for the S-MCLSP to improve the solvability of such problems. Below, we present the related theorem.

**Theorem 2.** *Let $\xi^l$ be a scenario with $l \in \Omega$. For each $t \in \mathcal{T}$ and $i \in \mathcal{I}$, let $(\dot{x}_{it}^{l}, \dot{y}_{it}^{l}, \dot{z}_{it}^{l})$ represent a vector of optimal solutions to scenario-$\xi^l$ problem and define the optimal objective value corresponding to the scenario-$\xi^l$ problem as:*

$$Z^{l}(\dot{x}_{it}^{l}, \dot{y}_{it}^{l}, \dot{z}_{it}^{l}) = p^{l} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \left(g_{it}\dot{x}_{it}^{l} + f_{it}\dot{y}_{it}^{l} + h_{it}\dot{z}_{it}^{l}\right).$$

*Then, the portion of the optimal solution to the S-MCLSP (12) corresponding to scenario $\xi^k$ such that $\xi^l \preceq \xi^k$ and $l \neq k$, $(x_{it}^{k*}, y_{it}^{k*}, z_{it}^{k*})$, satisfies the following inequality:*

$$p^{k} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \left(g_{it}x_{it}^{k} + f_{it}y_{it}^{k} + h_{it}z_{it}^{k}\right) \geq Z^{l}(\dot{x}_{it}^{l}, \dot{y}_{it}^{l}, \dot{z}_{it}^{l}) \qquad \forall k \in \beta_{\xi^l}^{-}. \tag{15}$$

**Proof.** The result follows immediately by applying the inequality (9) in Theorem 1 for the stochastic multi-stage multi-item capacitated lot-sizing problem. □

### 3.4. Strong scenario-dominance cuts for S-MCLSP

Here, as a main contribution of the paper, we present new scenario dominance cuts for the S-MCLS, which are stronger and more aggressive than the inequalities (9). Below we provide two propositions and their proofs for the strong scenario dominance cuts and an example to illustrate their application.

**Proposition 6.** *Let $\xi^l$ be a scenario with $l \in \Omega$. Let $\dot{y}_{it}^{l}$ represent a vector of optimal y-variable solutions to the scenario-$\xi^l$ problem (3). Then, for a given $t \in \mathcal{T}$ and for each $\xi^k$, $k \in \Omega$, such that $\xi^l \preceq \xi^k$, there exists a feasible solution to the S-MCLSP problem (12), $\hat{y}_{ih}^{k}$, that satisfies the following inequality:*

$$\sum_{h=1}^{t} y_{ih}^{k} \geq \sum_{h=1}^{t} \dot{y}_{ih}^{l}, \quad \forall i \in \mathcal{I}, \quad \forall k \in \beta_{\xi^l}^{-}. \tag{16}$$

**Proof.** We will prove this by contradiction. Assume that for some $i \in \mathcal{I}$, there exists a period $t \in \mathcal{T}$ such that $\sum_{h=1}^{t} \dot{y}_{ih}^{l} > \sum_{h=1}^{t} \hat{y}_{ih}^{k}$ is satisfied by each feasible solution $\hat{y}_{ht}^{k}$ to the original S-MCLSP problem. Then there exists at least one period $j \in \{1, \ldots, t\}$ such that $\dot{y}_{ij}^{l} > \hat{y}_{ij}^{k}$. In that case, $\dot{y}_{ij}^{l}$ must take the value 1. Define $\tilde{T} = \left\{ j \in [1, t] : \dot{y}_{ij}^{l} > \hat{y}_{ij}^{k} \right\}$ and order the elements of $\tilde{T}$ in ascending sequence, i.e., $\tilde{T} = \left\{ j_1, j_2, \ldots, j_m \right\}$.

As capacity is sufficiently large to produce in period $j_n$, $n = 1, \ldots, m$, under scenario $l$, i.e., $\dot{y}_{ij_n}^{l} = 1$ and $x_{ij_n}^{l} > 0$, a new solution to the original problem under scenario $k$ can be constructed iteratively for each $j_n \in \tilde{T}$ as presented in Algorithm 2 given below.

**Algorithm 2** Finding a feasible solution $(\acute{y}, \acute{x}, \acute{z})$ satisfying $\sum_{h=1}^{t} \dot{y}_{ih}^{l} \leq \sum_{h=1}^{t} \acute{y}_{ih}^{k}$

Set the initial values of $(\acute{y}, \acute{x}, \acute{z})$ to $(\hat{y}, \hat{x}, \hat{z})$ satisfying $\sum_{h=1}^{t} \dot{y}_{ih}^{l} > \sum_{h=1}^{t} \hat{y}_{ih}^{k}$ for each $i \in \mathcal{I}$:

$$\begin{aligned} \acute{y}_{ij}^{k} &= \hat{y}_{ij}^{k}, \ j = 1, \ldots, T, \\ \acute{x}_{ij}^{k} &= \hat{x}_{ij}^{k}, \ j = 1, \ldots, T, \\ \acute{z}_{ij}^{k} &= \hat{z}_{ij}^{k}, \ j = 1, \ldots, T. \end{aligned}$$

**for** $n = 1$ to $m$ **do**

$$\begin{aligned} \acute{y}_{ij_n}^{k} &= \dot{y}_{ij_n}^{l}, \\ \acute{x}_{ij_n}^{k} &= \acute{x}_{ij_n}^{k} + 1, \\ \acute{z}_{it}^{k} &= \acute{z}_{it}^{k} + 1, \ t = j_n, \ldots, T. \end{aligned}$$

**end for**

**for** $v \in \Omega$ such that $\xi_{[t]}^{v} = \xi_{[t]}^{k}$ for some $t \in \mathcal{T}$ **do**

$$\begin{aligned} \acute{y}_{ij}^{v} &= \acute{y}_{ij}^{k}, \ j = 1, \ldots, t, \\ \acute{x}_{ij}^{v} &= \acute{x}_{ij}^{k}, \ j = 1, \ldots, t, \\ \acute{z}_{ij}^{v} &= \acute{z}_{ij}^{k}, \ j = 1, \ldots, t, \\ \acute{z}_{ij}^{v} &= \acute{z}_{ij}^{v} + m, \ j = t+1, \ldots, T. \end{aligned}$$

**end for**

In the above algorithm, a new feasible solution $(\acute{y}, \acute{x}, \acute{s})$ satisfying $\sum_{h=1}^{t} \dot{y}_{ih}^{l} \leq \sum_{h=1}^{t} y_{ih}^{k}$ is obtained by modifying an initial feasible solution $(\hat{y}, \hat{x}, \hat{s})$ that satisfies $\sum_{h=1}^{t} \dot{y}_{ih}^{l} > \sum_{h=1}^{t} \hat{y}_{ih}^{k}$. The initial solution is iteratively updated for each $j_n \in \tilde{T}$, $n = 1, \ldots, m$ starting from $j_1$ in ascending order. In order to satisfy the non-anticipativity constraints (12e)–(12g), the solutions corresponding to a scenario $v$ which shares the same $y-$, $x-$, and $z-$ variables with scenario $k$ up to time $t \in \mathcal{T}$, i.e., $\xi_{[t]}^{v} = \xi_{[t]}^{k}$, with $z-$ variables for scenario $v$ at time $j = t+1, \ldots, T$ are also updated.

The new solution is non-negative and also satisfies the inventory balance constraints (12b) as the inventory at time $j_n \in \tilde{T}$ is increased as much as the production at time $j_n \in \tilde{T}$. Additional production at time $j_n$ is carried over as inventory in future periods without being used for demand. This solution also meets the capacity limits (12c) and (12d) because a setup value of $\dot{y}_{ij_n}^{l} = 1$, and thus, the production amount of $x_{ij_n}^{l} > 0$ is feasible under the scenario $l$ problem, and thus is also feasible for the scenario $k$ problem. Thus, we obtain a new solution deriving a contradiction to our initial assumption. □

**Proposition 7.** *Let $\xi^k$ be a scenario with $k \in \Omega$ and $\dot{x}_{it}^{k}$ represent a vector of x-variable optimal solutions to the scenario-$\xi^k$ problem (3). Then, for a given $t \in \mathcal{T}$ and for each $\xi^l$, $l \in \Omega$ such that $\xi^l \preceq \xi^k$, there exists a feasible solution to the S-MCLSP problem (12), $\hat{x}_{it}^{l}$, that satisfies the following inequality:*

$$\sum_{j=1}^{t} x_{it}^{l} \leq \sum_{j=1}^{t} \dot{x}_{it}^{k}, \quad \forall t \in \mathcal{T}, i \in \mathcal{I}, \quad \forall l \in \beta_{\xi^k}^{+}. \tag{17}$$

**Proof.** This is a proof by construction. Without loss of generality, assume that the original problem (12) is feasible, and let $(\hat{y}, \hat{x}, \hat{z})$ be a feasible solution to this problem. Because $d_t^k \geq d_t^l$ for each $t \in \mathcal{T}$, $(\hat{y}, \hat{x}, \hat{z})$ can be modified to construct another feasible solution to the original problem iteratively such that $\sum_{h=1}^{t} \hat{x}_{ih}^{l} = \sum_{h=1}^{t} \dot{x}_{ih}^{k}$ as presented in Algorithm 3 given below.

**Algorithm 3** Modifying a feasible solution $(\hat{y}, \hat{x}, \hat{z})$ to satisfy $\sum_{h=1}^{t} \hat{x}_{ih}^{l} = \sum_{h=1}^{t} \dot{x}_{ih}^{k}$

**for** $t = 1$ to $T$ **do**

$$\hat{y}_{it}^{l} = \dot{y}_{it}^{k},$$
$$\hat{x}_{it}^{l} = \dot{x}_{it}^{k},$$
$$\hat{z}_{it}^{l} = \dot{z}_{it}^{k} + (\sum_{j=1}^{t} d_{ij}^{k} - \sum_{j=1}^{t} d_{ij}^{l}).$$

**end for**

**for** $v \in \Omega$ such that $\xi_{[t]}^{v} = \xi_{[t]}^{l}$ for some $t \in \mathcal{T}$ **do**

$$\hat{y}_{ij}^{v} = \hat{y}_{ij}^{l}, \quad j = 1, \dots, t,$$
$$\hat{x}_{ij}^{v} = \hat{x}_{ij}^{l}, \quad j = 1, \dots, t,$$
$$\hat{z}_{ij}^{v} = \hat{z}_{ij}^{l}, \quad j = 1, \dots, t,$$
$$\hat{z}_{ij}^{v} = \hat{x}_{ij}^{v} - d_{ij}{}^{v} + \hat{z}_{i,j-1}^{v} \quad j = t+1, \dots, T.$$

**end for**

In the above algorithm, a new feasible solution satisfying $\sum_{h=1}^{t} \hat{x}_{ih}^{l} = \sum_{h=1}^{t} \dot{x}_{ih}^{k}$ is obtained by modifying an initial feasible solution to the original problem, $(\hat{y}, \hat{x}, \hat{s})$. The initial solution is iteratively updated for each scenario $l$, such that $\xi^{l} \preceq \xi^{k}$ for each $t \in \mathcal{T}$. In order to satisfy the non-anticipativity constraints (12e)–(12g), the solutions corresponding to scenario $v \in \Omega$, which share the same $y-$, $x-$, and $z-$ variables with scenario $l$ up to time $t \in \mathcal{T}$, and $z-$ variables after time $t$ are also updated. Note that $v \in \Omega$ such that $\xi^{v} \preceq \xi^{l}$ up to stage $t \in \mathcal{T}$ is also dominated by scenario $k$ up to stage $t \in \mathcal{T}$, due to the transitivity property of scenario dominance.

The new solution is non-negative and also satisfies the inventory balance constraints (12b) because the difference in total demands of scenario $k$ and $l$ is added to the inventory of scenario $l$ at time $t \in \mathcal{T}$. This solution also meets the capacity limits (12c) and (12d) because the production amount of $\dot{x}_{it}^{k} \geq 0$ is feasible under scenario $k$, and thus is also feasible for scenario $l$. □

**Remark 5.** The inequalities (16) and (17) generally lead to optimal solutions, while in some cases, those inequalities might cut off the optimal and even a feasible solution. By adding such aggressive cuts as user cuts, our goal is to considerably speed up the solution time without causing infeasibility. Our computational experiments have shown that the best solution obtained by adding (16) and (17) only deviates 0.05% on average from the optimal solution for the MCLSP instances, while the solution time has been significantly reduced (see Section 5.3). Therefore, strong scenario dominance cuts need to be carefully handled for implementation. In Section 5, we investigate the tradeoff between the quality of the best solution obtained by applying those inequalities versus the computational speed-up obtained with them. We, specifically, present the objective value gap defined by the percentage deviation between the best objective value found by CPLEX for the original problem and the best objective value obtained by (16) and (17) cuts to analyze the impact of adding aggressive inequalities that may cut off the optimal solution.

**Example 1.** Consider an S-MCLSP instance in which $T = 4$ and $I = 2$. This instance has $S = 8$ scenarios, and $\Omega = \{1, 2, 3, 4, 5, 6, 7, 8\}$. We assume that the parameter values at $t = 1$ are known and are set as zero in this example. The data pertaining the instance for $i = 1, 2$ and $t = 2, 3, 4$ is: $c_t = (21, 24, 24)$, $g_{1t} = (27, 27, 34)$, $g_{2t} = (41, 38, 42)$, $f_{1t} = (1070, 1096, 991)$, $f_{2t} = (928, 940, 1041)$, $h_{1t} = (1, 1, 1)$, $h_{2t} = (1, 1, 1)$ and the uncertain demand parameter $d_{it}^{1} = [(5, 3, 5), (1, 3, 1)]$, $d_{it}^{2} = [(5, 3, 6), (1, 3, 9)]$, $q_{it}^{3} = [(5, 6, 5), (1, 10, 1)]$, $d_{it}^{4} = [(5, 6, 6), (1, 10, 9)]$, $d_{it}^{5} = [(8, 3, 5), (6, 3, 1)]$, $d_{it}^{6} = [(8, 3, 6), (6, 3, 9)]$, $d_{it}^{7} = [(8, 6, 5), (6, 10, 1)]$, $d_{it}^{8} = [(8, 6, 6), (6, 10, 9)]$.

Now, let $\bar{\Omega} = \{1, 2, 4, 6, 8\}$. Then $Z^{1} = 321.5$, $Z^{2} = 606.5$, $Z^{4} = 484.25$, $Z^{6} = 642.875$, and $Z^{8} = 819.25$. Then the lower-bound inequality (13) is:

$$\sum_{\omega \in \Omega} p^{\omega} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}} \left( g_{it} x_{it}^{\omega} + f_{it} y_{it}^{\omega} + h_{it} z_{it}^{\omega} \right) \geq 2874.4 \tag{18}$$

and for $\bar{\Omega}$, the upper-bound inequality (14) is:

$$\sum_{\omega \in \Omega} p^{\omega} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}} \left( g_{it} x_{it}^{\omega} + f_{it} y_{it}^{\omega} + h_{it} z_{it}^{\omega} \right) \leq 6433. \tag{19}$$

Consider scenario $\xi^{5}$. The set of scenarios that are dominated by scenario $\xi^{5}$ is defined as $\beta_{\xi^5}^{+} = \{1, 5\}$, and the set of scenarios that dominate scenario $\xi^{5}$ is defined as $\beta_{\xi^5}^{-} = \{6, 7, 8\}$. Solving the scenario-$\xi^{5}$ problem, we have $Z^{5} = 484.25$. Thus, for each $k \in \beta_{\xi^5}^{-}$, the inequality (15) can be written as:

$$p^{k} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}} \left( g_{it} x_{it}^{k} + f_{it} y_{it}^{k} + h_{it} z_{it}^{k} \right) \geq 484.25.$$

The optimal objective function value of S-MCLSP (12) for this instance is 4969.75, and the linear programming relaxation value is 4134.65. By adding inequalities (13), (14), and (15) for $\bar{\Omega} = \{1, 2, 4, 6, 8\}$, the optimal relaxation solution is cut off, and the relaxation objective of (12) increases to 4587.81, thus the initial optimality gap is closed by 54.26%.

For $k = 6 \in \beta_{\xi^5}^{-}$, the inequalities corresponding to (16) for $t = 4$ and $i = 1$ can be written as:

$$y_{12}^{6} + y_{13}^{6} + y_{14}^{6} \geq 2.$$

For $l = 1 \in \beta_{\xi^5}^{+}$, the inequalities corresponding to (17) for $t = 4$ and $i = 1$ can be written as:

$$x_{12}^{1} + x_{13}^{1} + x_{14}^{1} \leq 16.$$

## 4. Scenario dominance cuts for S-MKP

### 4.1. The multi-stage stochastic knapsack problem

The first benchmark set corresponds to a class of stochastic multi-stage multi-dimensional mixed 0–1 knapsack problems (S-MKP). The deterministic and single-dimensional version of the problem has been studied by Marchand and Wolsey (1999). Mixed 0–1 knapsack formulation is important because many integer programs have been shown to be equivalent to it or include it in the form of capacity constraints. Thus, solution procedures for knapsack models can be adopted to solve general integer programs.

To formulate the S-MKP as a stochastic mixed-integer program (MIP), we define the following deterministic and stochastic parameters. Let $q_t^{\omega}$ be the cost coefficient associated with variable $y_t^{\omega}$ in period $t$ under scenario $\omega \in \Omega$. Also, let $c_{it}$ represent the cost coefficient associated with variable $x_{it}^{\omega}$ for item $i$ in period $t$, $d_t$ be the cost coefficient associated with variable $z_t^{\omega}$ in period $t$, $a_{it}$ be the value of item $i$ in period $t$, $r_t$, and $w_t$ be the coefficients corresponding to the continuous variable $y_t^{\omega}$ in period $t$, respectively associated with the budget constraints related to value and size of the knapsack, $b_t$ be the minimum total value of all selected items in period $t$, $v_{it}$ be the size of item $i$ in period $t$, and $h_t$ be the minimum total size of all selected items in period $t$.

We also define the following decision variables for period $t \in \mathcal{T}$ and each scenario $\omega \in \Omega$. Let $x_{it}^{\omega}$ be a binary variable to select an item $i$ in period $t$ and under scenario $\omega \in \Omega$, and let $z_t^{\omega}$ be a continuous variable in period $t$ under scenario $\omega \in \Omega$. Additionally, define binary variables

$y_t^\omega$, which are related to the binary $x_{it}^\omega$ variables. The $z_t^\omega$ and $y_t^\omega$ are defined to represent compensation values in the budget constraints. We assume all coefficients are non-negative w.l.o.g. Here, $q_t^\omega$ is the only (1-dimensional) random parameter here. The S-MKP (20) can be formulated as follows.

$$\min \sum_{\omega\in\Omega} p^\omega \sum_{t\in\mathcal{T}} \left( \sum_{i\in\mathcal{I}} c_{it}x_{it}^\omega + d_t z_t^\omega + q_t^\omega y_t^\omega \right) \tag{20a}$$

$$\text{s.t.} \sum_{j=1}^{t}\sum_{i\in\mathcal{I}} a_{ij}x_{ij}^\omega + r_t z_t^\omega \ge b_t \qquad \forall t\in\mathcal{T},\ \forall\omega\in\Omega \tag{20b}$$

$$\sum_{i\in\mathcal{I}} v_{it}x_{it}^\omega + w_t y_t^\omega \ge h_t \qquad \forall t\in\mathcal{T},\ \forall\omega\in\Omega \tag{20c}$$

$$x_{it}^\omega = x_{it}^{\omega'} \quad \forall\ i\in\mathcal{I},\ \forall t\in\mathcal{T}, \forall\omega,\omega'\in\Omega \text{ s.t. } \xi_{[t]}^\omega = \xi_{[t]}^{\omega'} \tag{20d}$$

$$z_t^\omega = z_t^{\omega'} \quad \forall t\in\mathcal{T}, \forall\omega,\omega'\in\Omega \text{ s.t. } \xi_{[t]}^\omega = \xi_{[t]}^{\omega'} \tag{20e}$$

$$y_t^\omega = y_t^{\omega'} \quad \forall t\in\mathcal{T}, \forall\omega,\omega'\in\Omega \text{ s.t. } \xi_{[t]}^\omega = \xi_{[t]}^{\omega'} \tag{20f}$$

$$z_t^\omega \ge 0 \qquad \forall t\in\mathcal{T}, \forall\omega\in\Omega \tag{20g}$$

$$x_{it}^\omega, y_t^\omega \in \{0,1\} \qquad \forall i\in\mathcal{I}, \forall t\in\mathcal{T},\ \forall\omega\in\Omega. \tag{20h}$$

The objective function (20a) minimizes the expected sum of knapsack costs over all items $i \in \mathcal{I}$, periods $t \in \mathcal{T}$, and scenarios $\omega \in \Omega$. Constraints (20b) and (20c) define knapsack-related budget constraints, where constraints (20b) and (20c) ensure that the sum of the value and size of selected items, respectively, should be larger than a given constant. Constraints (20d), (20e), and (20f) represent the nonanticipativity constraints for the $x$, $z$, and $y$ variables, respectively. Finally, constraints (20g) and (20h) represent lower bounds on the $z$ variables and binary integer restrictions on $x$ and $y$ variables, respectively.

### 4.2. Bounds for S-MKP

Here, we apply the M-SMIP lower (5) and upper bounds (7) to the case of S-MKP to strengthen the polyhedral representation of the S-MKP problems.

**Proposition 8.** *Let $\bar{\Omega} \subseteq \Omega$ be a subset of scenarios. For a given $s \in \bar{\Omega}$, let $(\dot{x}^s, \dot{z}^s, \dot{y}^s)$ and $Z^s(\dot{x}^s, \dot{z}^s, \dot{y}^s)$ be the optimal solution and the optimal objective function value of the scenario-$\xi^s$ problem, respectively. Let $Z(\dot{x}^s, \dot{z}^s, \dot{y}^s)$ be the objective function value of (20a) at the optimal solution of the scenario-$\xi^s$ problem, $(\dot{x}^s, \dot{z}^s, \dot{y}^s)$. Then the optimal solution to the S-MKP, $(x^*, z^*, y^*)$, satisfies*

(a) *the following lower bound on the optimal objective function value of the S-MKP problem (20):*

$$\sum_{\omega\in\Omega} p^\omega \sum_{t\in\mathcal{T}} \left( \sum_{i\in\mathcal{I}} c_{it}x_{it}^\omega + d_t z_t^\omega + q_t^\omega y_t^\omega \right) \ge \sum_{s\in\bar{\Omega}} Z^s(\dot{x}^s, \dot{z}^s, \dot{y}^s), \tag{21}$$

(b) *and the following upper bound on the optimal objective function value to the S-MKP problem (20):*

$$\sum_{\omega\in\Omega} p^\omega \sum_{t\in\mathcal{T}} \left( \sum_{i\in\mathcal{I}} c_{it}x_{it}^\omega + d_t z_t^\omega + q_t^\omega y_t^\omega \right) \le \min_{s\in\bar{\Omega}} Z(\dot{x}^s, \dot{z}^s, \dot{y}^s). \tag{22}$$

**Proof.** The result follows immediately by applying inequality (5) in Proposition 3 for a subset of scenarios $\bar{\Omega} \subseteq \Omega$ and inequality (7) in Proposition 4, respectively, for the stochastic multi-stage knapsack problem. □

### 4.3. Scenario-dominance cuts for S-MKP

Under this sub-section, we adapt the dominance cutting planes (9) to the case of S-MKP to provide faster and better solutions to this problem. We present the related theorem below.

**Theorem 3.** *Let $\xi^l$ be a scenario with $l \in \Omega$. For each $t \in \mathcal{T}$ and $i \in \mathcal{I}$, let $(\dot{x}_i^l, \dot{z}^l, \dot{y}^l)$ represent a vector of optimal solutions to scenario-$\xi^l$ problem and define the optimal objective value corresponding to the scenario-$\xi^l$ problem as:*

$$Z^l(\dot{x}_i^l, \dot{z}^l, \dot{y}^l) = p^l \sum_{t\in\mathcal{T}} \left( \sum_{i\in\mathcal{I}} c_{it}\dot{x}_{it}^l + d_t\dot{z}_t^l + q_t^l\dot{y}_t^l \right). \tag{23}$$

*Then, the portion of the optimal solution to the S-MKP Problem (20) corresponding to scenario $\xi^k$ such that $\xi^l \preceq \xi^k$ and $l \ne k$, $(x_{it}^{k*}, y_t^{k*}, z_t^{k*})$, is satisfied by the following inequality:*

$$p^k \sum_{t\in\mathcal{T}} \left( \sum_{i\in\mathcal{I}} c_{it}x_{it}^k + d_t z_t^k + q_t^k y_t^k \right) \ge Z^l(\dot{x}_i^l, \dot{z}^l, \dot{y}^l) \quad \forall k \in \beta_{\xi^l}^-.$$

**Proof.** The result follows immediately by applying the inequality (9) in Theorem 1 for the stochastic multi-stage knapsack problem. □

**Corollary 3.1.** *The inequality (24) can be strengthened as in the following inequality:*

$$p^k \sum_{t\in\mathcal{T}} \left( \sum_{i\in\mathcal{I}} c_{it}x_{it}^k + d_t z_t^k + q_t^k y_t^k \right) \ge Z^l(\dot{x}_{it}{}^l, \dot{z}_t{}^l, \dot{y}_t{}^l) \quad \forall k \in \beta_{\xi^l}^-. \tag{24}$$

**Proof.** The result follows by applying the inequality (11) in Corollary 1.1 for the S-MKP. □

### 4.4. Strong scenario-dominance cuts for S-MKP

In this sub-section, we present new strong scenario dominance cuts for the S-MKP. We provide the following two propositions and their proofs for those strong and aggressive dominance cuts, as well as an example implementation of them.

**Proposition 9.** *Given $(\dot{x}_{it}^l, \dot{z}_t^l, \dot{y}_t^l)$, the optimal objective value corresponding to the scenario-$\xi^l$ problem for stage $t \in \mathcal{T}$ as:*

$$Z_t^l(\dot{x}_{it}^l, \dot{z}_t^l, \dot{y}_t^l) = p^l \left( \sum_{i\in\mathcal{I}} c_{it}\dot{x}_{it}^l + d_t\dot{z}_t^l + q_t^l\dot{y}_t^l \right).$$

*Then, there exists a feasible solution to the S-MKP Problem (20) corresponding to scenario $\xi^k$ such that $\xi^l \preceq \xi^k$ and $l \ne k$, $(\hat{x}_{it}^k, \hat{y}_t^k, \hat{z}_t^k)$, that satisfies the following inequality for each $t \in \mathcal{T}$:*

$$p^k \left( \sum_{i\in\mathcal{I}} c_{it}x_{it}^k + d_t z_t^k + q_t^k y_t^k \right) \ge Z_t^l(\dot{x}_{it}^l, \dot{z}_t^l, \dot{y}_t^l) \quad \forall k \in \beta_{\xi^l}^-. \tag{25}$$

**Proof.** There exists a feasible solution to the S-MKP Problem (20) that is satisfied by inequalities (25), because those inequalities provide a lower bound on the $x$-, $z$-, and $y$-variables, which are not bounded above. □

**Proposition 10.** *Let $\xi^k$ be a scenario with $k \in \Omega$. For each $t \in \mathcal{T}$ and $i \in \mathcal{I}$, let $\dot{y}_t^k$ represent a vector of optimal $y$-variable solutions to scenario-$\xi^k$ problem. Then, there exists a feasible solution to the S-MKP Problem (20) corresponding to scenario $\xi^l$ such that $\xi^l \preceq \xi^k$ and $l \ne k$, $\hat{y}_t^l$, that satisfies the following inequality for each $t \in \mathcal{T}$:*

$$\sum_{j=1}^{t} q_j^l y_t^l \ge \sum_{j=1}^{t} q_j^k \dot{y}_j^k \quad \forall l \in \beta_{\xi^k}^+. \tag{26}$$

**Proposition 11.** *Let $\xi^k$ be a scenario with $k \in \Omega$. For each $t \in \mathcal{T}$ and $i \in \mathcal{I}$, let $\dot{z}_t^k$ represent a vector of optimal $z$-variable solutions to scenario-$\xi^k$ problem. Then, there exists a feasible solution to the S-MKP Problem (20) corresponding to scenario $\xi^l$ such that $\xi^l \preceq \xi^k$ and $l \ne k$, $\hat{z}_t^l$, that satisfies the following inequality for each $t \in \mathcal{T}$:*

$$\sum_{j=1}^{t} z_t^l \ge \sum_{j=1}^{t} \dot{z}_j^k \quad \forall l \in \beta_{\xi^k}^+. \tag{27}$$

**Remark 6.** The proof idea for both Propositions 10 and 11 is the same as that of Proposition 9 because the inequalities (26) and (27) provide a lower bound on the $y$- and $z$-variables, respectively, and there are no upper bounds on those variables.

**Remark 7.** Similar to the strong scenario dominance cuts defined for the S-MCLSP instances, the inequalities (25), (26), and (27) generally lead to optimal solutions, while they may cut off the optimal solution. In the latter case, our computational experiments have shown that the best solution obtained by adding (25), (26), and (27) only deviates 0.17% on average from the optimal solution for the S-MKP instances, while the solution time has been significantly reduced (see Section 5.3). Those cuts are also shown to improve the optimality gap of CPLEX by 0.13% for S-MKP instances with up to 81 random variables (see Section 5.5).

**Example 2.** Consider an S-MKP instance in which $T = 4$ and $I = 2$. This instance has $S = 8$ scenarios; $\Omega = \{1,2,3,4,5,6,7,8\}$. The data pertaining the instance for $t = 2,3,4$ is: $c_{1t} = (16,98,19)$, $c_{2t} = (78,92,99)$, $d_t = (80,86,85)$, $a_{1t} = (8,19,43)$, $a_{2t} = (41,37,99)$, $v_{1t} = (49,90,66)$, $v_{2t} = (74,15,7)$, $r_t = (45,78,73)$, $w_t = (28,59,97)$, $b_t = (70.5,137.3,240.0)$, $h_t = (113.3,123.0,127.5)$, and the uncertain parameter $q_t^1 = (14,45,38)$, $q_t^2 = (14,45,68)$, $q_t^3 = (14,79,38)$, $q_t^4 = (14,79,68)$, $q_t^5 = (61,45,38)$, $q_t^6 = (61,45,68)$, $q_t^7 = (61,79,38)$, $q_t^8 = (61,79,68)$.

Letting $\bar{\Omega} = \{1,2,3,4\}$, we have $Z^1 = 67.8$, $Z^2 = 71.6$, $Z^3 = 72.1$, and $Z^4 = 75.8$. Then the lowerbound inequality (21) is:

$$\sum_{\omega\in\Omega} p^{\omega} \sum_{t\in\mathcal{T}} \left( \sum_{i\in\mathcal{I}} c_{it}x_{it}^{\omega} + d_t z_t^{\omega} + q_t^{\omega} y_t^{\omega} \right) \geq 287.2. \tag{28}$$

and for $\bar{\Omega}$, the upperbound inequality (22) is:

$$\sum_{\omega\in\Omega} p^{\omega} \sum_{t\in\mathcal{T}} \left( \sum_{i\in\mathcal{I}} c_{it}x_{it}^{\omega} + d_t z_t^{\omega} + q_t^{\omega} y_t^{\omega} \right) \leq 592.9. \tag{29}$$

For $\bar{\Omega} = \{1,2,3,4,5,6,7,8\}$, the lower bound of the inequality (28) can be improved to 574.5, while the upper bound in the inequality (29) does not change.

Consider scenario $\xi^3$. The set of scenarios that are dominated by scenario $\xi^3$ is defined as $\beta^+_{\xi^3} = \{1,3\}$, and the set of scenarios that dominate scenario $\xi^3$ is defined as $\beta^-_{\xi^3} = \{4,7,8\}$. We have $Z^3 = 72.06$ by solving the scenario-$\xi^3$ problem. Thus, for each $k \in \beta^-_{\xi^3}$, the inequality (24) can be written as:

$$p^k \sum_{t\in\mathcal{T}} \left( \sum_{i\in\mathcal{I}} c_{it}x_{it}^{k} + d_t z_t^{k} + q_t^{l} y_t^{k} \right) \geq 72.06.$$

The optimal objective function value of S-MKP (20) for this instance is 574.51, and the linear programming relaxation value is 517.35. The inequalities (21), (22), and (24) for $\bar{\Omega} = \{1,2,3,4\}$ cut off the optimal relaxation solution, and the relaxation objective of (20) increases to 563.37, thus the initial optimality gap is closed by 80.5%.

Define $Z_t^{\omega}$ as the objective function value for scenario-$\xi^{\omega}$ problem at stage $t$. Solving the scenario-$\xi^3$ problem, we have $Z_2^3 = 16.5$, $Z_3^3 = 31.7$, and $Z_4^3 = 23.9$. For $k = 4 \in \beta^-_{\xi^3}$, the inequalities corresponding to (25) for $t = 3$ can be written as:

$$12.25x_{13}^4 + 11.5x_{23}^4 + 9.875z_3^4 + 10.75y_3^4 \geq 31.7.$$

On the other hand, the set of scenarios that are dominated by $\xi^3$ is defined as $\beta^+_{\xi^3} = \{1,3\}$, i.e., $\xi^1 \preceq \xi^3$ and $\xi^3 \preceq \xi^3$. For $k = 1 \in \beta^+_{\xi^3}$ and $t = 3$, the inequalities corresponding to (26) and (27) can be written respectively as:

$$14y_2^1 + 45y_3^1 \geq 83, \tag{30}$$

$$z_2^1 + z_3^1 \geq 1.775. \tag{31}$$

## 5. Computational experiments

In this section, we present results from our computational experiments showing the effectiveness of cuts and bounds based on the scenario dominance concept in solving multi-stage stochastic MIP instances. In particular, we solve a variety of S-MCLSP and S-MKP instances with different problem characteristics with the following approaches:

- **cpx**: CPLEX performance on solving the model by its default settings.
- **sdc+bc**: Lower (5) and upper bound (7) inequalities and sdc inequalities (11) generated for $\bar{\Omega} \subseteq \Omega$. See Algorithm 1 for the procedure to add inequalities (11).
- **ssdc**: Inequalities (16) and (17) for the S-MCLSP problem and inequalities (25)–(27) for the S-MKP problem. To add ssdc, the **Add Scenario Dominance Cut** procedure in Algorithm 1 is modified by the respective inequalities.
- **(l,S)**: The $(\ell, S)$ inequalities proposed by Barany et al. (1984). For all $i = 1,\ldots,M$, $\ell = 1,\ldots,T$, and $\omega = 1,\ldots,S$ the $(\ell, S)$ inequalities are defined for the S-MCLSP formulation as:

$$\sum_{t\in S} x_{it}^{\omega} + \sum_{t\in\bar{S}} d_{it}^{\omega,\ell} y_{it}^{\omega} \geq d_{i1}^{\omega,\ell}, \tag{32}$$

  where $S \subseteq \{1,\ldots,\ell\}$, $\bar{S} \subseteq \{1,\ldots,\ell\} \setminus S$ and $d_{it}^{\omega,\ell} = \sum_{k=t}^{\ell} d_{ik}^{\omega}$. The $(\ell, S)$ inequalities are valid for the feasible region of the S-MCLSP formulation.

The implementation specifications of all inequalities stated above and the details of the separation for adding the $(\ell, S)$ inequalities are described in Section 5.1.2 *Implementation Specifications*.

### 5.1. Test problems and implementation details

#### 5.1.1. Instance generation and test problems

For both lot-sizing and knapsack problem classes, various instances are randomly generated using different seeds from a uniform distribution. The parameter $T$ defines the number of stages in the stochastic scenario tree with two branches emanating from each node, resulting in a scenario tree with a size of $2^{T-1}$ scenarios in all of the instances.

**Stochastic Lot-Sizing Instances.** Test instances are generated similar to the scheme employed in Büyüktahtakın and Liu (2016), Büyüktahtakın et al. (2018b), and Yilmaz and Büyüktahtakın (2022). Various combinations of the number of stages $T \in \{5,6,7,8\}$, and the number of items $I \in \{2,3,4,5,8,9,10,60,70,80\}$ are considered.

The uncertain demand parameter $d_{it}^{\omega}$ has two levels (e.g., two branches in each scenario-tree node, Fig. 1): low (L) and high (H), where $d_{it}^{\omega} \in U[0,\bar{d}]$ for the low level and $d_{it}^{\omega} \in U\left[\bar{d}+1, 2\bar{d}\right]$ for the high level for each item $i \in \mathcal{I}$, and the average demand, $\bar{d}$, is set to 5. The capacity parameter in each period $t$ is generated by $c_t \in U\left[0.9\delta(\bar{d}+1), 1.1\delta(\bar{d}+1)\right]$, where $\delta = 2I$. The inventory holding cost $h_{it}$ is fixed at 1 for each period. The base setup-to-holding cost was set as $f_{it} \in U\left[0.9\theta\bar{h}, 1.1\theta\bar{h}\right]$, where $\theta = 1000$ and the average inventory holding cost, $\bar{h}$, is 1. Unit production costs $g_{it}$ are randomly generated from integer uniform distribution $g_{it} \in U[20,50]$. We generated ten random instances for various $(T, I)$ combinations with uniformly generated cost and capacity parameters, resulting in a total number of 120 S-MCLSP instances.

**Stochastic Knapsack Instances.** Test instances for S-MKP are generated in a similar fashion to the multi-stage stochastic knapsack instances of Büyüktahtakın (2022, 2023). The parameters $c_{it}$, $d_t$, $a_{ij}$, $r_t$, $b_t$, $w_t$, and $h_t$ are independent and identically distributed (i.i.d.) sampled from the uniform distribution over $\{1,\ldots,100\}$,e.g., $U[1,R]$, where $R = 100$, and $v_{it} \in U[1,R]$. The uncertain parameter $q_t^{\omega}$ for the 2-branch problem has two levels: low (L) and high (H), where $q_t^{\omega} \in U[1,R/2]$ for a low level and $q_t^{\omega} \in U[R/2+1,R]$ for a high level.

**Table 1**
Optimality gap due to $(\ell, S)$ and sdc+bc for S-MCLSP instances over $T$ = 5–8.

| | cpx | $(\ell, S)$ | | sdc+bc | | |
|---|---|---|---|---|---|---|
| (T, I) | InitGap (%) | Cut | GapImp (%) | Cut | RootGap (%) | GapImp (%) |
| (5, 60) | 15.8 | 919 | 12.5 | 4943 | 13.3 | 16.0 |
| (5, 70) | 16.4 | 1124 | 12.4 | 5753 | 13.7 | 15.9 |
| (5, 80) | 15.9 | 1144 | 12.9 | 6563 | 13.4 | 15.8 |
| (6, 8) | 18.1 | 368 | 12.8 | 2189 | 16.7 | 7.6 |
| (6, 9) | 17.6 | 389 | 12.9 | 2432 | 16.3 | 7.2 |
| (6, 10) | 18.7 | 509 | 12.7 | 2675 | 17.2 | 7.7 |
| (7, 3) | 17.9 | 382 | 11.6 | 2918 | 17.1 | 4.2 |
| (7, 4) | 19.9 | 481 | 12.7 | 3647 | 19.1 | 4.1 |
| (7, 5) | 19.5 | 654 | 11.8 | 4376 | 18.8 | 3.7 |
| (8, 2) | 20.8 | 656 | 11.7 | 6563 | 19.7 | 5.2 |
| (8, 3) | 21.8 | 1015 | 12.0 | 8750 | 20.9 | 3.8 |
| (8, 4) | 20.1 | 1176 | 11.6 | 10937 | 19.6 | 2.4 |
| Overall | 18.5 | 735 | 12.3 | 5146 | 17.2 | 7.8 |

Additionally, we define $b_t = \frac{3}{4}\left(\sum_{j=1}^{t}\sum_{i\in\mathcal{I}} a_{ij} + r_t\right)$ and $h_t = \frac{3}{4}\left(\sum_{i\in\mathcal{I}} v_{it} + w_t\right)$. The term $I$ is set at 50. By employing this scheme, we generate ten random feasible instances for each stage $T \in \{5,6,7,8,9,10\}$, resulting in a total number of 60 S-MKP instances.

### *5.1.2. Implementation specifications*

We append the cuts defined above as user cuts to the S-MCLSP and S-MKP formulations and then solve the resulting model using CPLEX 12.7.1 with its default settings. We evaluate the effectiveness of the proposed cuts in terms of the reduction in solution time, the integrality gap, as well as the number of nodes solved in the branch and bound tree relative to **cpx**. The best feasible solution found within 50 s is incorporated into the optimizer before running the model with proposed cuts to benefit from useful CPLEX cuts. As we focus on large-scale S-MIP problems, we set $p^{\omega} = 1/S$ as the probability of each scenario $\omega$ for both knapsack and lot-sizing instances generated.

Due to a large number of scenario dominance relations and resulting cuts, we perform some initial experiments to determine the subset of scenarios (see $\bar{\Omega}$ in Algorithm 1) to be solved and used for cut generation purposes. Per our preliminary computations, only five scenarios, $\bar{\Omega} = \{1, S/4, S/2, 3S/4, S\}$, where $S$ is the total number of scenarios, are used for cut generation for solving the S-MCLSP instances, while only the first 10% of the total scenarios are used for cut generation for solving the S-MKP. We note that more scenarios and scenario-dominance relations could also be used, but this comes with the computational cost of solving more scenario sub-problems to define a large set of dominance cuts. Our preliminary results show that the selected number of scenarios was sufficient to generate plenty of scenario dominance relations and cuts, and using them, we observe significant improvement in solution times. Thus, we consistently use the same number of scenarios in all of the computations.

**Number of Scenario Dominance Relations.** The stochastic lot-sizing instances have uncertainty in the right-hand side parameter $d_{it}^{\omega}$, which is $I \times T$ dimensional. Here, uncertain demand parameters are randomly generated over time; however, demand realizations of all items in a period $t$ follow an all-low ($d_{it}^{\omega} \in U[0, \bar{d}]$) or all-high distribution ($d_{it}^{\omega} \in U\left[\bar{d}+1, 2\bar{d}\right]$). For such instances with $T = 5$ and 16 scenarios, there are 146 dominance relations. For the stochastic lot-sizing instances with $T = 8$ and 128 scenarios, there are 4246 dominance relations on average.

For the stochastic knapsack instances, uncertainty is observed in the objective function parameter $q_t^{\omega}$, which is a $T$ dimensional random vector for a given $\omega \in \Omega$. Here, the values of all random variables are randomly generated using the uniform distribution described above. For $T = 5$, we have $2^4 = 16$ scenarios, and there are 146 dominance relations. For $T = 10$ instances with 512 scenarios, there are 38,854 dominance relations on average. We would like to note that independent from the studied instances, each scenario dominates only itself in the worst case, resulting in 512 dominance relations and cuts. In the deterministic version of the knapsack problem with zero random variables, we have 262,144 dominance relations, while for the stochastic knapsack instances with 81 random variables, we have only 512 dominance relations. Obtaining the scenario dominance relations with the sets only takes a few seconds for our instances. Those relations are put as input data for solving the problem.

**Separation Heuristic for ssdc.** In order to improve the quality of the solution obtained by ssdc for solving S-MCLSP, we employed a separation procedure for the cuts (16) and (17). Given an optimal solution $(\ddot{y}_{it}, \ddot{x}_{it}, \ddot{s}_{it})$ to the LP relaxation of the formulation (12), if $\sum_{h=1}^{t}\ddot{y}_{ih}^{k} \geq \sum_{h=1}^{t} y_{ih}^{l}$ for each $t \in \mathcal{T}$, where $y_{ih}^{l}$ is the optimal solution for the scenario-$\xi^{l}$ problem, then we add the inequality (16). Similarly, we add the inequality (17), if $\ddot{x}_{it}^{l} \leq x_{it}^{k}$ for each $t \in \mathcal{T}$, where $x_{ih}^{k}$ is the optimal solution for the scenario-$\xi^{k}$ problem. Here, different than the standard separation procedures (see, e.g., Atamtürk and Muñoz, 2004), we do not remove the LP relaxation solution but we investigate the inequality that potentially does not cutoff the optimal solution.

Computational experiments were conducted using the New Jersey Institute of Technology Kong Cluster. All computations and testing were performed on an 8-core Intel workstation with a 3.6 GHz CPU and 64 GB memory running Scientific Linux 6.4., using CPLEX 12.7.1. For all test instances, a time limit of 7200 CPU seconds was imposed. We used all eight threads for S-MCLSP instances. Due to the memory problems of the big size of S-MKP instances, we set the computations for S-MKP to a single thread only.

We describe our test instance generation scheme for both multi-item lot-sizing and stochastic multi-dimensional knapsack problems in Section 5.1.1. The detailed results of our computational experiments are presented in Sections 5.2–5.5.

### *5.2. Result specifications*

We define the following columns to present computational results in tables:

- **T**: Number of stages (periods).
- **Sce**: Number of scenarios.
- **I**: Number of items.
- **Exp**: Solution approach.
- **Cut**: Number of inequalities added to CPLEX as user inequalities.
- **Ctime**: CPU seconds required to generate the cuts, including the solution time of all scenario sub-problems.
- **Time**: Total CPU seconds required to solve the problem, including inequality generation time.
- **Tfac**: Time factor improvement by cuts over cpx (Tfac= $\text{Time}^1/\text{Time}^2$), where $\text{Time}^1$ is the **Time** by cpx and $\text{Time}^2$ is the **Time** by ssdc.
- **Node**: Number of nodes explored in the branch and bound tree in 10,000 s.
- **Obj**: Best objective value.
- **Gap**$^1$: Final optimality gap.
- **Gap**$^2$: Percentage deviation between the best objective value found by cpx ($\text{Obj}^1$) and the best objective value obtained by ssdc ($\text{Obj}^2$) [$\text{Gap}^2 = 100 \times (\text{Obj}^2/\text{Obj}^1\text{-}1)$].
- **InitGap**: Percentage integrality gap of the formulation before inequalities are added (InitGap = 100 × (bestobj — relaxlb)/bestobj), where relaxlb and bestobj are objective function values of the initial LP relaxation and the best feasible solution by cpx, respectively.
- **RootGap**: Percentage integrality gap of the formulation after inequalities are added (RootGap = 100 × (bestobj — rootlb)/ bestobj), where rootlb is the objective function value of the initial LP relaxation after cuts are added.
- **GapImp**: Percentage improvement in the integrality gap at the root node (GapImp = 100 × (1-relaxlb/rootlb).

**Table 2**
Experiments for S-MCLSP instances over 5–8 stages.

| (T, I) | Exp | Cut | Ctime | Time | Tfac | Node (in 10,000s) | Obj | Gap[1] (%) | Gap[2] (%) |
|---|---|---|---|---|---|---|---|---|---|
| (5, 60) | **cpx** | 0 | 0 | 422 | | 44 | 180254 | 0.000 | |
| | **ssdc** | 4943 | 59 | 77 | 6 | 4 | 180257 | | 0.002 |
| (5, 70) | **cpx** | 0 | 0 | 1273 | | 129 | 210531 | 0.000 | |
| | **ssdc** | 5753 | 61 | 70 | 18 | 3 | 210540 | | 0.004 |
| (5, 80) | **cpx** | 0 | 0 | 2144 | | 176 | 239824 | 0.000 | |
| | **ssdc** | 6563 | 69 | 81 | 27 | 2 | 239835 | | 0.005 |
| Ave | **cpx** | 0 | 0 | 1280 | | 116 | 210203 | 0.000 | |
| | **ssdc** | 5753 | 63 | 76 | 17 | 3 | 210211 | | 0.004 |
| (6, 8) | **cpx** | 0 | 0 | 3212 | | 497 | 27580 | 0.001 | |
| | **ssdc** | 2189 | 44 | 47 | 68 | 6 | 27587 | | 0.024 |
| (6, 9) | **cpx** | 0 | 0 | 2930 | | 461 | 30897 | 0.001 | |
| | **ssdc** | 2432 | 45 | 50 | 58 | 7 | 30901 | | 0.000 |
| (6, 10) | **cpx** | 0 | 0 | 3244 | | 478 | 35836 | 0.001 | |
| | **ssdc** | 2675 | 48 | 51 | 63 | 7 | 35848 | | 0.000 |
| Ave | **cpx** | 0 | 0 | 3129 | | 478 | 31438 | 0.001 | |
| | **ssdc** | 2432 | 46 | 50 | 63 | 7 | 31445 | | 0.008 |
| (7, 3) | **cpx** | 0 | 0 | 1419 | | 163 | 13588 | 0.001 | |
| | **ssdc** | 2918 | 39 | 42 | 34 | 7 | 13589 | | 0.011 |
| (7, 4) | **cpx** | 0 | 0 | 3088 | | 262 | 15849 | 0.002 | |
| | **ssdc** | 3647 | 43 | 49 | 63 | 6 | 15853 | | 0.028 |
| (7, 5) | **cpx** | 0 | 0 | 7209 | | 714 | 19992 | 0.008 | |
| | **ssdc** | 4376 | 52 | 89 | 81 | 10 | 20009 | | 0.086 |
| Ave | **cpx** | 0 | 0 | 3905 | | 380 | 16476 | 0.004 | |
| | **ssdc** | 3647 | 45 | 60 | 65 | 8 | 16484 | | 0.042 |
| (8, 2) | **cpx** | 0 | 0 | 454 | | 53 | 8842 | 0.000 | |
| | **ssdc** | 6563 | 39 | 45 | 10 | 6 | 8842 | | 0.002 |
| (8, 3) | **cpx** | 0 | 0 | 5823 | | 421 | 13296 | 0.009 | |
| | **ssdc** | 8750 | 53 | 483 | 12 | 52 | 13314 | | 0.136 |
| (8, 4) | **cpx** | 0 | 0 | 7214 | | 488 | 17460 | 0.013 | |
| | **ssdc** | 10937 | 53 | 2374 | 3 | 137 | 17515 | | 0.318 |
| Ave | **cpx** | 0 | 0 | 4497 | | 321 | 13199 | 0.007 | |
| | **ssdc** | 8750 | 49 | 967 | 5 | 65 | 13224 | | 0.152 |
| Overall | **cpx** | 0 | 0 | 3203 | | 324 | 67829 | 0.003 | |
| Ave | **ssdc** | 5146 | 50 | 288 | 37 | 21 | 67841 | | 0.051 |

**Table 3**
Optimality gap results due to sdc+bc cuts for S-MKP instances over 5–10 stages.

| | cpx | sdc + bc | | |
|---|---|---|---|---|
| (T, Sce) | InitGap (%) | Cut | RootGap (%) | GapImp (%) |
| (5, 16) | 0.76 | 17 | 0.55 | 27.8 |
| (6, 32) | 0.70 | 63 | 0.40 | 43.7 |
| (7, 64) | 0.71 | 188 | 0.34 | 52.7 |
| (8, 128) | 0.62 | 566 | 0.20 | 67.7 |
| (9, 256) | 0.75 | 1769 | 0.34 | 55.2 |
| (10, 512) | 0.72 | 61542 | 0.24 | 66.6 |
| Ave | 0.71 | 10691 | 0.34 | 52.3 |

Tables 1 and 3 only include columns **InitGap**, **Cut**, **RootGap**, and **GapImp**, while Tables 2 and 4 include all columns defined above except **InitGap**, **RootGap**, and **GapImp**. Table 5 includes a subset of those columns with two additional columns, **RV** and **ScenDom**, as defined in Section 5.5.

### *5.3. Results for stochastic lot-sizing instances*

In Table 1, the optimality gap improvement using the sdc+bc cuts is presented and compared with that of $(\ell, S)$. The percent gap improvement due to sdc+bc cuts is inversely proportional to stage $T$. When all 120 instances are averaged, sdc+bc cuts reduce the optimality gap from 18.5% to 17.2%, providing an average gap improvement benefit of 7.8%. On the other hand, $(\ell, S)$ provides an average optimality gap improvement of 12.3%.

Based on the computational results, we conclude that the main benefit of sdc, bc and $(\ell, S)$ is to close the initial gap defined by the linear programming relaxation rather than speeding up the solution time. Thus, in the next computational experiment, we omit sdc, bc and $(\ell, S)$ and focus on only the ssdc cuts to obtain computational speed-up in solving the augmented problem.

We present results regarding the efficiency of ssdc cuts on S-MCLSP instances with a variety of combinations of stages and items, as shown in Table 2. Those instances represent hard-to-solve S-MCLSP formulations with the number of variables ranging from 3456 to 15,360 and the number of constraints ranging from 3648 and 16,640. The difficulty of these instances increases with the time stage $T$ and the number of items $I$.

The ssdc cuts drastically reduce the solution time in all problems, as presented in Table 2. For instance, the solution time of 3129 CPU sec. for $T = 6$ instances is reduced to only 50 CPU sec., including the cut generation time of 46 CPU sec. The average time-factor improvement for $T = 6$ and $T = 7$ instances is 63 and 65, while it is 17 and 3 for $T = 5$ and $T = 8$ instances, respectively. Here, we also tested the $(\ell, S)$ inequalities proposed by Barany et al. (1984). Since $(\ell, S)$ inequalities did not provide any solution time improvement over cpx, we did not present specific results regarding the $(\ell, S)$ inequalities in Table 2. However, the ssdc cuts provide a computational improvement of a factor of at least 37 over the $(\ell, S)$ inequalities of Barany et al. (1984), which have long been known as the state-of-the-art for solving the multi-item lot-sizing problems. As shown in the overall averages in Table 2, ssdc cuts improve results by a factor of 37, averaging **Tfac** over all 120 instances, while the resulting deviation from the best cpx solution is 0.051%. On average, ssdc cuts reduce the number of nodes in the branch and bound tree by 94% percent while improving solution times by 91% percent.

**Table 4**
Experiments for S-MKP instances over 5–10 stages and $I = 50$.

| (T, Sce) | Exp | Cut | Ctime | Time | Tfac | Node (in 10,000s) | Obj | Gap[1] (%) | Gap[2] (%) |
|---|---|---|---|---|---|---|---|---|---|
| (5, 16) | **cpx** | 0 | 0 | 2418 | | 215.1 | 5101 | 0.000 | |
| | **ssdc** | 85 | 18 | 19 | 127 | 2.0 | 5101 | | 0.000 |
| (6, 32) | **cpx** | 0 | 0 | 5776 | | 350.7 | 6594 | 0.068 | |
| | **ssdc** | 418 | 43 | 49 | 118 | 4.9 | 6594 | | 0.004 |
| (7, 64) | **cpx** | 0 | 0 | 5846 | | 183.1 | 7439 | 0.059 | |
| | **ssdc** | 1484 | 50 | 75 | 78 | 3.2 | 7440 | | 0.015 |
| (8, 128) | **cpx** | 0 | 0 | 5766 | | 78.6 | 8960 | 0.064 | |
| | **ssdc** | 5144 | 60 | 77 | 75 | 0.9 | 8961 | | 0.021 |
| (9, 256) | **cpx** | 0 | 0 | 6651 | | 55.5 | 10228 | 0.132 | |
| | **ssdc** | 18129 | 128 | 153 | 44 | 0.2 | 10243 | | 0.147 |
| (10, 512) | **cpx** | 0 | 0 | 7246 | | 23.9 | 11325 | 0.116 | |
| | **ssdc** | 61542 | 427 | 589 | 12 | 0.2 | 11344 | | 0.172 |
| Ave | **cpx** | 0 | 0 | 5617 | | 151 | 8274 | 0.073 | |
| | **ssdc** | 14467 | 121 | 160 | 76 | 2 | 8281 | | 0.060 |

### *5.4. Results for stochastic knapsack instances*

In Table 3, we analyze the optimality gap improvement using the sdc+bc cuts. As the number of stages increases, the percent gap improvement due to sdc+bc cuts shows an increasing trend. In the overall averages in Table 3, sdc+bc cuts reduce the optimality gap from 0.71% to 0.34%, providing an average gap improvement benefit of 52.3%. While sdc+bc provides a large optimality gap improvement, they do not speed up the solution time. Thus, in the next set of computational experiments, we omit sdc and bc and focus on only the ssdc cuts. Table 4 summarizes results for S-MKP instances with stages from $T = 5$ to $T = 10$ where $I = 50$. The instances in Table 4 are hard, large-scale dynamic knapsack instances with the number of variables ranging from 3328 to 239,616 and the number of constraints changing from 2624 to 222,208. As expected, the difficulty of instances increases as the total time stage $T$ increases. Similar to the results with S-MCLSP experiments, for all instances, we observe that the ssdc improves the solution time drastically. Similarly, we observe a significant reduction in the number of nodes explored in the branch and bound tree with the addition of the ssdc cuts.

For instances with stage $T = 5$, ssdc provides the same objective value with cpx by reducing the cpx solution time from 2418 to 19 CPU seconds. For instances with stage $T = 6$, the cpx solution time is reduced by a factor of 118 using ssdc, while the ssdc objective value deviates from cpx best objective value by only 0.004%. In particular, for stage $T = 6$ instances, ssdc cuts reduce the solution time from 5776 CPU sec. to 49 CPU sec., including the cut generation time of 43 CPU sec. The number of cuts generated and thus the time for cut generation increases with stage because the number of scenario-$\omega$ problems to be solved for cut derivation increases with stage.

As shown in the overall averages in Table 4 ssdc cuts improve results by a factor of 76, averaging **Tfac** over all 60 instances, while the resulting gap from the best cpx solution is only 0.06%, similar to the solution time and gap performance of ssdc with the S-MCLSP instances. On average, the cpx solution time is 5617 CPU sec., while the time consumed to generate the ssdc cuts is 121 CPU sec., and the ssdc solution time is 39 CPU sec., adding up to a total solution time of 160 CPU sec.

### *5.5. Results with more random variables*

In this section, we explore the effectiveness of **ssdc** as we increase the number of random variables by more than three per stage. To perform those experiments, we have generated additional S-MKP instances where we re-defined the parameter $c_{it}$ as an uncertain parameter $c_{it}^{\omega}$, while assuming all other parameters deterministic. The random parameters $c_{it}^{\omega}$ are generated as stage-wise independent but are jointly distributed over $i$ for each $t \in \mathcal{T}$. A scenario $\xi^k$ dominates a scenario $\xi^l$ with respect to the problem (20) if $\left(p^k \geq p^l\right)$ and $\left(c_{it}^k \geq c_{it}^l\right)$ hold for each $i \in \mathcal{I}$ and $j = 2, \ldots, T$.

**Table 5**
Results for S-MKP instances with a varying number of random variables RV, $T = 10$, $I = 10$, and 512 scenarios.

| Exp | RV | ScenDom | Cut | Ctime | Ttime | Tfac | Gap[1](%) | Gap[2](%) |
|---|---|---|---|---|---|---|---|---|
| **cpx** | 0 | 262,144 | | | 739 | | 0.01 | |
| **ssdc** | | | 129,024 | 182 | 205 | 4 | | 0.00 |
| **cpx** | 9 | 19,683 | | | 7,221 | | 0.09 | |
| **ssdc** | | | 249,170 | 225 | 286 | 25 | | 1.24 |
| **cpx** | 18 | 3,889 | | | 5,210 | | 0.07 | |
| **ssdc** | | | 256,612 | 220 | 249 | 21 | | −1.54 |
| **cpx** | 27 | 1,563 | | | 5,170 | | 0.09 | |
| **ssdc** | | | 257,226 | 222 | 260 | 20 | | 0.06 |
| **cpx** | 36 | 742 | | | 5,903 | | 0.12 | |
| **ssdc** | | | 257,692 | 226 | 258 | 23 | | −0.62 |
| **cpx** | 45 | 710 | | | 5,151 | | 0.08 | |
| **ssdc** | | | 257,648 | 221 | 254 | 20 | | −0.75 |
| **cpx** | 54 | 538 | | | 5,874 | | 0.17 | |
| **ssdc** | | | 257,771 | 222 | 260 | 23 | | −0.65 |
| **cpx** | 63 | 563 | | | 5,880 | | 0.16 | |
| **ssdc** | | | 257,785 | 220 | 259 | 23 | | 0.40 |
| **cpx** | 72 | 627 | | | 5,780 | | 0.13 | |
| **ssdc** | | | 257,720 | 228 | 261 | 22 | | 0.15 |
| **cpx** | 81 | 512 | | | 6,045 | | 0.10 | |
| **ssdc** | | | 257,796 | 226 | 287 | 21 | | 0.43 |
| Average | 41 | 29,097 | | | 5,297 | | 0.10 | |
| | | | 243,844 | 219 | 258 | 20 | | −0.13 |

Table 5 presents the performance of **ssdc** with respect to **cpx** for a varying number of random variables in the scenario tree for S-MKP instances with $T = 10$, $I = 10$, and 512 scenarios. Each row of Table 5 presents results for an average of ten instances. In Table 5, **RV** shows the total number of random variables generated, and **ScenDom** presents the total number of scenario-dominance relations over all time periods for each **RV**. For example, the instances with **RV**$= 0$ are deterministic, while the instances with **RV**$= 81$ represent the case where nine items ($I = 9$) are randomly generated in each stage, and thus has a total of $I(T-1) = 81$ random variables.

The results in Table 5 show that **ssdc** decreases the average solution times of **cpx** for instances with up to 81 random variables by an overall average factor of 20. Specifically, the average solution time of one and a half hours is reduced to less than five minutes by **ssdc** while also improving the average objective function value found by **cpx**. As expected, the increasing number of random variables reduces the number of scenario-dominance relations. For example, for the deterministic version of the problem (**RV**$= 0$), we have 262,144 dominance relations, while for the stochastic instances with **RV**$= 27$, we have only

1563 dominance relations. The results imply that the increase in the number of random variables does not impact the performance of **ssdc** with respect to **cpx** for the considered instances because the algorithm still generates plenty of cutting planes due to an ample number of dominance relations. However, for extremely-hard instances for which a feasible solution cannot be found in a reasonable solution time by **cpx**, the performance of **ssdc** is also expected to worsen. In these experiments, the number of cuts generated is large for each case since we use both dominating and dominated sets for cut generation. For example, for the instances with **RV**= 81, we have the minimum number of scenario dominance relations equal to 512, and 257,796 **ssdc** cuts are generated. The MIP gap tends to increase as the number of random variables increases because the instances with larger RV are harder. The average percent deviation by **ssdc** over **cpx** (**Gap**$^2$) is negative, implying that **ssdc** improves the objective function found by **cpx** in two hours by 0.13% in five minutes.

## 6. Concluding remarks

In this study, we have explored the scenario dominance concept to effectively solve the general multi-stage stochastic mixed-integer programs with a focus on the lot-sizing and knapsack problems. The scenario dominance method derives implications based on a pairwise comparison and partial ordering of scenarios. Specifically, inferences obtained by the solution of scenario sub-problems and the partial ordering of scenarios are used to drive new cutting planes and bounds to improve the computational solvability of multi-stage stochastic programs. We generate specific cuts and bounds for the stochastic dynamic knapsack and stochastic capacitated lot-sizing problems based on the scenario dominance approach. Our extensive computational experiments demonstrate that the proposed methodology with strong dominance cuts provides a significant improvement in solving large-scale, multi-stage stochastic optimization problems with integer and continuous variables.

The results highlight that the specific implementation of the proposed bounds and dominance cuts depends on the user's preferences. If the user prefers the quality of the solution over the solution time, implementing the lower and upper bounds with the scenario dominance cuts (sdc) is more preferred to using strong dominance cuts (ssdc). On the other hand, if the user or decision maker prefers a quick solution with the cost of a little deviation from the optimal solution, the implementation of the strong dominance cuts is suggested as they are more aggressive than the former ones in terms of chopping off the feasible solutions and finding a faster solution. For example, in an energy production setting, where a production planning problem needs to be solved a few times a day (Xavier et al., 2021; Cerisola et al., 2009), the decision maker may want to implement strong dominance cuts over the bounds or dominance cuts. In another setting where the optimal solution is critical to finding a long-term solution, e.g., health-care infrastructure decision-making (Büyüktahtakın et al., 2018a; Yin et al., 2023a; Bushaj et al., 2022b), the use of bounds and cuts may be preferable to improve the relaxation of the formulations. In sum, bounds and sdc are more helpful in improving the lower and upper bounds in a branch and bound solver rather than reducing time, while strong dominance cuts (ssdc) are suggested to be used whenever a fast and close-to-optimal solution is more desirable over the optimal one.

To improve the quality of the ssdc, we employ simple preprocessing heuristics, such as running CPLEX for 50 s to benefit from CPLEX cuts and the best solution found in that time limit, before running the model with the associated dominance cuts. Since ssdc may cutoff the optimal solution, we also perform a heuristic separation to find out cuts that would potentially preserve the optimal solution. Further research on preprocessing algorithms and heuristics is needed to improve the solution quality of those aggressive cuts.

Besides improving solution quality, this study opens up other exciting avenues for further research. For example, the proposed methods apply to various scenario tree configurations, and thus investigating the performance of the scenario dominance cuts for different scenario structures is a possible future research direction. The proposed scenario dominance approach requires an element-by-element comparison of the entries of stochastic vectors and matrices in the M-SMIP problems to obtain a partial ordering of scenarios. Our computational experiments have shown that one can derive a large number of dominance relations and scenario dominance cuts by considering uncertainty only in the most important parameters of the problem. Thus, we focused on S-MKP instances with 9 random variables (for $T = 10$ instances) and S-MCLSP instances with a maximum of 320 random variables (for $T = 5$ and $I = 80$ instances). We have also investigated the limitation of the proposed approach for the S-MKP instances by increasing the number of random variables from 9 to 81.

Our approach is general in the sense that even if the scenarios are quite distinct, the random variable outcomes in each scenario are always comparable based on the definition of a scenario in stochastic programming. However, if the random variables are entirely independent, where quite different distributions are used to generate each of them, the scenario outcomes are also dissimilar. In that case, one can expect fewer dominance relations compared to the case where scenarios are generated using similar distributions. For example, in a multi-item production planning schema, one can expect that the marginal distributions of the demand for each item follow a similar shape. However, it is also possible that the marginal distribution of every item has a unique shape (Kaut, 2011). However, this setting will explode the number of scenarios in the scenario tree since each item requires a specific branch for each outcome. A future direction of this work could investigate the number and effectiveness of scenario dominance cuts, where the scenarios are generated using distinct distributions of the random variables with larger scenario trees than considered in this paper.

In our computations, we have restricted the number of scenario sub-problems, and used only a small subset of the scenarios for a cut generation. Further experimental analysis is needed to study the impact of various scenario selection schemes on the quality of the bounds and cuts obtained based on the scenario dominance concept. Thus, one important avenue for further research is to provide criteria for deciding how many and which scenarios should be chosen for solving scenario sub-problems.

In this paper, we have focused on the type of stochastic multi-dimensional knapsack and stochastic multi-item lot-sizing instances, which have computationally challenging deterministic formulations when the uncertain parameters are set to their expected values. Future research could investigate M-SMIP instances, which have relatively easy deterministic counterparts but a larger number of scenarios and stages than considered in this work, and thus remain challenging. In such cases, a parallel distributed implementation for solving the individual scenario problems could reduce the cut generation time and thus improve the overall solution performance.

The proposed scenario dominance method could also be used within a time- or scenario-decomposition framework proposed for multi-stage stochastic mixed-integer programs. For example, a classical scenario decomposition algorithm could be adjusted by using the new scenario sub-problem as defined here and enhanced by using the scenario-dominance bounds and cuts developed in this paper. Furthermore, since neither linearity nor convexity is not assumed, our approach has the potential to be adapted to solve stochastic programming problems with non-linear and non-convex characteristics. Future research might, for example, explore the proposed method for the case of non-linearity in multi-stage stochastic programs with discrete variables. Future studies could also explore the dominance and strong dominance cuts for M-SMIPs, including various risk measures in addition to the expectation in the objective function. For example, suppose Excess Probability (EP), which measures the probability of exceeding a prescribed target level (Schultz and Tiedemann, 2003), is chosen as a risk measure. In that case, the additional EP-related term in the objective function must be incorporated into the scenario dominance cuts (e.g., inequalities (11) and (25)). However, the strong dominance cuts that include only a specific type of variable (e.g., inequalities (26) and (27)) can be used with slight modifications in a mean-EP M-SMIP. The applications for which our approach could be beneficial also vary widely, ranging from multi-stage stochastic capacity allocation to stochastic network design. Hence, the derivation of scenario dominance cuts for each specific S-MIP problem remains an important future direction.

## CRediT authorship contribution statement

**İ. Esra Büyüktahtakın:** Conceptualization, Methodology, Software, Data curation, Validation, Visualization, Experimentation, Investigation, Writing – original draft, Writing – reviewing and editing.

## Data availability

All S-MKP and S-MCLSP instances generated can be found at the Systems Optimization and Machine Learning Laboratory problem library (SysOptiMaL SMIPLIB) (Büyüktahtakın, 2023).

## Acknowledgments

The author gratefully acknowledges the support of the National Science Foundation CAREER Award, United States of America co-funded by the CBET/ENG Environmental Sustainability program and the Division of Mathematical Sciences in MPS/NSF under Grant #CBET-1554018. We would like to thank two anonymous referees, the area editor and the editor, for their helpful and constructive comments that have significantly improved our exposition. We also thank the Academic Research and Computing Systems (ARCS) center of New Jersey Institute of Technology (NJIT) for their support with the NJIT ARCS Kong cluster, which is used to perform our computational testing.

## References

Abgottspon, H., Njálsson, K., Bucher, M.A., Andersson, G., 2014. Risk-averse medium-term hydro optimization considering provision of spinning reserves. In: Probabilistic Methods Applied to Power Systems, PMAPS, 2014 International Conference on. IEEE, pp. 1–6.

Ahmed, S., 2013. A scenario decomposition algorithm for 0–1 stochastic programs. Oper. Res. Lett. 41 (6), 565–569.

Ahmed, S., Garcia, R., 2003. Dynamic capacity acquisition and assignment under uncertainty. Ann. Oper. Res. 124 (1–4), 267–283.

Alonso-Ayuso, A., Escudero, L.F., Guignard, M., Weintraub, A., 2018. Risk management for forestry planning under uncertainty in demand and prices. European J. Oper. Res. 267 (3), 1051–1074.

Alonso-Ayuso, A., Escudero, L.F., Ortuno, M.T., 2003. BFC, a branch-and-fix coordination algorithmic framework for solving some types of stochastic pure and mixed 0–1 programs. European J. Oper. Res. 151 (3), 503–519.

Arpón, S., Homem-de Mello, T., Pagnoncelli, B., 2018. Scenario reduction for stochastic programs with conditional value-at-risk. Math. Program. 170 (1), 327–356.

Atamtürk, A., Muñoz, J.C., 2004. A study of the lot-sizing polytope. Math. Program. 99 (3), 443–465.

Barany, I., Van Roy, T.J., Wolsey, L.A., 1984. Strong formulations for multi-item capacitated lot sizing. Manage. Sci. 30 (10), 1255–1261.

Benders, J.F., 2005. Partitioning procedures for solving mixed-variables programming problems. Comput. Manag. Sci. 2 (1), 3–19.

Bertsimas, D., Mundru, N., 2022. Optimization-based scenario reduction for data-driven two-stage stochastic optimization. Oper. Res..

Birge, J.R., Louveaux, F., 2011. Introduction to Stochastic Programming. Springer Science & Business Media.

Boyd, E.A., 1993. Polyhedral results for the precedence-constrained knapsack problem. Discrete Appl. Math. 41 (3), 185–201.

Bruno, S., Ahmed, S., Shapiro, A., Street, A., 2016. Risk neutral and risk averse approaches to multistage renewable investment planning under uncertainty. European J. Oper. Res. 250 (3), 979–989.

Bushaj, S., Büyüktahtakın, İ.E., Haight, R.G., 2022a. Risk-averse multi-stage stochastic optimization for surveillance and operations planning of a forest insect infestation. European J. Oper. Res. 299 (3), 1094–1110.

Bushaj, S., Büyüktahtakın, İ.E., Yemshanov, D., Haight, R.G., 2021. Optimizing surveillance and management of emerald ash borer in urban environments. Nat. Resour. Model. 34 (1), e12267.

Bushaj, S., Yin, X., Beqiri, A., Andrews, D., Büyüktahtakın, İ.E., 2022b. A simulation-deep reinforcement learning (SiRL) approach for epidemic control optimization. Ann. Oper. Res. 1–33.

Büyüktahtakın, İ.E., 2011. Dynamic programming via linear programming. In: Wiley Encyclopedia of Operations Research and Management Science. Wiley, Hoboken, NJ.

Büyüktahtakın, İ.E., 2022. Stage-t scenario dominance for risk-averse multi-stage stochastic mixed-integer programs. Ann. Oper. Res. 309 (1), 1–35.

Büyüktahtakın, İ.E., 2023. SysOptiMaL SMIPLIB: Systems optimization and machine learning laboratory multi-stage stochastic multi-dimensional mixed 0-1 knapsack (S-MKP) and multi-stage stochastic multi-item capacitated lot-sizing (S-MCLSP) test problem library. https://soml.ise.vt.edu/people.html.

Büyüktahtakın, İ.E., des Bordes, E., Kıbış, E.Y., 2018a. A new epidemics-logistics model: Insights into controlling the ebola virus disease in West Africa. European J. Oper. Res. 265 (3), 1046–1063.

Büyüktahtakın, İ.E., Liu, N., 2016. Dynamic programming approximation algorithms for the capacitated lot-sizing problem. J. Global Optim. 65 (2), 231–259.

Büyüktahtakın, İ.E., Smith, J.C., Hartman, J.C., 2018b. Partial objective inequalities for the multi-item capacitated lot-sizing problem. Comput. Oper. Res. 91, 132–144.

CarøE, C.C., Schultz, R., 1999. Dual decomposition in stochastic integer programming. Oper. Res. Lett. 24 (1–2), 37–45.

Cerisola, S., Baíllo, Á., Fernández-López, J.M., Ramos, A., Gollmer, R., 2009. Stochastic power generation unit commitment in electricity markets: A novel formulation and a comparison of solution methods. Oper. Res. 57 (1), 32–46.

Cerisola, S., Latorre, J.M., Ramos, A., 2012. Stochastic dual dynamic programming applied to nonconvex hydrothermal models. European J. Oper. Res. 218 (3), 687–697.

Chen, Z.-L., Li, S., Tirupati, D., 2002. A scenario-based stochastic programming approach for technology and capacity planning. Comput. Oper. Res. 29 (7), 781–806.

Cobuloglu, H.I., Büyüktahtakın, İ.E., 2017. A two-stage stochastic mixed-integer programming approach to the competition of biofuel and food production. Comput. Ind. Eng. 107, 251–263.

Dantzig, G.B., Infanger, G., 1993. Multi-stage stochastic linear programs for portfolio optimization. Ann. Oper. Res. 45 (1), 59–76.

Escudero, L.F., Garín, M.A., Unzueta, A., 2016. Cluster Lagrangean decomposition in multistage stochastic optimization. Comput. Oper. Res. 67, 48–62.

Gade, D., Hackebeil, G., Ryan, S.M., Watson, J.-P., Wets, R.J.-B., Woodruff, D.L., 2016. Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. Math. Program. 157 (1), 47–67.

Guan, Y., Ahmed, S., Nemhauser, G.L., 2009. Cutting planes for multistage stochastic integer programs. Oper. Res. 57 (2), 287–298.

Hartman, J.C., Büyüktahtakin, İ.E., Smith, J.C., 2010. Dynamic-programming-based inequalities for the capacitated lot-sizing problem. IIE Trans. 42 (12), 915–930.

Heitsch, H., Römisch, W., 2003. Scenario reduction algorithms in stochastic programming. Comput. Optim. Appl. 24 (2–3), 187–206.

Israeli, E., Wood, R.K., 2002. Shortest-path network interdiction. Networks 40 (2), 97–111.

Kaut, M., 2011. Scenario generation for stochastic programming introduction and selected methods. SINTEF Technol. Soc. 3, 1–61.

Kellerer, H., Pferschy, U., Pisinger, D., 2004. Multidimensional knapsack problems. In: Knapsack Problems. Springer, pp. 235–283.

Kıbış, E.Y., Büyüktahtakın, İ.E., 2019. Optimizing multi-modal cancer treatment under 3D spatio-temporal tumor growth. Math. Biosci. 307, 53–69.

Kıbış, E.Y., Büyüktahtakın, İ.E., Haight, R.G., Akhundov, N., Knight, K., Flower, C.E., 2021. A multistage stochastic programming approach to the optimal surveillance and control of the emerald ash borer in cities. INFORMS J. Comput. 33 (2), 808–834.

Löhndorf, N., Wozabal, D., Minner, S., 2013. Optimizing trading decisions for hydro storage systems using approximate dual dynamic programming. Oper. Res. 61 (4), 810–823.

Lulli, G., Sen, S., 2004. A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems. Manage. Sci. 50 (6), 786–796.

Madansky, A., 1960. Inequalities for stochastic linear programming problems. Manage. Sci. 6 (2), 197–204.

Marchand, H., Wolsey, L.A., 1999. The 0-1 knapsack problem with a single continuous variable. Math. Program. 85 (1), 15–33.

Müller, A., Stoyan, D., 2002. Comparison Methods for Stochastic Models and Risks, Vol. 389. Wiley New York.

Mulvey, J.M., Vladimirou, H., 1992. Stochastic network programming for financial planning problems. Manage. Sci. 38 (11), 1642–1664.

Park, K., Park, S., 1997. Lifting cover inequalities for the precedence-constrained knapsack problem. Discrete Appl. Math. 72 (3), 219–241.

Rahimian, H., Bayraksan, G., Homem-de Mello, T., 2019. Identifying effective scenarios in distributionally robust stochastic programs with total variation distance. Math. Program. 173 (1–2), 393–430.

Römisch, W., Schultz, R., 2001. Multistage stochastic integer programs: An introduction. In: Online Optimization of Large Scale Systems. Springer, pp. 581–600.

Ruszczyński, A., 2002. Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra. Math. Program. 93 (2), 195–215.

Schultz, R., Tiedemann, S., 2003. Risk aversion via excess probabilities in stochastic programs with mixed-integer recourse. SIAM J. Optim. 14 (1), 115–138.

Sen, S., 2005. Algorithms for stochastic mixed-integer programming models. Handbooks Oper. Res. Management Sci. 12, 515–558.

Shapiro, A., Dentcheva, D., Ruszczyński, A., 2009. Lectures on Stochastic Programming: Modeling and Theory. SIAM.

Shapiro, A., Tekaya, W., da Costa, J.P., Soares, M.P., 2013. Risk neutral and risk averse stochastic dual dynamic programming method. European J. Oper. Res. 224 (2), 375–391.

Thome, F., Pereira, M., Granville, S., Fampa, M.H.C.F., 2013. Non-convexities representation on hydrothermal operation planning using SDDP. URL: www.psr-inc.com, submitted.

Watson, J.-P., Woodruff, D.L., 2011. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. Comput. Manag. Sci. 8 (4), 355–370.

Xavier, Á.S., Qiu, F., Ahmed, S., 2021. Learning to solve large-scale security-constrained unit commitment problems. INFORMS J. Comput. 33 (2), 739–756.

Yilmaz, D., Büyüktahtakın, İ.E., 2022. An expandable learning-optimization framework for sequentially dependent decision-making. Eur. J. Oper. Res. (under review for publication).

Yin, X., Bushaj, S., Yuan, Y., Büyüktahtakın, İ.E., 2023a. COVID-19: Agent-based simulation-optimization to vaccine center location vaccine allocation problem. IISE Trans. (under review for publication).

Yin, X., Büyüktahtakın, İ.E., 2021. A multi-stage stochastic programming approach to epidemic resource allocation with equity considerations. Health Care Manage. Sci. 24 (3), 597–622.

Yin, X., Büyüktahtakın, İ.E., 2022. Risk-averse multi-stage stochastic programming to optimizing vaccine allocation and treatment logistics for effective epidemic response. IISE Trans. Healthc. Syst. Eng. 12 (1), 52–74.

Yin, X., Büyüktahtakın, İ.E., Patel, B.P., 2023b. COVID-19: Data-driven optimal allocation of ventilator supply under uncertainty and risk. European J. Oper. Res. 304, 255–275.

Zou, J., Ahmed, S., Sun, X.A., 2019. Stochastic dual dynamic integer programming. Math. Program. 175 (1–2), 461–502.