Cutting planes for multi-stage stochastic integer programs*

Yongpei Guan[‡], Shabbir Ahmed[†] and George L. Nemhauser[†]
[‡]School of Industrial Engineering, University of Oklahoma, Norman, OK 73019
[†]School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332

September 15, 2006

Abstract

This paper addresses the problem of finding cutting planes for multi-stage stochastic integer programs. We give a general method for generating cutting planes for multi-stage stochastic integer programs based on combining inequalities that are valid for the individual scenarios. We apply the method to generate cuts for a stochastic version of a dynamic knapsack problem and to stochastic lot sizing problems. We give computational results which show that these new inequalities are very effective in a branch-and-cut algorithm.

1 Introduction

This paper deals with polyhedral aspects of multi-stage stochastic integer programs. Our basic idea is to extend known results concerning cutting planes for a deterministic model of the problem to a stochastic model. In other words, suppose we know valid inequalities that make it possible to solve efficiently the deterministic model by linear programming or branch-and-cut. In this paper we show how to use this knowledge to get valid inequalities for a stochastic scenario-tree based model of the problem so that it too can be solved by a branch-and-cut algorithm. Multi-period production planning problems are a typical example where there is considerable knowledge of the convex hull of feasible solutions for various deterministic problems. In [9] we showed how to apply this idea for uncapacitated lot-sizing problems by generalizing the well-known (ℓ, S) inequalities [2] to a stochastic setting. Here we generalize the basic ideas of [9] so that the results can be applied to general multi-stage stochastic integer programs involving a scenario tree model of the uncertain parameters. The key idea of our approach is to combine deterministic valid inequalities corresponding to different scenarios to obtain valid inequalities for the whole scenario tree. The general framework is studied in detail in the context of stochastic dynamic knapsack problems and stochastic lot-sizing problems. For these special cases, we provide facet and convex hull defining conditions, and discuss separation procedures. We also present computational results which show that the approach is computationally feasible for stochastic lot-sizing problems.

The remainder of the paper is organized as follows. In the next section we present notation and terminology used throughout the paper, and also discuss the underlying combination principle in our approach. A general framework for obtaining valid inequalities for stochastic scenario-tree integer programs is given in Section 3. Applications to stochastic dynamic knapsack problems and stochastic lot-sizing problems are presented in Sections 4 and 5, respectively. Section 6 presents computational results, and Section 7 gives conclusions and directions for future research.

*This research has been supported in part by the National Science Foundation under grants DMI-0121495 and DMI-0522485.

2 Notation and Preliminaries

2.1 Multi-stage stochastic integer programs

Consider the deterministic T-period mixed integer program

$$\begin{aligned} \min \quad & \sum_{t=1}^{T}(\alpha_t x_t + \beta_t y_t) \\ \text{s.t.} \quad & \sum_{\tau=1}^{t}(G_{t\tau}x_\tau + A_{t\tau}y_\tau) \geq b_t \quad t = 1,\ldots,T \\ & x_t \in \mathbb{R}^p_+, \quad y_t \in \mathbb{Z}^n_+ \qquad\quad t = 1,\ldots,T. \end{aligned} \tag{1}$$

In (1), $A_{t\tau}$ and $G_{t\tau}$ are matrices, and α_t, β_t and b_t are vectors, respectively, of appropriate dimensions. We assume, without loss of generality, that the decision vectors in each of the time period $t = 1,\ldots,T$ are of identical dimension.

Now consider the extension of (1) to a stochastic setting. We assume that the problem parameters (α, β, G, A, b) evolve as a discrete time stochastic process with a finite probability space. This information structure can be interpreted as a scenario tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ with T levels (or stages) where a node $i \in \mathcal{V}$ in stage t of the tree gives the state of the system that can be distinguished by information available up to time stage t. The probability associated with the state represented by node i is p_i. The set of nodes on the path from the root node (indexed as $i = 0$) to a node i is denoted by $\mathcal{P}(i)$. The decisions (x_i, y_i) corresponding to a node i are assumed to be made after observing the realizations $(\alpha_i, \beta_i, \{G_{ij}\}_{j\in\mathcal{P}(i)}, \{A_{ij}\}_{j\in\mathcal{P}(i)}, b_i)$ but are non-anticipative with respect to future realizations. The goal is to minimize *expected* total costs. The multi-stage stochastic integer programming extension of (1) is then

$$\begin{aligned} \min \quad & \sum_{i\in\mathcal{V}} p_i(\alpha_i x_i + \beta_i y_i) \\ \text{s.t.} \quad & \sum_{j\in\mathcal{P}(i)} (G_{ij}x_j + A_{ij}y_j) \geq b_i \quad i \in \mathcal{V} \\ & x_i \in \mathbb{R}^p_+, \quad y_i \in \mathbb{Z}^n_+ \qquad\quad i \in \mathcal{V}. \end{aligned} \tag{2}$$

Formulation (2) is completely general. Any multi-stage stochastic integer program defined over a scenario tree can be modelled according to this formulation (cf. [16, 17]). Specific examples of such problems include stochastic lot-sizing problems [9, 12] (see also Section 5), stochastic capacity planning models [1, 18], and the stochastic unit commitment problem [15, 19].

2.2 Path and Tree sets

We denote the set of feasible solutions of the multi-stage stochastic integer program (2) by $X_\mathcal{T}$, and refer to this set as the *tree set*. In this paper, we develop valid inequalities for the tree set $X_\mathcal{T}$ by combining given valid inequalities for *path sets* of the form

$$X_i = \Big\{(x_j, y_j)_{j\in\mathcal{P}(i)} : \sum_{k\in\mathcal{P}(j)} (G_{jk}x_k + A_{jk}y_k) \geq b_j, \quad x_j \in \mathbb{R}^p_+, \quad y_j \in \mathbb{Z}^n_+ \quad j \in \mathcal{P}(i)\Big\}$$

for some nodes $i \in \mathcal{V}$. Note that the path set X_i includes only those constraints of $X_\mathcal{T}$ that correspond to the nodes on the path $\mathcal{P}(i)$ from the root node 0 to node i, and hence is a relaxation of the tree set $X_\mathcal{T}$. Moreover, the path set X_i is essentially the feasible region of the deterministic multi-period problem (1) with $t(i)$ periods, where $t(i)$ is the stage number of node i in the scenario tree $\mathcal{T}$. Consequently, known valid inequalities for the deterministic model (1) are valid for the path set X_i and also for the tree set $X_\mathcal{T}$. The (deterministic) valid inequalities corresponding to different path sets, called *path inequalities*, can be combined to obtain a new valid inequality, called a *tree inequality*, for the tree set. This idea has been previously explored in [9] where valid inequalities for deterministic uncapacitated lot-sizing were combined to derive valid inequalities for stochastic lot-sizing, and in [8] where valid inequalities for general deterministic two-stage integer programs were combined to obtain inequalities for two-stage stochastic integer programs. The underlying combination scheme in these papers, as well as in our approach, is a simple operation known as *pairing* [8] which is described next.

2.3 Pairing

Throughout this paper we adopt the following convention. Given two vectors a_1 and a_2 of the same dimension, the operations $\min(a_1, a_2)$ and $\max(a_1, a_2)$ are understood to be carried out component-wise. Given a vector a and a scalar b, we define $a+b = a+b\mathbb{1}$ and $\min\{a, b\} = \min\{a, b\mathbb{1}\}$, where $\mathbb{1}$ is a vector of ones of the same dimension as a. Also, since all variables are non-negative, we say that an inequality $a_1 x \geq b_1$ *dominates* another inequality $a_2 x \geq b_2$ if $a_1 \leq a_2$ and $b_1 \geq b_2$.

Theorem 1 [8] *Suppose the inequalities $g_1 x + a_1 y \geq b_1$ and $g_2 x + a_2 y \geq b_2$ with $b_1 \leq b_2$ are valid for the set $X \subset \mathbb{R}^p_+ \times \mathbb{Z}^n_+$, then the* pairing *inequality*

$$\varphi x + \phi y \geq b_2,$$

where $\varphi = \max\{g_1, g_2\}$ and $\phi = \min\{a_1 + (b_2 - b_1), \max\{a_1, a_2\}\}$, is valid for X.

The pairing inequality is a split cut that can be derived as in [4] or via the mixed-integer rounding procedure [13, 14] and, in the special case where all coefficients are nonnegative, via *mixing* [10].

Given a set of valid inequalities, the pairing operation can be carried out repeatedly to generate new valid inequalities. The order in which the inequalities are paired differentiates the inequalities. A natural order is *sequential pairing*. Given K valid inequalities

$$g_i x + a_i y \geq b_i \quad i = 1, \ldots, K$$

for a set $X \subset \mathbb{R}^p_+ \times \mathbb{Z}^n_+$, such that $b_1 \leq b_2 \leq \cdots \leq b_K$, the *sequential pairing inequality* is obtained by pairing the inequality for $i = 1$ with that for $i = 2$, and then pairing the resulting inequality with that for $i = 3$ and so on in the sequence $i = 1, \ldots, K$. In [8], problem structures where sequential pairing dominates any other pairing order has been studied. One such structure is that of two-stage stochastic integer programs.

3 From Paths to Trees

In this section, we derive a family of valid inequalities for the tree set $X_\mathcal{T}$ from a given set of path inequalities. We assume that the coefficients of the path inequalities are non-negative. This assumption can be enforced by weakening any coefficient a_j by $\max\{0, a_j\}$ (since all variables are assumed to be non-negative). We need the following additional notation regarding scenario trees. Each node i of the scenario tree $\mathcal{T}$, except the root node (indexed as $i = 0$), has a unique parent, and each non-terminal node i is the root of a subtree $\mathcal{T}(i) = (\mathcal{V}(i), \mathcal{E}(i))$, which contains all descendants of node i. Thus $\mathcal{T} = \mathcal{T}(0)$ and $\mathcal{V} = \mathcal{V}(0)$. The time period (level) corresponding to a node i will be denoted by $t(i)$. Given a subset of nodes $R \subseteq \mathcal{V}$, let $\mathcal{V}_R = \cup_{i \in R} \mathcal{P}(i)$, and $R(j) = R \cap \mathcal{V}(j)$ for each $j \in \mathcal{V}_R$.

3.1 The Tree inequalities

Theorem 2 *Given a subset of nodes $R = \{i_1, \ldots, i_K\} \subseteq \mathcal{V}$, suppose that the inequalities*

$$\sum_{j \in \mathcal{P}(i)} (g_{ij} x_j + a_{ij} y_j) \geq b_i \tag{3}$$

are valid for the path sets X_i for all $i \in R$, and are such that $g_{ij} \in \mathbb{R}^p_+$, $a_{ij} \in \mathbb{R}^n_+$ and $b_{i_1} \leq b_{i_2} \leq \cdots \leq b_{i_K}$. Then the tree inequality

$$\sum_{j \in \mathcal{V}_R} \varphi_j(R) x_j + \phi_j(R) y_j \geq b_{i_K} \tag{4}$$

is valid for the tree set $X_\mathcal{T}$, where

$$\varphi_j(R) = \max_{i \in R}\{g_{ij}\} \quad \text{and} \quad \phi_j(R) = \min\{\max_{i \in R}\{a_{ij}\}, \sum_{i_k \in R(j)} (b_{i_k} - b_{i_{k-1}})\},$$

with $b_{i_0} = 0$.

Proof: We show, by induction, that the tree inequality (4) corresponding to $R_k = \{i_1, \ldots, i_k\}$ is valid for $X_{\mathcal{T}}$ for all $k \in \{1, \ldots, K\}$.

For the base case ($k = 1$), note that, after coefficient tightening, the path inequality for X_{i_1} is

$$\sum_{j \in \mathcal{P}(i_1)} (g_{i_1 j} x_j + \min\{a_{i_1 j}, b_{i_1}\} y_j) \geq b_{i_1}. \tag{5}$$

Inequality (5) is precisely the tree inequality (4) with $R = \{i_1\}$ (in this case, $\mathcal{V}(R) = \mathcal{P}(i_1)$ and $R(j) = i_1$ for all $j \in \mathcal{V}(R)$). Moreover any path inequality is valid for $X_{\mathcal{T}}$.

Assume now that the inequality

$$\sum_{j \in \mathcal{V}_{R_k}} (\varphi_j(R_k) x_j + \phi_j(R_k) y_j) \geq b_{i_k} \tag{6}$$

is valid for $X_{\mathcal{T}}$ for some $k \in \{1, \ldots, K\}$. The path inequality for $X_{i_{k+1}}$ is

$$\sum_{j \in \mathcal{P}(i_{k+1})} (g_{i_{k+1} j} x_j + a_{i_{k+1} j} y_j) \geq b_{i_{k+1}}. \tag{7}$$

Next we pair the inequalities (6) and (7) using Theorem 1. Note that the pairing inequality has a right-hand-side equal to $b_{i_{k+1}}$ and includes variables from all the nodes in $\mathcal{V}_{R_{k+1}} = \mathcal{V}_{R_k} \cup \mathcal{P}(i_{k+1})$. We next show that the coefficients in the inequality obtained by pairing (6) and (7) are less than or equal to those of the tree inequality (4) corresponding to R_{k+1}. If $a_{ij} = a_j$ in each of the path inequalities (3), then the coefficients in the inequality obtained by pairing (6) and (7) are equal to that of the tree inequality (4).

We partition $\mathcal{V}_{R_{k+1}}$ into three sets (i) $\mathcal{P}(i_{k+1}) \setminus \mathcal{V}_{R_k}$, (ii) $\mathcal{V}_{R_k} \setminus \mathcal{P}(i_{k+1})$ and (iii) $\mathcal{V}_{R_k} \cap \mathcal{P}(i_{k+1})$.

(i) For each $j \in \mathcal{P}(i_{k+1}) \setminus \mathcal{V}_{R_k}$, we have

$$\begin{aligned} \varphi &= \max\{0, g_{i_{k+1} j}\} \\ &= \max_{i \in R_{k+1}} \{g_{ij}\} \\ &= \varphi_j(R_{k+1}), \end{aligned}$$

where the second equality follows from the fact that $g_{ij} = 0$ for all $i \in R_k$ for any $j \in \mathcal{P}(i_{k+1}) \setminus \mathcal{V}_{R_k}$. Also

$$\begin{aligned} \phi &= \min\{\phi_j(R_k) + b_{i_{k+1}} - b_{i_k}, \max\{0, a_{i_{k+1},j}\}\} \\ &= \min\{b_{i_{k+1}} - b_{i_k}, \max_{i \in R_{k+1}} \{a_{ij}\}\} \\ &= \min\{\max_{i \in R_{k+1}} \{a_{ij}\}, \sum_{i_k \in R_{k+1}(j)} (b_{i_k} - b_{i_{k-1}})\} \\ &= \phi_j(R_{k+1}), \end{aligned}$$

where the second equation follows from the fact that $a_{ij} = 0$ for all $i \in R_k$ for any $j \in \mathcal{P}(i_{k+1}) \setminus \mathcal{V}_{R_k}$, and the third equation follows from the fact that $R_{k+1}(j) = \{k+1\}$ for any $j \in \mathcal{P}(i_{k+1}) \setminus \mathcal{V}_{R_k}$.

(ii) For each $j \in \mathcal{V}_{R_k} \setminus \mathcal{P}(i_{k+1})$, we have

$$\begin{aligned} \varphi &= \max\{\varphi_j(R_k), 0\} \\ &= \max\{\max_{i \in R_k}\{g_{ij}\}, 0\} \\ &= \max_{i \in R_{k+1}} \{g_{ij}\} \\ &= \varphi_j(R_{k+1}). \end{aligned}$$

Also

$$\begin{aligned}
\phi &= \min\{\phi_j(R_k) + b_{i_{k+1}} - b_{i_k}, \max\{0, \phi_j(R_k)\}\} \\
&= \phi_j(R_k) \\
&= \min\{\max_{i \in R_k}\{a_{ij}\}, \sum_{i_k \in R_k(j)} (b_{i_k} - b_{i_{k-1}})\} \\
&= \min\{\max_{i \in R_{k+1}}\{a_{ij}\}, \sum_{i_k \in R_{k+1}(j)} (b_{i_k} - b_{i_{k-1}})\} \\
&= \phi_j(R_{k+1}),
\end{aligned}$$

where the first equation follows from the fact that $\phi_j(R_k) \geq 0$, and the last equation follows from the fact that $a_{i_{k+1}j} = 0$ and $R_{i_{k+1}}(j) = R_k(j)$ for $j \in \mathcal{V}_{R_k} \setminus \mathcal{P}(i_{k+1})$.

(iii) For each $j \in \mathcal{V}_{R_k} \cap \mathcal{P}(i_{k+1})$, we have

$$\begin{aligned}
\varphi &= \max\{\varphi_j(R_k), g_{i_{k+1}j}\} \\
&= \max\{\max_{i \in R_k}\{g_{ij}\}, g_{i_{k+1}j}\} \\
&= \max_{i \in R_{k+1}}\{g_{ij}\} \\
&= \varphi_j(R_{k+1});
\end{aligned}$$

and

$$\phi = \min\{\phi_j(R_k) + b_{i_{k+1}} - b_{i_k}, \max\{\phi_j(R_k), a_{i_{k+1}j}\}\} \tag{8}$$

where $\phi_j(R_k) = \min\{\max_{i \in R_k}\{a_{ij}\}, \sum_{i_k \in R_k(j)}(b_{i_k} - b_{i_{k-1}})\}$. Consider the following two cases.

(a) If $\max_{i \in R_k}\{a_{ij}\} \leq \sum_{i_k \in R_k(j)}(b_{i_k} - b_{i_{k-1}})$, then $\phi_j(R_k) = \max_{i \in R_k}\{a_{ij}\}$, and from (8)

$$\begin{aligned}
\phi &= \min\{\max_{i \in R_{k+1}}\{a_{ij}\}, \max_{i \in R_k}\{a_{ij}\} + b_{i_{k+1}} - b_{i_k}\} \\
&\leq \min\{\max_{i \in R_{k+1}}\{a_{ij}\}, \sum_{i_k \in R_k(j)} (b_{i_k} - b_{i_{k-1}}) + b_{i_{k+1}} - b_{i_k}\} \\
&= \min\{\max_{i \in R_{k+1}}\{a_{ij}\}, \sum_{i_k \in R_{k+1}(j)} (b_{i_k} - b_{i_{k-1}})\} \\
&= \phi_j(R_{k+1}).
\end{aligned}$$

(b) If $\max_{i \in R_k}\{a_{ij}\} > \sum_{i_k \in R_k(j)}(b_{i_k} - b_{i_{k-1}})$, then $\phi_j(R_k) = \sum_{i_k \in R_k(j)}(b_{i_k} - b_{i_{k-1}})$, and from (8)

$$\begin{aligned}
\phi &= \min\{\max\{\sum_{i_k \in R_k(j)} (b_{i_k} - b_{i_{k-1}}), a_{i_{k+1}j}\}, \sum_{i_k \in R_k(j)} (b_{i_k} - b_{i_{k-1}}) + b_{i_{k+1}} - b_{i_k}\} \\
&\leq \min\{\max\{\max_{i \in R_k}\{a_{ij}\}, a_{i_{k+1}j}\}, \sum_{i_k \in R_{k+1}(j)} (b_{i_k} - b_{i_{k-1}})\} \\
&= \min\{\max_{i \in R_{k+1}}\{a_{ij}\}, \sum_{i_k \in R_{k+1}(j)} (b_{i_k} - b_{i_{k-1}})\} \\
&= \phi_j(R_{k+1}).
\end{aligned}$$

It then follows that the tree inequality (4) corresponding to R_{k+1} is dominated by an inequality obtained by pairing the valid inequalities (6) and (7), and is therefore valid for $X_{\mathcal{T}}$. □

Example 1 Consider a tree set corresponding to the scenario tree depicted in Figure 1. Assume the three valid path inequalities corresponding to nodes 2, 3 and 4 are, respectively,

$$\begin{aligned}
2x_1 + 5y_1 + 2x_2 + 5y_2 &\geq 15, \quad (9) \\
3x_1 + 4y_1 + 3x_3 + 4y_3 &\geq 17, \quad \text{and} \quad (10) \\
3x_1 + 6y_1 + 3x_3 + 5y_3 + 9x_4 + 5y_4 &\geq 18. \quad (11)
\end{aligned}$$

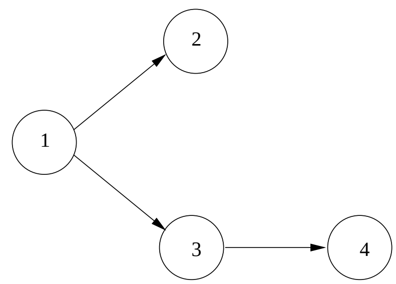

Figure 1: Scenario tree for Example 1

Then, according to Theorem 2, the two tree inequalities (4) corresponding to $R = \{2, 3\}$ and $R = \{2, 4\}$,

$$\begin{aligned} 3x_1 + 2x_2 + 3x_3 + 5y_1 + 5y_2 + 2y_3 &\geq 17 \text{ and} \\ 3x_1 + 2x_2 + 3x_3 + 9x_4 + 6y_1 + 5y_2 + 3y_3 + 3y_4 &\geq 18, \end{aligned} \tag{12}$$

are valid. □

3.2 Strengthened Tree inequalities

If the coefficients of the path inequalities are such that, for any j, $g_{ij} = g_j$ and $a_{ij} = a_j$ for all i, then the tree inequality can be strengthened.

Theorem 3 *Suppose that the inequalities*

$$\sum_{j \in \mathcal{P}(i)} (g_j x_j + a_j y_j) \geq b_i \tag{13}$$

with $g_j \in \mathbb{R}^p_+$ and $a_j \in \mathbb{R}^n_+$ are valid for the path sets X_i for all $i \in \mathcal{V}$, and (without loss of generality) $b_j \leq b_i$ for all $j \in \mathcal{P}(i)$. Given $R = \{i_1, \ldots, i_K\} \subseteq \mathcal{V}$ such that $b_{i_1} \leq b_{i_2} \leq \cdots \leq b_{i_K}$, let $i'_k = argmin\{t(j) : j \in \mathcal{P}(i_k) \text{ and } b_j > b_{i_{k-1}}\}$ for each $i_k \in R$, and let $\Omega_R = \cup_{i_k \in R} \mathcal{P}(i'_k, i_k)$ and $\Omega_R(j) = \Omega_R \cup \mathcal{V}(j)$ for any j. Then the tree inequality (4) corresponding to Ω_R dominates the tree inequality corresponding to R.

Proof: The tree inequality corresponding to the set R is

$$\sum_{j \in \mathcal{V}_R} (g_j x_j + \phi_j(R) y_j) \geq b_{i_K} \tag{14}$$

where $\phi_j(R) = \min\{a_j, \sum_{i_k \in R(j)} (b_{i_k} - b_{i_{k-1}})\}$, with $b_{i_0} = 0$. From the definition of Ω_R, it follows that $\mathcal{V}_{\Omega_R} = \mathcal{V}_R$, and since $b_j \leq b_i$ for all $j \in \mathcal{P}(i)$, we also have $\max_{i \in \Omega_R}\{b_i\} = b_{i_K}$. So the tree inequality corresponding to Ω_R is

$$\sum_{j \in \mathcal{V}_R} (g_j x_j + \phi_j(\Omega_R) y_j) \geq b_{i_K} \tag{15}$$

where $\phi_j(\Omega_R) = \min\{a_j, \sum_{i_k \in \Omega_R(j)} (b_{i_k} - b_{i_{k-1}})\}$. It is easy to see that $\phi_j(\Omega_R) \leq \phi_j(R)$ for each $j \in \mathcal{V}_R$. Thus (15) dominates (14). □

Example 1 (contd.) By relaxing the inequalities (9) and (10) to

$$3x_1 + 6y_1 + 2x_2 + 5y_2 \geq 15 \text{ and} \tag{16}$$

$$3x_1 + 6y_1 + 3x_3 + 5y_3 \geq 17, \tag{17}$$

the path inequalities (16), (17) and (11) corresponding to nodes 2, 3 and 4 satisfy the requirements of Theorem 3. Recall that the tree inequality corresponding to $R = \{2, 4\}$ is (12). The extension of R according to Theorem 3 is $\Omega_R = \{2, 3, 4\}$. The tree inequality corresponding to Ω_R is

$$3x_1 + 2x_2 + 3x_3 + 9x_4 + 6y_1 + 5y_2 + 3y_3 + y_4 \geq 18, \tag{18}$$

and it dominates (12), the tree inequality for R. □

4 The stochastic dynamic knapsack problem

The deterministic dynamic knapsack set

$$X_{\text{DK}} = \Big\{(x, y) \in \mathbb{R}_+ \times \{0,1\}^T : x + \sum_{\tau=1}^{t} a_\tau y_\tau \geq b_t \quad t = 1, \ldots, T\Big\},$$

where $a_t \in \mathbb{R}_+$ and $b_t \in \mathbb{R}_+$, has been studied in [11]. Assuming that the parameters a_t and b_t are stochastic and evolve according to the scenario tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, and using the notation already described, the *stochastic dynamic knapsack* set is

$$X_{\text{SDK}} = \Big\{(x, y) \in \mathbb{R}_+ \times \{0,1\}^{|\mathcal{V}|} : x + \sum_{j \in \mathcal{P}(i)} a_j y_j \geq b_i, \ i \in \mathcal{V}\Big\}, \tag{19}$$

where $a_i \in \mathbb{R}_+$ and $b_i \in \mathbb{R}_+$ for all $i \in \mathcal{V}$. Without loss of generality, we assume $b_j \leq b_i$ if $j \in \mathcal{P}(i)$.

The stochastic dynamic knapsack set X_{SDK} is a simple special case of the tree set $X_{\mathcal{T}}$, involving a single binary variable and a single constraint corresponding to each node of the scenario tree, and an additional continuous variable x corresponding to the root node. Considering the original node-specific constraints as the base path inequalities, we can apply Theorem 2 to obtain the valid tree inequality

$$x + \sum_{j \in \mathcal{V}_R} \phi_j(R) y_j \geq b_{i_K}, \tag{20}$$

where $R = \{i_1, \ldots, i_K\} \subseteq \mathcal{V}$ and $\phi_j(R) = \min\{a_j, \sum_{i_k \in R(j)} (b_{i_k} - b_{i_{k-1}})\}$ with $b_{i_0} = 0$. Moreover, since X_{SDK} satisfies the assumptions of Theorem 3, the tree inequality (20) can be strengthened by replacing R with Ω_R, where Ω_R is defined as in Theorem 3.

Example 2 Consider an instance of X_{SDK} where the scenario tree has 5 nodes as shown in Figure 2. The problem parameters are

$$a_1 = 40, a_2 = 15, a_3 = 20, a_4 = 20, a_5 = 40 \quad \text{and} \quad b_1 = 5, b_2 = 15, b_3 = 17, b_4 = 20, b_5 = 40.$$

From (20), we obtain the valid inequalities

$$x + 15y_1 + 10y_2 \geq 15, \quad \text{corresponding to} \quad R = \{1, 2\}, \tag{21}$$

$$x + 20y_1 + 10y_2 + 5y_3 + 5y_4 \geq 20, \quad \text{orresponding to} \quad R = \{1, 2, 4\}, \text{ and} \tag{22}$$

$$x + 40y_1 + 10y_2 + 25y_3 + 5y_4 + 20y_5 \geq 40, \quad \text{corresponding to} \quad R = \{1, 2, 4, 5\}. \tag{23}$$

The inequality (22) corresponding to $R = \{1, 2, 4\}$ is dominated by the inequality

$$x + 20y_1 + 10y_2 + 5y_3 + 3y_4 \geq 20 \tag{24}$$

corresponding to $\Omega_R = \{1, 2, 3, 4\}$, and the inequality (23) corresponding to $R = \{1, 2, 4, 5\}$ is dominated by the inequality

$$x + 40y_1 + 10y_2 + 25y_3 + 3y_4 + 20y_5 \geq 40 \tag{25}$$

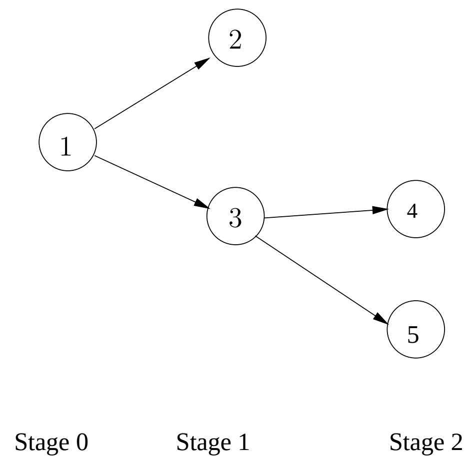

Figure 2: Scenario tree for Example 2

corresponding to $\Omega_R = \{1, 2, 3, 4, 5\}$. □

We now present some sufficient conditions under which the tree inequality (20) is facet-defining. We only consider inequalities corresponding to subsets $R \subseteq \mathcal{V}$ such that $\Omega_R = R$. We denote such subsets as $\Omega = \{b_{i_1}, \ldots, b_{i_K}\}$, and so the corresponding tree inequality is

$$x + \sum_{j \in \mathcal{V}_\Omega} \phi_j(\Omega) y_j \geq b_{i_K}, \tag{26}$$

with $\phi_j(\Omega) = \min\{a_j, \sum_{i_k \in \Omega(j)} (b_{i_k} - b_{i_{k-1}})\}$ and $b_{i_0} = 0$.

Theorem 4 *The tree inequality (26) is facet-defining for X_{SDK} if*

(1) for each $j \in \mathcal{V}_\Omega$, $a_j \geq \max\{b_i, i \in \Omega(j)\}$, and

(2) for each pair $j \in \Omega$ and $r \in \mathcal{V}(j)$, $b_j + \sum_{k \in \mathcal{P}(r) \setminus \mathcal{P}(j)} a_k \geq b_r$, and

(3) for each $j \in \mathcal{V} \setminus \mathcal{V}_\Omega$, there exists a node $s(j) \in \mathcal{P}(j) \cap \mathcal{V}_\Omega$ such that $a_{s(j)} + \sum_{k \in \mathcal{P}(r) \setminus \mathcal{V}_\Omega} a_k \geq b_r$ for each $r \in \mathcal{P}(j) \setminus \{\mathcal{V}_\Omega \cup j\}$ and $a_{s(j)} + \sum_{k \in \mathcal{P}(r) \setminus \{\mathcal{V}_\Omega \cup j\}} a_k \geq b_r$ for each $r \in \mathcal{V}(j)$.

Proof: See [6]. □

Example 2 (contd.) The inequalities (21) and (24) are facet-defining since they satisfy all three sufficient conditions. However, Theorem 4 does not specify whether inequality (25) is facet-defining since $a_3 = 20 < \max\{b_i, i \in \Omega(3)\} = b_5 = 40$, so the sufficient condition (1) is not satisfied. □

In general, Theorem 4 does not imply that the tree inequalities (26) are sufficient to describe the convex hull of X_{SDK}. Moreover, there are exponentially many tree inequalities, and we do not have an efficient separation scheme for the general case. However when the coefficients a_j are large relative to the right-hand-side coefficients we can obtain stronger results.

4.1 Special case: Large coefficients

We consider instances of X_{SDK} with large coefficients, in particular $a_j \geq \max\{b_k, k \in \mathcal{V}(j)\}$ for all $j \in \mathcal{V}$. In this case $\phi_j(\Omega) = \sum_{i_k \in \Omega(j)} (b_{i_k} - b_{i_{k-1}})$ for all $j \in \mathcal{V}$.

Theorem 5 *If $a_j \geq \max\{b_k, k \in \mathcal{V}(j)\}$ for all $j \in \mathcal{V}$, then the family of inequalities (26) corresponding to all $\Omega \subseteq \mathcal{V}$, along with $0 \leq y_j \leq 1$ for all $j \in \mathcal{V}$ describe the convex hull of X_{SDK}.*

Proof: See [6]. □

Theorem 5 generalizes the convex hull results for the deterministic case, i.e., $|\Omega| = 1$, in [2], and the case where there are only two stages, i.e., $t(j) \leq 2$ for each $j \in \mathcal{V}$, in [10].

Example 2 (contd.) If we modify the coefficients to $a_1 = a_2 = a_3 = a_4 = a_5 = 40$, then inequalities (26) corresponding to $\Omega = \{1\}, \{1,2\}, \{1,3\}, \{1,3,4\}, \{1,3,5\}, \{1,2,3\}, \{1,2,3,4\}, \{1,2,3,5\}, \{1,3,4,5\}, \{1,2,3,4,5\}$ together with $0 \leq x \leq 40$ and $0 \leq y_1, \ldots, y_5 \leq 1$ describe the convex hull of all feasible solutions. □

4.2 Separation

For the case of large coefficients, separation of the tree inequalities can be carried out by solving shortest path problems on a directed graph G with nodes $\mathcal{V}$ and arcs (i,j) for all i with $b_j > b_i$. Given a point (x^*, y^*), the separation problem of determining whether there exists a violated tree inequality can be reduced to finding a shortest path from node 0 to node k for each $k \in \mathcal{V}$ where the length of arc (i,j) is given by $\sum_{r \in \mathcal{P}(j)} (b_j - b_i) y_r^*$. This is true because a path $P = (0, i_1, i_2, \ldots, i_k)$ in G corresponds to a valid tree inequality of the form (20) with $R = \{0, i_1, i_2, \ldots, i_k\}$ since the length of the path plus x^* is equal to the left-hand side of the inequality. Therefore, there is a violated inequality with right-hand side b_{i_k} if and only if the length of a shortest path from 0 to k is less than $b_{i_k} - x^*$. Using Dijkstra's algorithm the separation problem can be solved in $O(|\mathcal{V}|^2)$ time and we can find as many as $|\mathcal{V}| - 1$ violated inequalities from the shortest paths from 0 to k for each $k \in \mathcal{V}$.

Theorem 6 *If $a_j \geq \max\{b_k, k \in \mathcal{V}(j)\}$ for all $j \in \mathcal{V}$, then there exists a polynomial-time separation algorithm for the tree inequalities (26).*

When the condition that $a_j \geq \max\{b_k, k \in \mathcal{V}(j)\}$ does not hold, the above algorithm may be used as a separation heuristic by first finding a tree inequality assuming $a_j \geq \max\{b_k, k \in \mathcal{V}(j)\}$ for all j and then tightening the coefficients of the y_j variables to $\min\{a_j, \phi_j(R)\}$.

4.3 Dominance of sequential pairing

Theorem 7 *A valid inequality generated by an arbitrary sequence of pairing operations on a subset of the original inequalities of X_{SDK} is dominated by a convex combination of the tree inequalities (26) for all $\Omega \subseteq \mathcal{V}$.*

Proof: First consider the case $a_j \geq \max\{b_k, k \in \mathcal{V}(j)\}$ for all j. In this case, the claim is certainly true since, by Theorem 5, the tree inequalities suffice to describe the convex hull of X_{SDK}. Thus, if $a_j \geq \max\{b_k, k \in \mathcal{V}(j)\}$ for all j, a valid inequality

$$x + \sum_{j \in \mathcal{V}_R} \theta_j(R) y_j \geq b(R),$$

where $b(R) = \max_{j \in R}\{b_j\}$, obtained by an arbitrary sequence of pairing operations of the original inequalities for a subset $R \subseteq \mathcal{V}$, is dominated by a convex combination of tree inequalities (26)

$$x + \sum_{j \in \mathcal{V}_{\Omega^k}} \phi_j(\Omega^k) y_j \geq b(\Omega^k) \quad k = 1, \ldots, K$$

corresponding to subsets of nodes $\Omega^1, \ldots, \Omega^K$. That is, there exists a set of weights $\lambda_1, \ldots, \lambda_K$ with $\lambda_k \geq 0$ and $\sum_{k=1}^K \lambda_k = 1$, such that for all j

$$\theta_j(R) \geq \sum_{k=1}^K \lambda_k \phi_j(\Omega^k) \;\; \text{and} \;\; b(R) \leq \sum_{k=1}^K \lambda_k b(\Omega^k). \tag{27}$$

Now, consider the case of general coefficients a_j. Observe that a valid inequality obtained by an arbitrary sequence of pairing operations on the original constraints of X_{SDK} corresponding to a subset of nodes $R \subseteq \mathcal{V}$ is of the form

$$x + \sum_{j \in \mathcal{V}_R} \min\{a_j, \theta_j(R)\} y_j \geq b(R),$$

where $\theta_j(R)$ is the coefficient if $a_j \geq \max\{b_k, k \in \mathcal{V}(j)\}$ for all j. Similarly, a tree inequality corresponding to $\Omega^k \subseteq \mathcal{V}$ is of the form

$$x + \sum_{j \in \mathcal{V}_{\Omega^k}} \min\{a_j, \phi_j(\Omega^k)\} y_j \geq b(\Omega^k),$$

where $\phi_j(R)$ is the coefficient if $a_j \geq \max\{b_k, k \in \mathcal{V}(j)\}$ for all j. Since $b(R) \leq \sum_{k=1}^{K} \lambda_k b(\Omega^k)$ from (27), we only need to verify that for each $j \in \mathcal{V}$

$$\min\{a_j, \theta_j(R)\} \geq \sum_{k=1}^{K} \lambda_k \min\{a_j, \phi_j(\Omega^k)\},$$

with $\lambda_k \geq 0$ and $\sum_{k=1}^{K} \lambda_k = 1$. Indeed, if $a_j \geq \theta_j(R)$, then we have

$$\min\{a_j, \theta_j(R)\} = \theta_j(R) \geq \sum_{k=1}^{K} \lambda_k \phi_j(\Omega^k) \geq \sum_{k=1}^{K} \lambda_k \min\{a_j, \phi_j(\Omega^k)\}$$

where the first inequality follows from (27). On the other hand, if $a_j \leq \theta_j(R)$ then

$$\min\{a_j, \theta_j(R)\} = a_j \geq \sum_{k=1}^{K} \lambda_k a_j \geq \sum_{k=1}^{K} \lambda_k \min\{a_j, \phi_j(\Omega^k)\}.$$

□

5 Stochastic lot-sizing

A multi-stage stochastic integer programming formulation of the single-item stochastic lot-sizing problem defined over a scenario tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ is (cf. [9])

$$\begin{array}{lll} \min & \sum_{i \in \mathcal{V}} p_i(\alpha_i s_i + \beta_i x_i + \gamma_i y_i) + \alpha_{a(0)} s_{a(0)} & \\ \text{s.t.} & s_{a(i)} + x_i = d_i + s_i & i \in \mathcal{V} \\ & 0 \leq x_i \leq a_i y_i & i \in \mathcal{V} \\ & s_{a(0)} \geq 0,\ s_i \geq 0,\ y_i \in \{0,1\} & i \in \mathcal{V}, \end{array}$$

where s, x and y denotes the inventory, production, and set-up variables, and the parameters α, β, γ, and a denotes holding costs, productions costs, set-up costs, and production capacities, respectively. Eliminating the inventory variables s_i for $i \in \mathcal{V}$ and using s to denote the initial inventory variable $s_{a(0)}$, the feasible region of the stochastic lot-sizing problem is

$$X_{\text{SLP}} = \Big\{(s, x, y) \in \mathbb{R}_+ \times \mathbb{R}_+^{|\mathcal{V}|} \times \{0,1\}^{|\mathcal{V}|} : s + \sum_{j \in \mathcal{P}(i)} x_j \geq d_{0i},\ x_i \leq a_i y_i,\ i \in \mathcal{V}\Big\}, \tag{28}$$

where $d_{0i} = \sum_{j \in \mathcal{P}(i)} d_j$, i.e., the cumulative demand up to node i. Replacing x_i with $a_i y_i$, we have the relaxation of X_{SLP}

$$X_{\text{RSLP}} = \Big\{(s, y) \in \mathbb{R}_+ \times \{0,1\}^{|\mathcal{V}|} : s + \sum_{j \in \mathcal{P}(i)} a_j y_j \geq b_i,\ i \in \mathcal{V}\Big\}, \tag{29}$$

where $b_i = \max_{j \in \mathcal{P}(i)} \{d_{0j}\}$. Note that X_{RSLP} is precisely the stochastic dynamic knapsack set X_{SDK}. Hence, the valid inequalities developed in Section 4 are also valid for X_{SLP}. The following lemma allows us to include the x_j variables in the valid tree inequalities for X_{RSLP}.

Lemma 1 *If $s + \sum_{j \in \mathcal{V}_R} \pi_j y_j \geq \pi_0$ is a valid inequality for X_{SLP} for some $R \subseteq \mathcal{V}$, and $S_R \subseteq \mathcal{V}_R$, then*

$$s + \sum_{j \in S_R} x_j + \sum_{j \in \overline{S}_R} \pi_j y_j \geq \pi_0 \tag{30}$$

where $\overline{S}_R = \mathcal{V}_R \setminus S_R$, is a valid inequality for X_{SLP}.

Proof: Consider a point $(s^*, x^*, y^*) \in X_{\text{SLS}}$. Now construct a point $(\widehat{s}, \widehat{x}, \widehat{y})$ such that $\widehat{x}_j = x_j^*$ and $\widehat{y}_j = y_j^*$ for each $j \in \mathcal{V} \setminus S_R$, $\widehat{x}_j = \widehat{y}_j = 0$ for each $j \in S_R$, and $\widehat{s} = s^* + \sum_{j \in S_R} x_j^*$. Then for each $i \in \mathcal{V}$,

$$\widehat{s} + \sum_{j \in \mathcal{P}(i)} \widehat{x}_i = s^* + \sum_{j \in S_R} x_i^* + \sum_{j \in \mathcal{P}(i) \setminus S_R} x_i^* \geq s^* + \sum_{j \in \mathcal{P}(i)} x_i^* \geq d_{0i}.$$

Thus $(s, \widehat{x}, \widehat{y}) \in X_{\text{SLP}}$. Then

$$\pi_0 \leq \widehat{s} + \sum_{j \in \mathcal{V}_R} \pi_j \widehat{y}_j = s^* + \sum_{j \in S_R} x_j^* + \sum_{j \in \mathcal{V}_R \setminus S_R} \pi_j y_j^*.$$

Therefore, inequality (30) is valid for X_{SLS}. □

Theorem 8 *Given a subset $R = \{i_1, \ldots, i_K\} \subseteq \mathcal{V}$, such that $b_{i_1} \leq \cdots \leq b_{i_K}$ where $b_i = \max_{j \in \mathcal{P}(i)}\{d_{0j}\}$, and a subset $S_R \subseteq \mathcal{V}_R$, the inequality*

$$s + \sum_{j \in S_R} x_j + \sum_{j \in \overline{S}_R} \phi_j(R) y_j \geq b_{i_K} \tag{31}$$

is valid for X_{SLP}, where $\overline{S}_R = \mathcal{V}_R \setminus S_R$ and $\phi_j(R) = \min\{a_j, \sum_{i_k \in R(j)}(b_{i_k} - b_{i_{k-1}})\}$ with $b_{i_0} = 0$.

Proof: The result follows immediately by applying Lemma 1 to inequality (20) for the stochastic dynamic knapsack relaxation X_{RSLP}. □

Next, we consider the tree inequalities (31) for the special case where the production quantities are uncapacitated. For the case with constant production capacities, valid inequalities derived from a mixing set relaxation of X_{SLP} are presented in [5].

5.1 The uncapacitated case

Here we assume that $a_j \geq \max\{b_k, k \in \mathcal{V}(j)\}$ for all $j \in \mathcal{V}$. Guan *et al.* [9] proposed the following class of $(\mathcal{Q}, S_{\mathcal{Q}})$ inequalities for this case.

Consider a set of nodes $\mathcal{Q} = \{i_1, i_2, \ldots, i_K\} \subset \mathcal{V}$, and let $\mathcal{Q}(j) = \mathcal{Q} \cap \mathcal{V}(j)$ for all $j \in \mathcal{V}$s, such that

(i) $d_{0i_1} \leq d_{0i_2} \leq \ldots \leq d_{0i_K}$, and

(ii) if, for any node j, $i_m, i_n \in \mathcal{Q}(j)$, then $\{i_{m+1}, i_{m+2}, \ldots, i_{n-1}\} \in \mathcal{Q}(j)$.

For each $i \in \mathcal{V}_{\mathcal{Q}}$ define

$$\begin{aligned} \overline{D}_{\mathcal{Q}}(i) &= \max\{d_{0j} : \ j \in \mathcal{Q}(i)\}, \\ \widetilde{D}_{\mathcal{Q}}(i) &= \begin{cases} 0, & \text{if } \{j : j \in \mathcal{Q} \setminus \mathcal{Q}(i) \text{ such that } d_{0j} \leq \overline{D}_{\mathcal{Q}}(i)\} = \emptyset \\ \max\{d_{0j} : \ j \in \mathcal{Q} \setminus \mathcal{Q}(i) \text{ such that } d_{0j} \leq \overline{D}_{\mathcal{Q}}(i)\}, & \text{otherwise,} \end{cases} \\ M_{\mathcal{Q}}(i) &= \max\{d_{ij} : j \in \mathcal{Q}(i)\}, \text{ and} \\ \delta_{\mathcal{Q}}(i) &= \min\left\{\overline{D}_{\mathcal{Q}}(i) - \widetilde{D}_{\mathcal{Q}}(i), M_{\mathcal{Q}}(i)\right\}. \end{aligned}$$

Then, given $S_{\mathcal{Q}} \subseteq \mathcal{V}_{\mathcal{Q}}$ and $\overline{S}_{\mathcal{Q}} = \mathcal{V}_{\mathcal{Q}} \setminus S_{\mathcal{Q}}$, the $(\mathcal{Q}, S_{\mathcal{Q}})$ inequality

$$s + \sum_{j \in S_{\mathcal{Q}}} x_j + \sum_{j \in \overline{S}_{\mathcal{Q}}} \delta_{\mathcal{Q}}(j) y_j \geq M_{\mathcal{Q}}(0) \tag{32}$$

is valid for X_{SLP}.

Guan *et al.* [9] provide sufficient conditions under which the $(\mathcal{Q}, S_{\mathcal{Q}})$ inequalities are facet-defining for the uncapacitated lot-sizing problem. For two-stage problems, these inequalities suffice to describe the convex hull [7].

Theorem 9 *The $(\mathcal{Q}, S_{\mathcal{Q}})$ inequality (32) is a tree inequality of the form (31) corresponding to $R = \mathcal{Q}$.*

Proof: The result follows from the definition of the set $\mathcal{Q}$ and noting that, in this case, for any $j \in \mathcal{V}_{\mathcal{Q}}$, $\delta_{\mathcal{Q}}(j) = \phi_j(\mathcal{Q})$. □

As observed in [5, 6], tree inequalities beyond the $(\mathcal{Q}, S_{\mathcal{Q}})$ inequalities may be needed. The following example illustrates this fact.

Example 3 Consider a stochastic uncapacitated lot-sizing problem for the scenario tree structure shown in Figure 1. Let $d_1 = 10, d_2 = 15, d_3 = 5$ and $d_4 = 20$. The tree inequality corresponding to $R = \{1, 3, 2, 4\}$ with $S_R = \{1\}$ is

$$s + x_1 + 10y_2 + 15y_3 + 10y_4 \geq 35.$$

This inequality is facet-defining. However, it is not a $(\mathcal{Q}, S_{\mathcal{Q}})$ inequality since the set $\{1, 3, 2, 4\}$ does not satisfy the necessary requirements on $\mathcal{Q}$. □

5.2 Separation

Consider the uncapacitated case first. Separation of the tree inequalities (31) corresponds to finding a subset of nodes R and a partition of $\mathcal{V}_R$ into S_R and $\overline{S}_R$. Unfortunately this does not appear to be easy, so we use a heuristic approach. Recall that if we fix $S_R = \emptyset$ then the lot-sizing tree inequalities (31) are the dynamic knapsack tree inequalities (26) and hence can be separated exactly in polynomial time by a shortest path scheme as in Section 4.2. Once we have identified the most violated dynamic knapsack tree inequality (26), i.e., a subset of nodes R, we can then set $S_R = \{j \in \mathcal{V}_R : x_j^* < \phi_j(R) y_j^*\}$, where (x^*, y^*) is the current fractional solution, to find a tree inequality (31). This heuristic can be further enhanced by setting S_R as above for each of the dynamic knapsack tree inequalities identified in the course of the separation algorithm of Section 4.2 and obtaining a resulting lot-sizing tree inequality, and then choosing the most violated lot-sizing tree inequality from among these.

For the capacitated case, we first use the above scheme to find tree inequalities assuming $a_j \geq \max\{b_k, k \in \mathcal{V}(j)\}$ and then tighten the coefficients of the y_j variables to $\min\{a_j, \phi_j(R)\}$.

6 Computational Experiments

In this section, we present computational results with a branch and cut algorithm to demonstrate the computational effectiveness of the inequalities generated by our pairing scheme on randomly generated instances of single-item uncapacitated and capacitated stochastic lot-sizing problems. All computations have been carried out on a Linux workstation with dual 2.4 GHz Intel Xeon processors and 2 GB RAM.

6.1 Instance generation

Instances were generated based on different structures of the underlying scenario trees, different ratios of the production cost to the inventory holding cost, and different ratios of the setup cost to the inventory holding cost. We assumed that the underlying scenario tree is balanced with T stages and K branches per stage. For the uncapacitated instances we used stage-branch combinations $(T, K) = (9, 2), (10, 2), (6, 3)$ and $(7, 3)$; production to holding cost ratios $\beta/h = 2$ and 4; and setup to holding cost ratios $\gamma/h = 200$ and 400.

Corresponding to each of the 16 combinations of K, T, γ/h and β/h three random instances were generated. In these instances, corresponding to each node i of the tree, the holding cost h_i is a random number uniformly distributed in the interval $[0, 10]$; the production cost β_i is uniformly distributed in the interval $[0.8(\alpha/h)\bar{h}, 1.2(\alpha/h)\bar{h}]$ where $\bar{h}$ is the average holding cost; the setup cost γ_i is uniformly distributed in the interval $[0.8(\beta/h)\bar{h}, 1.2(\beta/h)\bar{h}]$; and demand d_i is uniformly distributed in the interval $[0, 100]$. Finally, all K children of a node occur with equal probability $1/K$.

For the capacitated instances, we used $(T, K) = (9, 2)$ and $(6, 3)$. Two sizes of productions capacities a_i were used, a large capacity that is uniformly distributed in the interval $[40T, 60T]$ and a small capacity that is uniformly distributed in the interval $[20T, 40T]$. All other parameters were generated in the same way as in the uncapacitated case.

6.2 Results

We used CPLEX 8.1 in the default mode as a control and compared its performance to our customized algorithm which augments default CPLEX by repeatedly solving the linear programming relaxation and adding the most violated cut found by the separation heuristics until no more cuts can be found, at each node of the branch-and-cut tree. To get a better understanding of the value of our cuts, we also evaluated how much they improved the LP at the root node.

Computational results for the stochastic uncapacitated case are shown in Tables 1 and 2. Table 1 gives the effectiveness of the tree inequalities in tightening the LP relaxation gap at the root node. The LP relaxation gap of the original formulation without adding any cuts is shown in the column labelled "LP Gap %." It is calculated with respect to the best feasible solution found by our branch-and-cut algorithm. The column labelled "Path" corresponds to the results from adding all violated path inequalities (i.e., $|R| = 1$); the column labelled "$(\mathcal{Q}, S_{\mathcal{Q}})$" corresponds to the results after adding $(\mathcal{Q}, S_{\mathcal{Q}})$ inequalities developed; the column labelled "Tree" corresponds to the results after adding violated *tree* inequalities (31) by the heuristic separation algorithm.

For each combination of K, T, γ/h and β/h, there are two rows corresponding to the columns labelled "Path," "$(\mathcal{Q}, S_{\mathcal{Q}})$" and "Tree." The first row gives the LP relaxation gap after adding inequalities, and the second row gives the number of inequalities added. Note that all reported numbers are averages over three instances. Significant tightening of the LP relaxation is achieved via our approach. In most cases, the LP relaxation gap is reduced from over 20% to less than 1%. Furthermore, in most cases, we observe significant improvement by adding tree inequalities to the path inequalities. The tree inequalities also give better performance than the $(\mathcal{Q}, S_{\mathcal{Q}})$ inequalities, and many fewer tree inequalities are needed to get this improved performance.

The final column labelled "$|R|$" records the minimum, average and maximum number of elements in R corresponding to each tree inequality which gives an indication of how much of the scenario tree is used by each inequality. We observed that $|R|$ ranges from 2 to 40 with an average around 10. We also noticed that the average number of elements in R for the cases with $K = 2$ is less than those with $K = 3$.

Table 2 presents our branch-and-cut results. We compared the number of cuts added by default CPLEX and by our branch-and-cut scheme respectively, the relative optimality gap upon termination, the number of nodes explored (apart from the root node), and the total CPU time. For the two rows corresponding to each combination of K, T, γ/h and β/h in the table, the first one gives the performance of default CPLEX and the second one gives the performance of our branch-and-cut scheme. The reported data is averaged over three instances. In the column labelled "optimality gap," the numbers in square brackets indicate the number of instances *not* solved to default CPLEX optimality tolerance within the allotted time limit of one hour. The default CPLEX MIP solver added several types of cuts including cover cuts, flow cuts, Gomory fractional cuts and mixed integer rounding cuts. Our branch-and-cut algorithm added up to 500 tree inequalities as cuts at each node after the CPLEX default cuts have been added. For the total CPU time, as shown in the column labelled "CPU secs," we report the average CPU time for instances that are solved to default CPLEX optimality tolerance within the allotted time limit of one hour. Label "$***$" represents the case that *no* instance is solved to default CPLEX optimality tolerance within the allotted time.

Our branch-and-cut algorithm performs much better than default CPLEX. Our algorithm solves to optimality half of the instances for $K = 2$ case and 10 out of 24 instances for $K = 3$ case, while the default CPLEX can not solve any of the instances to optimality. For those instances unsolved by both algorithms, our algorithm yielded much smaller optimality gaps. Moreover, our cuts dramatically reduced the number of nodes in the branch and bound tree and, although we added many more cuts, the running times were smaller as well. Furthermore by limiting the number of cuts added as a function of tree depth, we were able to decrease the running times a bit more than those shown in Table 2.

Table 1: Results of the root node for the uncapacitated case

K	T	γ/h	β/h	LP gap %	Path	$(\mathcal{Q}, S_{\mathcal{Q}})$	Tree	$\|R\|$
2	9	200	2	16.74	2.37	0.77	0.11	$(2, 8, 28)$
					433	12440	916	
2	9	200	4	13.76	2.20	0.92	0.30	$(2, 8, 28)$
					417	13106	784	
2	9	400	2	20.67	3.54	1.04	0.13	$(2, 9, 31)$
					318	13094	1027	
2	9	400	4	18.22	3.10	1.37	0.33	$(2, 8, 24)$
					359	12760	734	
2	10	200	2	15.56	2.73	1.26	1.01	$(2, 7, 23)$
					849	15399	620	
2	10	200	4	20.12	3.97	1.50	0.92	$(2, 7, 23)$
					631	15809	1165	
2	10	400	2	12.63	2.30	0.93	0.33	$(2, 9, 30)$
					846	15254	1112	
2	10	400	4	18.90	4.66	1.96	0.76	$(2, 9, 30)$
					729	15656	1579	
3	6	200	2	19.19	3.92	0.96	0.32	$(2, 12, 34)$
					197	11059	849	
3	6	200	4	16.13	3.57	0.73	0.18	$(2, 12, 33)$
					183	10877	922	
3	6	400	2	25.04	5.11	1.67	0.16	$(2, 14, 36)$
					172	11265	1127	
3	6	400	4	22.05	4.65	1.59	0.32	$(2, 13, 40)$
					168	11405	1158	
3	7	200	2	22.01	4.17	1.96	1.10	$(2, 9, 31)$
					739	15020	1386	
3	7	200	4	17.64	3.12	1.51	1.35	$(2, 10, 30)$
					696	14507	991	
3	7	400	2	30.80	8.92	3.82	2.04	$(2, 12, 35)$
					634	14779	1656	
3	7	400	4	24.48	4.24	2.31	0.98	$(2, 12, 38)$
					638	14810	2056	

Table 2: Results of the branch-and-cut algorithm for the uncapacitated case

K	T	γ/h	β/h	# cuts	optimality gap	# nodes	CPU secs
2	9	200	2	563	0.59[3]	1657049	***
				3823	0	248	149.4
2	9	200	4	551	0.47[3]	1640825	***
				8425	0	189	894.3
2	9	400	2	596	0.99[3]	1570548	***
				14642	0.02[1]	264	956.5
2	9	400	4	521	0.92[3]	1616461	***
				16420	0.08[1]	190	437
2	10	200	2	780	1.78[3]	943455	***
				18567	0.17[2]	655	3264
2	10	200	4	1026	0.95[3]	835008	***
				21241	0.05[2]	133	3521
2	10	400	2	885	2.1[3]	891822	***
				17450	0.42[3]	946	***
2	10	400	4	858	2.02[3]	924457	***
				27642	1.31[3]	85	***
3	6	200	2	723	0.61[3]	1996296	***
				9046	0.08[1]	76	87
3	6	200	4	801	0.24[3]	1988059	***
				5545	0	156	512.5
3	6	400	2	566	0.81[3]	2608384	***
				7535	0	291	1045
3	6	400	4	546	0.65[3]	3005068	***
				9812	0.17[1]	130	195.1
3	7	200	2	1129	2.29[3]	790023	***
				29009	0.69[3]	24	***
3	7	200	4	1014	1.77[3]	828985	***
				37766	0.98[3]	45	***
3	7	400	2	945	3.62[3]	1000364	***
				25187	1.24[3]	0	***
3	7	400	4	1069	2.55[3]	1123622	***
				26690	0.82[3]	0	***

Table 3: Results of the root node for the capacitated case

Capacity	K	T	γ/h	β/h	LP gap %	Path	Tree	$\lvert R\rvert$
	2	9	200	2	14.57	2.64	0.17	$(2, 6, 19)$
						439	657	
	2	9	200	4	11.18	2.16	0.28	$(2, 6, 20)$
						404	598	
	2	9	400	2	17.54	3.28	0.24	$(2, 8, 24)$
						332	721	
	2	9	400	4	14.73	3.18	0.25	$(2, 9, 28)$
						342	1013	
$U[40T, 60T]$	3	6	200	2	13.84	5.16	1.62	$(2, 8, 27)$
						202	897	
	3	6	200	4	10.91	4.04	1.33	$(2, 8, 21)$
						208	785	
	3	6	400	2	16.06	8.00	2.80	$(2, 10, 29)$
						178	894	
	3	6	400	4	14.18	7.34	2.50	$(2, 12, 32)$
						182	1039	
	2	9	200	2	12.82	3.16	0.49	$(2, 6, 21)$
						403	696	
	2	9	200	4	9.56	2.65	0.43	$(2, 6, 20)$
						383	701	
	2	9	400	2	15.07	4.56	1.23	$(2, 8, 20)$
						316	893	
	2	9	400	4	12.33	4.06	1.03	$(2, 8, 24)$
						339	944	
$U[20T, 40T]$	3	6	200	2	12.36	4.35	0.48	$(2, 8, 24)$
						189	773	
	3	6	200	4	8.64	3.85	0.43	$(2, 8, 22)$
						185	877	
	3	6	400	2	14.08	6.35	1.19	$(2, 9, 25)$
						162	918	
	3	6	400	4	11.28	5.79	1.12	$(2, 10, 28)$
						175	1004	

Tables 3 and 4 present results for the capacitated case. We also tested three instances for each combination. Table 3 shows the optimality gap reduction after adding path inequalities and the substantially bigger reductions after adding tree inequalities at the root node. For the branch-and-cut algorithm, as shown in Table 4, default CPLEX cannot solve any of the instances to optimality while our algorithm solves 16 out of 24 instances to optimality, including all two branch instances. For those unsolved instances, our algorithm obtains smaller optimality gaps and all final gaps are smaller than 0.5%. For the small capacitated case, 21 out of the 24 instances are solved to optimality by our algorithm while default CPLEX can only solve 4 out of the 24 instances to optimality. The final optimality gaps by our approach are within 0.15%.

7 Conclusions and future research

We have presented a general method for generating valid inequalities for multi-stage stochastic integer programs based on combining inequalities that are valid for the individual scenarios. We have applied the method to a stochastic version of a dynamic knapsack problem and to stochastic lot sizing problems. Our computational results show that these new inequalities are very effective in a branch-and-cut algorithm and give much better results than default CPLEX. Since multi-stage stochastic integer programs are very difficult to solve, and arise in many domains, including network reliability, routing, capacity planning and scheduling, we are now investigating the application of our method to different structural models. Decomposition methods involving Lagrangian relaxation [3, 15] and column generation [12, 17, 18] have been very effective in solving various classes of multi-stage stochastic integer programs. Integration of the proposed cut generation scheme within such decomposition frameworks is an important unresolved issue.

Table 4: Results of the branch-and-cut algorithm for the capacitated case

Capacity	K	T	γ/h	β/h	# cuts	Optimality gap	# nodes	CPU secs
$U[40T, 60T]$	2	9	200	2	590	0.55[3]	1680099	***
					2695	0	194	73.3
	2	9	200	4	569	0.36[3]	1682063	***
					4567	0	215	121.2
	2	9	400	2	538	1.11[3]	1727792	***
					6498	0	208	244.4
	2	9	400	4	551	0.85[3]	1801839	***
					10789	0	257	821.9
	3	6	200	2	487	0.37[3]	2370519	***
					5300	0	222	204.4
	3	6	200	4	530	0.18[3]	2167495	***
					5057	0	173	215.6
	3	6	400	2	561	1.1[3]	2164056	***
					24018	0.45[3]	245	***
	3	6	400	4	586	0.94[3]	2053463	***
					24398	0.47[3]	284	***
$U[20T, 40T]$	2	9	200	2	561	0.45[3]	1762030	***
					3396	0	248	138.6
	2	9	200	4	589	0.30[3]	1732004	***
					4328	0	266	269.1
	2	9	400	2	653	0.82[3]	1667112	***
					6915	0	414	626.3
	2	9	400	4	573	0.63[3]	1806365	***
					8940	0	497	835.6
	3	6	200	2	630	0.23[2]	1494500	2444.6
					10948	0.15[1]	211	1269.3
	3	6	200	4	520	0.21[2]	1800160	2404.0
					5096	0	209	222.7
	3	6	400	2	571	0.26[2]	1504807	2428.7
					14879	0.15[1]	230	1448
	3	6	400	4	483	0.24[2]	1835754	2411.3
					11406	0.06[1]	384	1323.3

References

[1] S. Ahmed, A. J. King, and G. Parija. A multi-stage stochastic integer programming approach for capacity expansion under uncertainty. *Journal of Global Optimization*, 26:3–24, 2003.

[2] I. Barany, T. van Roy, and L. A. Wolsey. Uncapacitated lot-sizing: The convex hull of solutions. *Mathematical Programming Study*, 22:32–43, 1984.

[3] C. C. Carøe and R. Schultz. Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24:37 45, 1999.

[4] W. Cook, R. Kannan, and A. J. Schrijver. Chvátal closures for mixed integer programming problems. *Mathematical Programming*, 47:155–174, 1990.

[5] M. Di Summa and L. A. Wolsey. Lot-sizing on a tree. CORE Discussion Paper, 2006/44, 2006.

[6] Y. Guan. *Pairing Inequalities and Stochastic Lot-Sizing Problems: A Study in Integer Programming.* PhD thesis, Georgia Institute of Technology, 2005.

[7] Y. Guan, S. Ahmed, A. J. Miller, and G. L. Nemhauser. On formulations of the stochastic uncapacitated lot-sizing problem. *Operations Research Letters*, 34:241–250, 2006.

[8] Y. Guan, S. Ahmed, and G. L. Nemhauser. Sequential pairing of mixed integer inequalities. *Discrete Optimization*, to appear, 2006.

[9] Y. Guan, S. Ahmed, G. L. Nemhauser, and A. J. Miller. A branch-and-cut algorithm for the stochastic uncapacitated lot-sizing problem. *Mathematical Programming*, 105:55–84, 2006.

[10] O. Günlük and Y. Pochet. Mixing MIR inequalities for mixed integer programs. *Mathematical Programming*, 90:429–457, 2001.

[11] M. Loparic, H. Marchand, and L. A. Wolsey. Dynamic knapsack sets and capacitated lot-sizing. *Mathematical Programming*, 95:53–69, 2003.

[12] G. Lulli and S. Sen. A branch and price algorithm for multi-stage stochastic integer programs with applications to stochastic lot sizing problems. *Management Science*, 50:786–796, 2004.

[13] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, New York, 1988.

[14] G. L. Nemhauser and L. A. Wolsey. A recursive procedure for generating all cuts for 0-1 mixed integer programs. *Mathematical Programming*, 46:379–390, 1990.

[15] M. P. Nowak and W. Römisch. Stochastic Lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty. *Annals of Operations Research*, 100:251–272, 2000.

[16] W. Römisch and R. Schultz. Multistage stochastic integer programs: An introduction. In *Online Optimization of Large Scale Systems* (M. Grötschel, S.O. Krumke, J. Rambau eds.), Springer-Verlag, pages 579–598, 2001.

[17] S. Sen. Algorithms for stochastic mixed-integer programming models. In *Handbook of Discrete Optimization,* (K. Aardal, G.L. Nemhauser, and R. Weismantel eds.), North-Holland Publishing Co., pages 515–558, 2005.

[18] K. Singh, A.B. Philpott, and K. Wood. Dantzig-Wolfe decomposition for solving multi-stage stochastic capacity planning problems. submitted to *Operations Research*, 2005.

[19] S. Takriti, J. R. Birge, and E. Long. A stochastic model for the unit commitment problem. *IEEE Transactions on Power Systems*, 11:1497–1508, 1996.

Supplement: Proofs of Theorems 4 and 5

Theorem 4 *Inequality (26) is facet-defining for* $X_{\rm SDK}$ *if*

(1) for each $j \in \mathcal{V}_\Omega$, $a_j \geq \max\{b_i, i \in \Omega(j)\}$,

(2) for each pair $j \in \Omega$ *and* $r \in \mathcal{V}(j)$, $b_j + \sum_{k\in\mathcal{P}(r)\setminus\mathcal{P}(j)} a_k \geq b_r$,

(3) for each $j \in \mathcal{V} \setminus \mathcal{V}_\Omega$, *there exists a* $s(j) \in \mathcal{P}(j) \cap \mathcal{V}_\Omega$ *such that* $a_{s(j)} + \sum_{k\in\mathcal{P}(r)\setminus\mathcal{V}_\Omega} a_k \geq b_r$ *for each* $r \in \mathcal{P}(j) \setminus \{\mathcal{V}_\Omega \cup j\}$ *and* $a_{s(j)} + \sum_{k\in\mathcal{P}(r)\setminus\{\mathcal{V}_\Omega\cup j\}} a_k \geq b_r$ *for each* $r \in \mathcal{V}(j)$.

Proof: We construct $\dim(X) = n+1$ linearly independent vectors belonging to X that satisfy (26) at equality.

We construct a vector corresponding to each of the $n+1$ variables. Let e^x and e^{y_j} be unit vectors in $\mathbb{R}^{n+1}$ corresponding to the coordinates x and y_j for $j = 1, \ldots, n$. For each $j \in \mathcal{V}_\Omega$, let $\rho(j) = \min\{k : k \in \Omega(j)\}$ and $\Phi(j) = \{i \in \Omega \cup a(j) : b_i \leq b_{\rho(j)-1}\}$. Note here, we have $b_{a(j)} \leq b_{\rho(j)-1}$ according to the definition of Ω and therefore $a(j) \in \Phi(j)$. Define $\Psi(j) = \cup_{k\in\Phi(j)}\mathcal{P}(k)$ and $\Lambda(j) = \cup_{k\in\Psi(j)}\mathcal{C}(k) \setminus \Psi(j)$. The vectors are denoted by $\{u^j\}_{j=0}^n$ and are constructed as follows.

(i) Vector u^0 corresponds to variable x and is given by

$$u^0 = b_Q e^x + \sum_{r\in\mathcal{V}\setminus\mathcal{V}_\Omega} e^{y_r}.$$

(ii) For each y_j where $j \in \mathcal{V}_\Omega$, the corresponding vector u^j is given by

$$u^j = b_{\rho(j)-1} e^x + \sum_{r\in\Lambda(j)} e^{y_r} + \sum_{r\in\mathcal{V}\setminus\mathcal{V}_\Omega} e^{y_r}.$$

(iii) For each y_j where $j \in \mathcal{V} \setminus \mathcal{V}_\Omega$, the corresponding vector u^j is given by

$$u^j = u^{s(j)} - e^j.$$

Feasibility: We need to show that $\{u^j\}_{j=0}^n$ satisfies (19) for each $i \in \mathcal{V}$.

(i) The vector u^0 clearly satisfies (19) corresponding to each $i \in \mathcal{V}_\Omega$ since the left-hand-side of (19) is $b_Q \geq b_i$ for each $i \in \mathcal{V}_\Omega$. Corresponding to each $i \in \mathcal{V} \setminus \mathcal{V}_\Omega$, let $\xi(i) = \text{argmax}\{b_k : k \in \mathcal{P}(i) \cap \Omega\}$ and the left-hand-side of (19) is

$$\begin{aligned} & b_Q + \textstyle\sum_{k\in\mathcal{P}(i)\setminus\mathcal{P}(\xi(i))} a_k \\ \geq\ & b_{\xi(i)} + \textstyle\sum_{k\in\mathcal{P}(i)\setminus\mathcal{P}(\xi(i))} a_k \\ \geq\ & b_i, \end{aligned}$$

where the first inequality follows from the fact that $b_Q \geq b_{\xi(i)}$ since $\xi(i) \in \Omega$ and the second inequality follows from condition (2).

(ii) It is easy to verify that the vector u^j corresponding to each $j \in \mathcal{V}_\Omega$ satisfies (19) for each $i \in \Psi(j)$. The reason is that the left-hand-side value is greater than or equal to $b_{\rho(j)-1} \geq b_i$ for each $i \in \Psi(j)$.

For each $i \in \mathcal{V}_\Omega \setminus \Psi(j)$, let $\lambda(i) = \{k : k \in \mathcal{P}(i) \cap \Lambda(j)\}$. Then, the left-hand-side of (19) is

$$b_{\rho(j)-1} + a_{\lambda(i)} \geq b_i$$

where the inequality follows from condition (1).

For each $i \in \mathcal{V} \setminus \mathcal{V}_\Omega$, the left-hand-side of (19) for the case that $\lambda(i)$ exists is

$$\begin{aligned} \geq\ & b_{\rho(j)-1} + a_{\lambda(i)} + \textstyle\sum_{k\in\mathcal{P}(i)\setminus\mathcal{P}(\xi(i))} a_k \\ \geq\ & b_{\xi(i)} + \textstyle\sum_{k\in\mathcal{P}(i)\setminus\mathcal{P}(\xi(i))} a_k \\ \geq\ & b_i \end{aligned}$$

where the second inequality follows from condition (1) and the last inequality follows from condition (2). Note here we can also provide a similar argument if $\lambda(i)$ does not exist.

(iii) The vector u^j corresponding to each $j \in \mathcal{V} \setminus \mathcal{V}_\Omega$ satisfies (19) for each $i \in \mathcal{V}_\Omega$. It follows from the fact that $s(j) \in \mathcal{V}_\Omega$ and $j \in \mathcal{V} \setminus \mathcal{V}_\Omega$. Condition (3) shows that u^j satisfies (19) for each $i \in \mathcal{V} \setminus \mathcal{V}_\Omega$.

Tightness: We need to show that $\{u^j\}_{j=0}^n$ satisfies (26) at equality.

(i) It is easy to verify that the vector u^0 satisfies (26) at equality.

(ii) The vector u^j for each $j \in \mathcal{V}_\Omega$ satisfies (26) at equality since the left-hand-side of (26) is

$$\begin{aligned} &= b_{\rho(j)-1} + \textstyle\sum_{r \in \Lambda(j)} \phi_r(\Omega) \\ &= b_{\rho(j)-1} + \textstyle\sum_{r \in \Lambda(j)} \sum_{i \in \Omega(r)} (b_i - b_{i-1}) \\ &= b_Q \end{aligned}$$

where the second equation follows from the fact that $a_r \geq \sum_{i \in \Omega(r)} (b_i - b_{i-1})$ for each $r \in \Lambda(j)$. This is because that $r \in \Lambda(j) \subseteq \mathcal{V}_Q$ and based on condition (i), we have $a_r \geq \max\{b_i, i \in \Omega(r)\} \geq \sum_{i \in \Omega(r)} (b_i - b_{i-1})$. The last equation follows from the fact that $\Omega(r_1) \cap \Omega(r_2) = \emptyset$ if $r_1, r_2 \in \Lambda(j)$ and $k \in \cup_{r \in \Lambda(j)} \Omega(r)$ for each $k \in \Omega$ such that $b_k \geq b_j$.

(iii) The vector u^j for each $j \in \mathcal{V} \setminus \mathcal{V}_\Omega$ satisfies (26) at equality since $u^{s(j)}$ satisfies (26) at equality based on (ii) and $u^j = u^{s(j)} - e^j$.

Linear independence: To verify the linear independence of the $n+1$ vectors $\{u^j\}_{j=0}^n$, observe that we can first obtain $|\mathcal{V} \setminus \mathcal{V}_\Omega|$ unit vectors by getting $e^{y_j} = u^j - u^{s(j)}$ corresponding to each $j \in \mathcal{V} \setminus \mathcal{V}_\Omega$. Besides this, for each $j \in \mathcal{V}_\Omega \cup \{0\}$, let

$$v^j = u^j - \sum_{r \in \mathcal{V} \setminus \mathcal{V}_\Omega} e^{y_r}.$$

We can form a matrix where the $|\mathcal{V}_\Omega|$ vectors form the rows of the matrix and the vector corresponding to i will be placed above the vector corresponding to j if $\max\{b_k : k \in \mathcal{V}_\Omega(i)\} > \max\{b_k : k \in \mathcal{V}_\Omega(j)\}$ or $\min\{b_k : k \in \mathcal{V}_\Omega(i)\} > \min\{b_k : k \in \mathcal{V}_\Omega(j)\}$ if $\max\{b_k : k \in \mathcal{V}_\Omega(i)\} = \max\{b_k : k \in \mathcal{V}_\Omega(j)\}$. Each column corresponds to each node in $\mathcal{V}_\Omega$. Column i is placed ahead of column j if $\max\{b_k : k \in \mathcal{V}_\Omega(i)\} > \max\{b_k : k \in \mathcal{V}_\Omega(j)\}$ or $\min\{b_k : k \in \mathcal{V}_\Omega(i)\} > \min\{b_k : k \in \mathcal{V}_\Omega(j)\}$ if $\max\{b_k : k \in \mathcal{V}_\Omega(i)\} = \max\{b_k : k \in \mathcal{V}_\Omega(j)\}$. From the definition of $\Lambda(j)$ and the construction process of each u^j, we can easily observe that these vectors form a lower triangle matrix. Therefore, all these $n+1$ vectors are linearly independent. □

Theorem 5 *If $a_j \geq \max\{b_k, k \in \mathcal{V}(j)\}$ for each $j \in \mathcal{V}$, then the family of inequalities (26) for all $\Omega \subseteq \mathcal{V}$, together with $0 \leq x \leq b_\mathcal{V}$ and $0 \leq y_j \leq 1$ for each $j \in \mathcal{V}$ describe the convex hull of X_{SDK}.*

Proof: We notice that if $a_j \geq \max\{b_k, k \in \mathcal{V}(j)\}$, then all inequalities (26) are valid and facet-defining based on Theorem 4 and $\phi_j(\Omega) = \sum_{i \in \Omega(j)} (b_i - b_{i-1})$ for each $j \in \mathcal{V}_\Omega$. We also notice that inequalities (19) are dominated by inequalities (26). We only need to show no fractional extreme points exist after adding inequalities (26) to X_{SDK}. In the following, we prove this by contradiction and assume $u^0 = \{x^0, y_1^0, \ldots, y_n^0\}$ is an extreme point that contains fractional elements.

First, assume there is no inequality in (26) such that u^0 satisfies it at equality. Without loss of generality, assume the jth element of u^0 is fractional. Then there exists two points $u^1 = u^0 + \epsilon e^{y_j}$ and $u^2 = u^0 - \epsilon e^{y_j}$ feasible for X_{SDK}. It contradicts with the assumption that u^0 is an extreme point since $u^0 = (u^1 + u^2)/2$.

If there are some inequalities in (26) such that u^0 satisfies them at equality, we define the set of nodes corresponding to the right-hand-side of each inequality as Φ. That is,

$$\Phi = \{j \in \mathcal{V} : x^0 + \sum_{k \in \mathcal{V}_\Omega} \phi_k(\Omega) y_k^0 = b_j \text{ for some } \Omega \subseteq \mathcal{V}\}.$$

Let $\alpha^* = \text{argmin}\{b_j : j \in \Phi\}$ and Ω_j be any set corresponding to each $j \in \Phi$ such that $x^0 + \sum_{k \in \mathcal{V}_{\Omega_j}} \phi_k(\Omega_j) y_k^0 = b_j$. In the following, we complete the proof in several steps.

Claim 1: $\alpha^* \in \Omega_j$ **for each** $j \in \Phi$
Proof: If not, there exists a node $\beta^* \in \Phi$ such that $\Omega_{\beta^*} \cap \Phi = \beta^*$ and $j \notin \Phi$ for each $j \in \mathcal{P}(\beta^*) \setminus \beta^*$. Without loss of generality, let $\Omega_{\alpha^*} = (\alpha^1, \alpha^2, \ldots, \alpha^{k_1}, \alpha^*)$ and $\Omega_{\beta^*} = (\beta^1, \beta^2, \ldots, \beta^{k_2}, \beta^*)$. We have

$$x^0 + \sum_{k \in \mathcal{V}_\Gamma} \phi_k(\Gamma) y_k^0 > b_{k_2} \text{ where } \Gamma = \{\beta^1, \beta^2, \ldots, \beta^{k_2}\} \tag{33}$$

and

$$x^0 + \sum_{k \in \mathcal{V}_{\Omega_{\beta^*}}} \phi_k(\Omega_{\beta^*}) y_k^0 = b_{\beta^*} \tag{34}$$

That is,

$$\begin{aligned} b_{\beta^*} &= x^0 + \sum_{k \in \mathcal{V}_{\Omega_{\beta^*}}} \phi_k(\Omega_{\beta^*}) y_k^0 \\ &= x^0 + \sum_{k \in \mathcal{V}_\Gamma} \phi_k(\Gamma) y_k^0 + (b_{\beta^*} - b_{k_2}) \sum_{k \in \mathcal{P}(\beta^*)} y_k^0 \\ &> b_{k_2} + (b_{\beta^*} - b_{k_2}) \sum_{k \in \mathcal{P}(\beta^*)} y_k^0 \end{aligned}$$

where the first equation follows from (34) and the inequality follows from (33). Thus, we have

$$\sum_{k \in \mathcal{P}(\beta^*)} y_k^0 < 1. \tag{35}$$

Now consider the inequality corresponding to set $\Theta = \{\alpha^1, \alpha^2, \ldots, \alpha^{k_1}, \alpha^*, \beta^*\}$ and we have

$$\begin{aligned} & x^0 + \sum_{k \in \mathcal{V}_\Theta} \phi_k(\Theta) y_k^0 \\ = \; & x^0 + \sum_{k \in \mathcal{V}_{\Omega_{\alpha^*}}} \phi_k(\Omega_{\alpha^*}) y_k^0 + (b_{\beta^*} - b_{\alpha^*}) \sum_{k \in \mathcal{P}(\beta^*)} y_k^0 \\ = \; & b_{\alpha^*} + (b_{\beta^*} - b_{\alpha^*}) \sum_{k \in \mathcal{P}(\beta^*)} y_k^0 \\ < \; & b_{\beta^*}, \end{aligned}$$

where the inequality follows from (35). It contradicts with the fact that $x^0 + \sum_{k \in \mathcal{V}_\Theta} \phi_k(\Theta) y_k^0 \geq b_{\beta^*}$ is a valid inequality. Therefore, the conclusion holds.

Note here, According to the definition of α^*, we have $x^0 + \sum_{k \in \mathcal{V}_\Lambda} \phi_k(\Lambda) y_k^0 > b_{k_1}$ where $\Lambda = \{\alpha^1, \alpha^2, \ldots, \alpha^{k_1}\}$. Then similarly we have

$$\sum_{k \in \mathcal{P}(\alpha^*)} y_k^0 < 1 \tag{36}$$

since

$$\begin{aligned} b_{\alpha^*} &= x^0 + \sum_{k \in \mathcal{V}_{\Omega_{\alpha^*}}} \phi_k(\Omega_{\alpha^*}) y_k^0 \\ &= x^0 + \sum_{k \in \mathcal{V}_\Lambda} \phi_k(\Lambda) y_k^0 + (b_{\alpha^*} - b_{k_1}) \sum_{k \in \mathcal{P}(\alpha^*)} y_k^0 \\ &> b_{k_1} + (b_{\alpha^*} - b_{k_1}) \sum_{k \in \mathcal{P}(\alpha^*)} y_k^0, \end{aligned}$$

which will be used later. $\square$

Claim 2:

$$\sum_{k\in\mathcal{P}(j)} y_k^0 = 1 \tag{37}$$

for each $j \in \Phi \setminus \{\alpha^*\}$**.**

Proof: Note here, for each $j \in \mathcal{V}$ such that $b_j \geq b_{\alpha^*}$, since $\alpha^* \in \Phi$ and the inequality corresponding to the set $\Omega_j = \Omega_{\alpha^*} \cup \{j\}$ is

$$\begin{aligned} & x^0 + \sum_{k\in\mathcal{V}_{\Omega_j}} \phi_k(\Omega_j) y_k^0 \\ = \; & x^0 + \sum_{k\in\mathcal{V}_{\Omega_{\alpha^*}}} \phi_k(\Omega_{\alpha^*}) y_k^0 + (b_j - b_{\alpha^*}) \sum_{k\in\mathcal{P}(j)} y_k^0 \\ \geq \; & b_j. \end{aligned}$$

Then, we have

$$\sum_{k\in\mathcal{P}(j)} y_k^0 \geq 1 \tag{38}$$

for each $j \in \mathcal{V} \setminus \{\alpha^*\}$ such that $b_j \geq b_{\alpha^*}$.

We also notice that for each $j \in \Phi$ and assuming $\Omega_j = \{\alpha^1, \ldots, \alpha^r, j\}$, we have

$$\begin{aligned} b_j &= x^0 + \sum_{k\in\mathcal{V}_{\Omega_j}} \phi_k(\Omega_j) y_k^0 \\ &= x^0 + \sum_{k\in\mathcal{V}_{\Omega_j\setminus\{j\}}} \phi_k(\Omega_j \setminus \{j\}) y_k^0 + (b_j - b_{\alpha^r}) \sum_{k\in\mathcal{P}(j)} y_k^0 \\ &\geq b_{\alpha^r} + (b_j - b_{\alpha^r}) \sum_{k\in\mathcal{P}(j)} y_k^0. \end{aligned}$$

That is, we have

$$\sum_{k\in\mathcal{P}(j)} y_k^0 \leq 1 \tag{39}$$

for each $j \in \Phi$. Combining (38) and (39), we have

$$\sum_{k\in\mathcal{P}(j)} y_k^0 = 1$$

for each $j \in \Phi \setminus \{\alpha^*\}$. $\square$

Claim 3: If there is a $j \in \Phi$ **such that** $\alpha^* \in \mathcal{P}(j)$**, then** $r \in \Phi$ **for each** $r \in \mathcal{P}(j) \setminus \mathcal{P}(\alpha^*)$**.**

Proof: Since $j \in \Phi$, then according to (37), we have $\sum_{k\in\mathcal{P}(j)} y_k^0 = 1$. For each $r \in \mathcal{P}(j) \setminus \mathcal{P}(\alpha^*)$, assuming $\Omega_r = \Omega_{\alpha^*} \cup \{r\}$, we have

$$\begin{aligned} b_r &= x^0 + \sum_{k\in\mathcal{V}_{\Omega_r}} \phi_k(\Omega_r) y_k^0 \\ &= b_{\alpha^*} + (b_r - b_{\alpha^*}) \sum_{k\in\mathcal{P}(r)} y_k^0 \\ &\leq b_{\alpha^*} + (b_r - b_{\alpha^*}) \sum_{k\in\mathcal{P}(j)} y_k^0 \\ &= b_r \end{aligned}$$

where the inequality follows from the fact that $r \in \mathcal{P}(j)$ and the last equality follows the fact that $\sum_{k\in\mathcal{P}(j)} y_k^0 = 1$. The above inequality shows that $\sum_{k\in\mathcal{P}(r)} y_k^0 = \sum_{k\in\mathcal{P}(j)} y_k^0 = 1$ for each $r \in \mathcal{P}(j) \setminus \mathcal{P}(\alpha^*)$. Therefore, $r \in \Phi$ for each $r \in \mathcal{P}(j) \setminus \mathcal{P}(\alpha^*)$. Then, we have $\mathcal{P}(j) \cap \mathcal{C}(\alpha^*) \in \Phi$. Since $j \in \Phi$, following from (37), we have

$$y_k^0 = 0 \text{ for each } k \in \mathcal{P}(j) \setminus (\mathcal{P}(\alpha^*) \cup \mathcal{C}(\alpha^*)). \tag{40}$$

Similarly, for any pair $(i, j) \in \Phi$ such that $i \in \mathcal{P}(j)$ and $i \neq \alpha^*$, we have $\sum_{k\in\mathcal{P}(i)} y_k^0 = 1$ and $\sum_{k\in\mathcal{P}(j)} y_k^0 = 1$ follows from (37) and then

$$y_k^0 = 0 \text{ for each } k \in \mathcal{P}(j) \setminus \mathcal{P}(i). \tag{41}$$

Based on the results obtained from above Step 1 and Step 2. In the following step, we show that no fractional solution points exist. □

Now we show that no fractional solution exists. At first, let $\mathcal{V}_\Phi = \cup_{j\in\Phi}\mathcal{P}(j)$. If there exists a $k \in \mathcal{V} \setminus \mathcal{V}_\Phi$ such that y_k^0 is fractional, there are two points $u^1 = u^0 + \epsilon e^{y_k}$ and $u^2 = u^0 - \epsilon e^{y_k}$ feasible for X_{SDK}. It contradicts with the assumption that u^0 is an extreme point since $u^0 = (u^1 + u^2)/2$.

Now we only need to prove that y_k^0 is not fractional for each $k \in \mathcal{V}_\Phi$. Based on (36) and (39), we have

$$y_k^0 < 1 \text{ for each } k \in \mathcal{P}(\alpha^*) \setminus \{\alpha^*\} \text{ and } y_k^0 \leq 1 \text{ for each } k \in \mathcal{V}_\Phi \setminus \mathcal{P}(\alpha^*).$$

We prove the claim based on the analysis of three cases.

Case 1: $x^0 = 0$. Then, $x^0 + d_1 y_1^0 = d_1 y_1^0 \leq d_1$, which implies that $y_1^0 = 1$ and $\alpha^* = 1$. Based on (37), we have $y_k^0 = 0$ for each $k \in \mathcal{V}_\Phi \setminus \{\alpha^*\}$.

Case 2: $x^0 \neq 0$ and $\alpha^* = 1$. Following (40), we have $y_k^0 = 0$ for each $k \in \mathcal{V}_\Phi \setminus \mathcal{C}(1)$. If there are fractional solutions, then $0 < y_1^0 < 1$ and $y_1^0 + y_j^0 = 1$ for each $j \in C(1) \cap \mathcal{V}_\Phi$. Thus, $y_j^0 = y_k^0$ for any pair $(j, k) \in C(1) \cap \mathcal{V}_\Phi$. Then, any valid inequalities (26) with the right-hand-side value b_ℓ where $\ell \in \mathcal{C}(1) \cap \mathcal{V}_\Phi$ will be equivalent to $x^0 + b_\ell y_1^0 + (b_\ell - b_1) y_\ell^0 = b_\ell$. Then two alternative points $u^1 = u^0 + \epsilon b_1 e^x - \epsilon e^{y_1} + \sum_{k\in\mathcal{C}(1)\cap\mathcal{V}_\Phi} \epsilon e^{y_k}$ and $u^2 = u^0 - \epsilon b_1 e^x + \epsilon e^{y_1} - \sum_{k\in\mathcal{C}(1)\cap\mathcal{V}_\Phi} \epsilon e^{y_k}$ are feasible for X_{SDK}. It contradicts with the assumption that u^0 is an extreme point since $u^0 = (u^1 + u^2)/2$.

Case 3: $x^0 \neq 0$ and $\alpha^* \neq 1$. Let Φ_1 be the set of nodes in $\Phi \setminus \{\alpha^*\}$ such that no nodes in $\mathcal{V}_{\Phi_1} \setminus \{\Phi_1 \cup \{\alpha^*\}\}$ belong to set Φ. That is, $\mathcal{V}_{\Phi_1} \cap \Phi = \Phi_1 \cup \{\alpha^*\}$. Similarly, let Φ_2 be the set of nodes in Φ such that $\cup_{j\in\Phi_2}\mathcal{V}(j) \cap \Phi = \Phi_2$. Based on (41), we have $y_k^0 = 0$ for each $k \in \mathcal{V}_{\Phi_2} \setminus \mathcal{V}_{\Phi_1}$. Let $\Phi_1' = \{j \in \Phi_1 :$ there exists a node $k \in \mathcal{P}(j)$ such that $0 < y_k^0 < 1\}$. Note here we have $\sum_{k\in\mathcal{P}(j)} y_k^0 = 1$ for each $j \in \Phi_1'$ according to (37). Then, there exists at least one pair $(k_1, k_2) \in \mathcal{P}(j)$ for each $j \in \Phi_1'$ such that $0 < y_{k_1}^0, y_{k_2}^0 < 1$.

Now, we initialize two sets $\Pi_1 = \emptyset$, $\Pi_2 = \emptyset$ and label each node $j \in \Phi_1'$ be zero (i.e., $\ell(j) = 0$ for each $j \in \Phi_1'$). Then, for each element $j \in \Phi_1'$ according to the nondecreasing sequence of b_j, we do the following steps.

(1) Let $s(j) = \text{argmin}\{k \in \mathcal{P}(j) \setminus \Pi_1 : 0 < y_k^0 < 1\}$.

(2) If $\ell(j) = 0$, let $\Pi_1 = \Pi_1 \cup \{s(j)\}$, update $\ell(j) = 1$ and $\ell(r) = 1$ for each $r \in \mathcal{V}(j) \cap \Phi_1'$. Go back to (1).

(3) Else If $\ell(j) = 1$, let $\Pi_2 = \Pi_2 \cup \{s(j)\}$, update $\ell(j) = 2$ and $\ell(r) = 2$ for each $r \in \mathcal{V}(j) \cap \Phi_1'$. Stop.

We can easily observe there exists two points

$$u^1 = u^0 + \epsilon \sum_{k\in\Pi_1\cap\mathcal{P}(\alpha^*)} \phi_k(\Omega_{\alpha^*}) e^x - \epsilon \sum_{k\in\Pi_2\cap\mathcal{P}(\alpha^*)} \phi_k(\Omega_{\alpha^*}) e^x - \epsilon \sum_{k\in\Pi_1} e^{y_k} + \epsilon \sum_{k\in\Pi_2} e^{y_k}$$

and

$$u^2 = u^0 - \epsilon \sum_{k\in\Pi_1\cap\mathcal{P}(\alpha^*)} \phi_k(\Omega_{\alpha^*}) e^x + \epsilon \sum_{k\in\Pi_2\cap\mathcal{P}(\alpha^*)} \phi_k(\Omega_{\alpha^*}) e^x + \epsilon \sum_{k\in\Pi_1} e^{y_k} - \epsilon \sum_{k\in\Pi_2} e^{y_k}$$

feasible for X_{SDK}. It contradicts with the assumption that u^0 is an extreme point since $u^0 = (u^1 + u^2)/2$. □