

Two-stage stochastic lot-sizing problem under cost uncertainty

Zhili Zhou · Yongpei Guan

Published online: 27 February 2013
© Springer Science+Business Media New York 2013

Abstract For production planning problems, cost parameters can be uncertain due to marketing activities and interest rate fluctuation. In this paper, we consider a single-item two-stage stochastic lot-sizing problem under cost parameter uncertainty. Assuming cost parameters will increase or decrease after time period p each with certain probability, we minimize the total expected cost for a finite horizon problem. We develop an extended linear programming formulation in a higher dimensional space that can provide integral solutions by showing that its constraint matrix is totally unimodular. We also project this extended formulation to a lower dimensional space and obtain a corresponding extended formulation in the lower dimensional space. Final computational experiments demonstrate that the extended formulation is more efficient and performs more stable than the two-stage stochastic mixed-integer programming formulation.

Keywords Polyhedral combinatorics · Integer programming · Lot-sizing · Stochastic programming

1 Introduction

The traditional single-item uncapacitated lot-sizing problem (see, e.g., Wagner and Whitin 1958 and Nemhauser and Wolsey 1999) is to determine a production plan for a product to satisfy demands in a finite time horizon (e.g., T time periods) while minimizing total costs that include setup, production, and inventory holding costs. In certain practice, demands can maintain to be deterministic and cost parameters are uncertain. For instance, for several cases shown in Heikkilä (2002), manufactures obtain deterministic demand information when a good relationship is maintained with customers. In addition, long life circle

Z. Zhou
IBM Research Collaboratory, IBM, Singapore 486048, Singapore
e-mail: zhili@sg.ibm.com

Y. Guan (✉)
Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA
e-mail: guan@ise.ufl.edu

products and customized products provide deterministic demand information as well. For instance, in the chemical industry, a company produces mainly functional products, which are defined as ones that have long product life cycles and deterministic demands (see, e.g., Lee and Chen 2005). In the furniture industry, a portion of a company's products are built after the company gets orders from customers. Under these situations, the demand can be easily forecasted accurately. However, the cost parameter forecast may need to be adjusted monthly or quarterly. The cost parameter uncertainty can come from different ways. In the business environment, when manufacturers purchase materials and resources from upper-stream suppliers, the purchasing cost varies with the upper-stream marketing activities such as promotions (see, e.g., Narasimhan 1988 and Simester 1997). Because promotions are marketing tools of the upper-stream suppliers, they are uncertain and directly reflected as the production cost for the lower-stream manufacturers. When a promotion happens, the production cost for the lower-stream manufacturers will change accordingly. In addition, the interest rate and currency exchange rate fluctuations also increase or decrease the production costs of manufacturers, especially in the global business environment (see, e.g., Campa and Goldberg 2005).

Motivated by the above problem settings, in this paper we concentrate on studying production planning problems under cost parameter uncertainty, and in particular, analyzing the mathematical insights of a special case, e.g., a single-item problem. We study the single-item lot-sizing problem in which cost parameters after a given time period (e.g., time period p) will be uncertain and follow a discrete probability distribution with finite support. We formulate the problem as a two-stage multi-period stochastic integer program, which is an extension of the single-item deterministic uncapacitated lot-sizing problem. There has been extensive research on the deterministic lot-sizing problems. A survey of results and solution approaches are described in Pochet and Wolsey (2006). In the following literature review part, we focus on convex hull description and extended formulation studies on single-item lot-sizing problems for both deterministic and stochastic cases.

For the single-item deterministic uncapacitated lot-sizing problem, the convex hull description has been derived in Barany et al. (1984). In Pochet and Wolsey (1988), the first polyhedral study is performed for this problem with backlogging, and the convex hull description for this problem with backlogging is described in Küçükyavuz and Pochet (2007). In addition, Wagelmans et al. (1992) introduced the Wagner-Whitin costs, i.e., $\alpha_i + h'_i \geq \alpha_{i+1}$, $b'_i + \alpha_{i+1} \geq \alpha_i$ with $1 \leq i \leq T - 1$, where α_i , h'_i and b'_i are unit production, inventory, and backlogging costs for time period i , and implemented an $\mathcal{O}(T)$ time dynamic programming algorithm to solve the problem. Accordingly, Pochet and Wolsey (1994) developed an extended formulation for the single-item Wagner-Whitin cost case with $\mathcal{O}(T^2)$ constraints when backlogging is not considered, and with $\mathcal{O}(2^T)$ constraints plus an $\mathcal{O}(T^2 \log T)$ time separation algorithm when backlogging is considered.

For the single-item stochastic uncapacitated lot-sizing problem (SULS), most recent research works focus on developing efficient algorithms. The first polynomial time algorithm was studied in Guan and Miller (2008). Huang and Ahmed (2009) considered the case without setup cost and developed primal and dual algorithms for a multi-stage stochastic linear programming formulation for the problem. The computational complexity of the proposed algorithm is shown within $\mathcal{O}(N^2)$, where N is the number of nodes in the scenario tree. Huang and Küçükyavuz (2008) considered the variation of the problem with the consideration of random lead times, and developed a dynamic programming algorithm that runs in $\mathcal{O}(N^3)$ time. Recently, Jiang and Guan (2011) improves Huang and Küçükyavuz (2008)'s algorithm to be $\mathcal{O}(N^2)$ time. The reformulation for the problem was originally introduced in Ahmed et al. (2003). Later on, in Guan et al. (2006a), the extended (strong) formulation

approach was proved to be equivalent to adding the (ℓ, S) inequalities in the original formulation, in terms of providing LP lower bounds. However, both approaches could not provide integral solutions for SULS. Thus, efficient cutting planes (see, e.g., Guan et al. 2006b; Summa and Wolsey 2008) have been studied for its deterministic equivalent scenario tree based reformulation. An extended formulation that provides integral solutions for SULS up to now is only for two-period cases, which was developed in Guan et al. (2006a). Other related research works are referred to Feiringa and Sastri (1990), Ahmed and Garcia (2003), Terkaj and Tolio (2006), and Zanjania et al. (2010).

In this paper, we study the extended formulation of two-stage stochastic lot-sizing problem with backlogging and uncertain cost parameters. Compared with SULS without backlogging, SULS with backlogging is more complicated with more flexibility to satisfy demands. First, we develop an extended formulation in a higher dimensional space that can provide integral solutions by showing that its constraint matrix is totally unimodular. Second, we project this extended formulation to a lower dimensional space and obtain a corresponding extended formulation in the lower dimensional space. In other words, we provide an alternative linear formulation with more decision variables for the two-stage stochastic lot-sizing problem that can provide integral solutions. For the computational performance, we first report the value of the stochastic solution of our proposed two-stage multi-period stochastic integer program. Then, instead of solving a stochastic mixed-integer programming problem, the SULS with backlogging problem is directly solved as a linear program, which improves the computational efficiency and performs stably for different sizes of test instances.

This paper contributes to expanding the study of extended formulations for the deterministic lot-sizing problem to the two-stage stochastic programming setting. It provides one of the first studies on extended formulations for the stochastic lot-sizing problems. In addition, the SULS with backlogging is a subproblem of multi-item stochastic lot-sizing problems. The extended formulation of SULS with backlogging could help simplify the subproblem for multi-item stochastic lot-sizing problems and analyze their polyhedral structures. The extended formulation can also be embedded into the decomposition framework (e.g., the Lagrange relaxation decomposition) to speed up solving large scale production planning under uncertainty problems.

The remaining part of this paper is organized as follows: In Sect. 2, we define the Wagner-Whitin cost condition for the two-stage stochastic lot-sizing problem. We also discover the optimality condition and use it to generate an extended formulation in a higher dimensional space. We prove that the constraint matrix for the extended formulation is totally unimodular. In Sect. 3, we project the extended formulation back to a lower dimensional space such that we can find valid inequalities that can describe the convex hull of the problem in the lower dimensional space. In Sect. 4, we report the computational results and demonstrate the efficiency of the proposed extended formulation. Finally, in Sect. 5, we conclude our research.

2 Mathematical formulation

In this section, we first provide the list of notation in the following table. The detailed explanation will be provided when we introduce them.

As described in the introduction part, we assume the cost parameters for period 1 to a given period (defined as period p) are deterministic. After the given period (i.e., period p),

cost parameters are uncertain and follow a discrete probability distribution with finite support. The corresponding two-stage multi-period stochastic lot-sizing problem can be formulated as follows (cf. Birge and Louveaux 1997):

$$\begin{aligned}
 \min \quad & \sum_{i=1}^p (\alpha_i x_i + \beta'_i z_i + h'_i s_i + b'_i \ell_i) + E_{\xi} \mathcal{Q}(x, z, s, \ell, \xi(w)) \\
 \text{s.t.} \quad & x_i + s_{i-1} + \ell_i = d_i + s_i + \ell_{i-1}, \quad 1 \leq i \leq p \\
 & x_i \leq M z_i, \quad 1 \leq i \leq p \\
 & x_i, s_i, \ell_i \geq 0, \quad z_i \in \{0, 1\}, \quad 1 \leq i \leq p
 \end{aligned}$$

where

$$\mathcal{Q}(x, z, s, \ell, \xi(w)) = \min_{x^2, z^2, s^2, \ell^2} \left\{ \sum_{i=p+1}^T \begin{pmatrix} \alpha_i(w) x_i^2(w) \\ + \beta'_i(w) z_i^2(w) \\ + h'_i(w) s_i^2(w) \\ + b'_i(w) \ell_i^2(w) \end{pmatrix} \middle| \begin{array}{l} x_i^2(w) + s_{i-1}^2(w) + \ell_i^2(w) = d_i + s_i^2(w) + \ell_{i-1}^2(w) \\ x_i^2(w) \leq M_i z_i^2(w) \\ x_i^2(w), s_i^2(w), \ell_i^2(w) \geq 0, \quad z_i^2(w) \in \{0, 1\} \\ t(p) + 1 \leq t(i) \leq T \end{array} \right\}.$$

Note here as indicated in Table 1, $t(i)$ represents the time period of node i and $s_{i-1}^2(w) = s_p$ if $t(i-1) = t(p)$. Decision variables (z_i, x_i, s_i, ℓ_i) and $(z_i^2(w), x_i^2(w), s_i^2(w), \ell_i^2(w))$ represent the setup decision, production level, inventory level, and backlogging level on the first and second stages, respectively. The corresponding cost parameters are (β', α, h', b') and $(\beta'(w), \alpha(w), h'(w), b'(w))$. Parameter d_i represents the demand in time period i , which is deterministic.

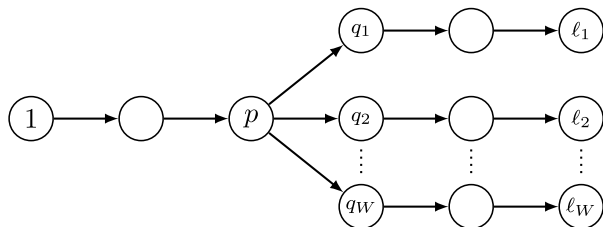
After exploring possible realizations of the second-stage random variables $(\beta'_i, \alpha_i, h'_i, b'_i)$, we can generate a two-stage stochastic scenario tree for the problem as shown in Fig. 1. Nodes $1, \dots, p$ are the first-stage nodes and nodes q_1, \dots, ℓ_w are the second-stage nodes, where the branching node p connects the first and the second stages and is unique on the scenario tree. Assuming there are W possible scenarios, as listed in Table 1, we let $\mathcal{P}_w, 1 \leq w \leq W$, represent the branch where scenario w happens and accordingly let $\rho_w, 1 \leq w \leq W$, represent the probability that scenario w will happen. Since there is no demand uncertainty, we have $d_i = d_{t(i)} \geq 0$ for the second-stage nodes as well. We let \mathcal{V} represent the set of nodes in the scenario tree, $\mathcal{V}(i)$ represent the set of nodes which are descendants of node i , $\mathcal{P}(i, j)$ represent the set of nodes on the path from node i to node j , and $\mathcal{P}_i^j(w)$ represent the set of nodes on the path from node i to node j , which is on \mathcal{P}_w and $t(j) = t$. For brevity, we let $\mathcal{P}(j)$ represent the set of nodes on the path from the root node to node j . We let $\mathcal{C}(i)$ represent the set of children of node i and \mathcal{L} represent the set of leaf nodes. Finally, we let $W(i)$ represent the set of scenarios that will occur after node i . For instance, if $i \leq p$, then $W(i) = \{1, \dots, W\}$; otherwise, if $t(i) \geq t(p) + 1$, then $W(i)$ contains a single element w such that $i \in \mathcal{P}_w$. The deterministic equivalent formulation of the two-stage stochastic lot-sizing under cost uncertainty problem, referred to as two-stage stochastic lot-sizing with backlogging (SULSB), can be described as follows:

$$\begin{aligned}
 \min \quad & \sum_{i=1}^p (\alpha_i x_i + \beta'_i z_i + h'_i s_i + b'_i \ell_i) + \sum_{w=1}^W \rho_w \left[\sum_{i \in \mathcal{P}_w, t(i) \geq t(p)+1} (\alpha_i x_i + \beta'_i z_i + h'_i s_i + b'_i \ell_i) \right] \\
 \text{s.t.} \quad & x_i + s_{a(i)} + \ell_i = d_i + s_i + \ell_{a(i)}, \quad i \in \mathcal{V}, \\
 \text{(SULSB)} \quad & x_i \leq M_i z_i, \quad i \in \mathcal{V}, \\
 & x_i, s_i, \ell_i \geq 0, \quad z_i \in \{0, 1\}, \quad i \in \mathcal{V},
 \end{aligned}$$

Table 1 Notation for two-stage SULS with backloging

Notation	Representation
\mathcal{V}	The set of nodes in the scenario tree
$\mathcal{V}(i)$	The set of descendants of node i
$\mathcal{C}(i)$	The set of children of node i
$\mathcal{P}(i, j)$	The set of nodes on the path from nodes i to j
$\mathcal{P}(j)$	The set of nodes on the path from the root node to node j
\mathcal{L}	The set of leaf nodes
W	The set of scenarios
$W(i)$	The set of scenarios that will occur after node i
\mathcal{P}_ω	The set of nodes on the branch corresponding to scenario ω
p_ω	The probability for the scenario ω to happen
$t(i)$	The time index of node i
p	The index for the branching node
$a(i)$	The parent node of node i
$(\beta'_i, \alpha_i, h'_i, b'_i)$	The setup and unit production, inventory, and backloging costs on the first stage
$(\beta'_i(\omega), \alpha_i(\omega), h'_i(\omega), b'_i(\omega))$	The setup and unit production, inventory, and backloging costs on the second stage
d_i	The demand in time period i
(z_i, x_i, s_i, ℓ_i)	The setup decision and production, inventory, and backloging levels on the first stage
$(z_i^2(\omega), x_i^2(\omega), s_i^2(\omega), \ell_i^2(\omega))$	The setup decision and production, inventory, and backloging levels on the second stage
f_i, g_i	The auxiliary decision variables to indicate whether node i is stocked and backloged
m_t^i, n_t^i	The auxiliary decision variables to indicate if inventory or backloging from node i covers up to time period t
$\psi(i)$	The time period of the earliest descendant of node i which is set up or backloged, i.e., $\psi(i) = \min\{t(j) : z_j = 1 \text{ or } g_j = 1, j \in \mathcal{V}(i) \setminus \{i\}\}$
$\eta(i, t)$	The ancestor node of node i at time t
$\xi(i, t, \omega)$	The descendant of node i at time period t and at the branch corresponding to scenario ω

Fig. 1 The scenario tree for the two-stage SULSB



where (x_i, s_i, ℓ_i, z_i) presents the decision variables for node i and, as indicated in Table 1, $a(i)$ is the parent node of node i . Without loss of generality, we assume $s_0 = 0, \ell_0 = 0$, and tighten $M_i = \sum_{t=t(i)}^T d_t$.

Now, we study the optimal solution forms of inventory and backlogging levels for the two-stage SULSB with Wagner-Whitin costs (defined as SULSB-WW) and generate a reformulation which can describe the integral polyhedron of the problem in a higher dimensional space.

First, we define Wagner-Whitin costs for two-stage SULSB. We substitute $x_i = d_i + s_i - \ell_i - s_{a(i)} + \ell_{a(i)}$ to eliminate decision variable x_i in the original formulation. Then, we get a reformulation of the problem in the (s, ℓ, z) space as follows:

$$\begin{aligned} & \sum_{i=1}^p (\alpha_i x_i + \beta'_i z_i + h'_i s_i + b'_i \ell_i) + \sum_{w=1}^W \rho_w \left[\sum_{i \in \mathcal{P}_w, t(i) \geq t(p)+1} (\alpha_i x_i + \beta'_i z_i + h'_i s_i + b'_i \ell_i) \right] \\ &= \sum_{i=1}^p [\alpha_i (d_i + s_i - \ell_i - s_{i-1} + \ell_{i-1}) + \beta'_i z_i + h'_i s_i + b'_i \ell_i] \\ & \quad + \sum_{w=1}^W \rho_w \left[\sum_{i \in \mathcal{P}_w, t(i) \geq t(p)+1} \alpha_i (d_i + s_i - \ell_i - s_{a(i)} + \ell_{a(i)}) + \beta'_i z_i + h'_i s_i + b'_i \ell_i \right] \\ &= \sum_{i=1}^{p-1} [(\alpha_i + h'_i - \alpha_{i+1}) s_i + (b'_i - \alpha_i + \alpha_{i+1}) \ell_i + \beta'_i z_i] + \sum_{i=1}^p \alpha_i d_i \\ & \quad + \left(\alpha_p + h'_p - \sum_{w=1}^W \rho_w \alpha_{q_w} \right) s_p + \left(b'_p - \alpha_p + \sum_{w=1}^W \rho_w \alpha_{q_w} \right) \ell_p + \beta'_p z_p \\ & \quad + \sum_{w=1}^W \sum_{j \in \mathcal{P}_w, t(j) \geq t(p)+1} \rho_w (h'_j + \alpha_j - \alpha_{C(j)}) s_j \\ & \quad + \sum_{w=1}^W \sum_{j \in \mathcal{P}_w, t(j) \geq t(p)+1} \rho_w (b'_j - \alpha_j + \alpha_{C(j)}) \ell_j \\ & \quad + \sum_{w=1}^W \sum_{j \in \mathcal{P}_w, t(j) \geq t(p)+1} \rho_w \beta'_j z_j + \sum_{w=1}^W \sum_{j \in \mathcal{P}_w, t(j) \geq t(p)+1} \rho_w \alpha_j d_j \\ &= \sum_{i \in \mathcal{V}} (h_i s_i + b_i \ell_i + \beta_i z_i) + (\text{a constant}) \end{aligned}$$

where

$$h_i = \begin{cases} \alpha_i + h'_i - \alpha_{i+1}, & 1 \leq t(i) \leq t(p) - 1 \\ \alpha_p + h'_p - \sum_{w=1}^W \rho_w \alpha_{q_w}, & t(i) = t(p), \\ \rho_w (h'_i + \alpha_i - \alpha_{C(i)}), & t(i) \geq t(p) + 1, \end{cases}$$

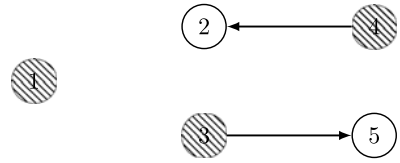
$$b_i = \begin{cases} b'_i - \alpha_i + \alpha_{i+1}, & 1 \leq t(i) \leq t(p) - 1, \\ b'_p - \alpha_p + \sum_{w=1}^W \rho_w \alpha_{q_w}, & t(i) = t(p), \\ \rho_w (b'_i - \alpha_i + \alpha_{C(i)}), & t(i) \geq t(p) + 1, \end{cases}$$

and $\beta_i = \beta'_i$ if $1 \leq t(i) \leq t(p)$ and $\beta_i = \rho_w \beta'_i$ otherwise. We let $\alpha_{C(i)} = 0$ if $i \in \mathcal{L}$.

Definition 1 A two-stage SULSB is said to have Wagner-Whitin costs if for all $i \in \mathcal{V}$,

$$h_i \geq 0, \quad b_i \geq 0, \quad \text{and} \quad \beta_i \geq \sum_{j \in \mathcal{C}(i)} \beta_j. \tag{1}$$

Fig. 2 The scenario tree for a three-period SULSB



For the deterministic version, the optimal solution forms of inventory and backlogging levels are studied by Pochet and Wolsey (1994). For two-stage SULSB, we consider the optimal solution forms of inventory level s_i and backlogging level ℓ_i for each $i \in \mathcal{V}$. In the optimal solution for SULSB-WW, the demand for each node i will be satisfied (1) by setting up the production at node i , (2) by inventory left from its parent node, or (3) by backlogging from its children. Before we describe the proposition, we show a three-period example to demonstrate the optimal solution form as shown in Fig. 2. In this example, if the unit inventory cost at node 1 and the setup costs at nodes 2 and 5 are very high, then it may happen in the optimal solution that productions are set up at nodes 1, 3, and 4. Demand at node 1 is satisfied by the production of itself. Demand at node 2 is satisfied by backlogging from node 4. Node 3 covers demands in nodes 3 and 5. Thus the second-stage nodes 2 and 3 are backlogged and set up respectively. Meanwhile, the first-stage node 1 does not cover any demand in the second stage.

To describe the property of the optimal solution and generate the extended formulation, besides listing the notation in Table 1, we provide the detailed explanation as follows:

- f_i : binary decision variable to indicate if node i is stocked (if yes, $f_i = 1$; otherwise, $f_i = 0$)
- g_i : binary decision variable to indicate if node i is backlogged (if yes, $g_i = 1$; otherwise, $g_i = 0$)
- m_t^i : binary decision variable to indicate if the inventory left from node i covers the demand at time period t , $t \geq t(i) + 1$ (if yes, $m_t^i = 1$; otherwise $m_t^i = 0$)
- n_t^i : binary decision variable to indicate if the backlogging at node i covers the demand at time period t , $t \leq t(i)$ (if yes, $n_t^i = 1$; otherwise $n_t^i = 0$)
- $\psi(i)$: the time period of the earliest descendant of node i which is set up or backlogged, i.e., $\psi(i) = \min\{t(j) : z_j = 1 \text{ or } g_j = 1, j \in \mathcal{V}(i) \setminus \{i\}\}$
- $\eta(i, t)$: the ancestor node of node i at time t , i.e., $\eta(i, t) = \{k \in \mathcal{P}(i) : t(k) = t\}$
- $\xi(i, t, w)$: a descendant of node i at time period t and at the branch corresponding to scenario w .

In addition, as shown in Fig. 3, we define $\Psi(i) = \{r \in \mathcal{V}(i), t(i) < t(r) < \psi(i)\}$, $\Lambda(i) = \bigcup_{r \in \mathcal{L} \cap \mathcal{V}(i)} \arg \min\{t(k) : k \in \mathcal{P}(i, r) \setminus \{i\} \text{ and } z_k = 1\}$, and $\Phi(i) = \bigcup_{j \in \Lambda(i)} \mathcal{P}(i, j) \setminus \{i, j\}$.

Proposition 1 For two-stage SULSB-WW, there exists an optimal inventory level of the form:

$$s_i = \sum_{t=t(i)+1}^{\psi(i)-1} d_t, \quad i \in \mathcal{V}, \tag{2}$$

and an optimal backlogging level of the form:

$$\ell_j = \max_{1 \leq \tau \leq t(j)} \sum_{t=\tau}^{t(j)} d_t \left[g_{\eta(j,t)} - \sum_{r \in \mathcal{P}(\eta(j,t),j)} z_r \right]^+, \quad j \in \Phi(i). \tag{3}$$

In addition, the optimal solutions of two-stage SULSB-WW satisfy the following conditions:

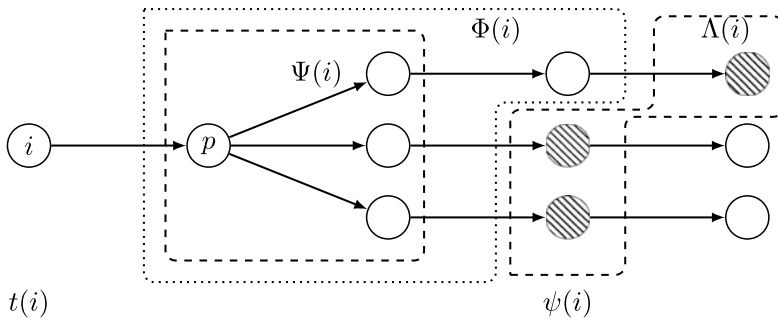


Fig. 3 The subtree of node i

$$f_i + g_i + z_i = 1, \quad i \in \mathcal{V} \tag{4}$$

$$g_i + z_i \geq g_{i-}, \quad i \in \mathcal{V} \setminus \{1\} \tag{5}$$

$$f_{i-} + z_{i-} \geq f_i, \quad i \in \mathcal{V} \setminus \{1\}. \tag{6}$$

Proof Let $\Lambda = \{i \in \mathcal{V} : z_i = 1\}$. First, for each $i \in \Lambda$, we prove that (2) and (3) hold by showing (2) and

$$\ell_j = \sum_{t=\psi(i)}^{t(j)} d_t \quad \text{for each } j \in \Phi(i) \setminus \Psi(i) \tag{7}$$

hold first. Then, we show (3) holds based on (7).

Proof of (2) and (7). We prove (2) and (7) by contradiction under three cases:

Case 1: (2) does not hold and (7) holds. In this case, the optimal inventory level s_i^* is either larger or less than $\sum_{t=t(i)+1}^{\psi(i)-1} d_t$.

Case 1.1. If $s_i^* < \sum_{t=t(i)+1}^{\psi(i)-1} d_t$, then there exists at least one node $j \in \Psi(i)$ for which $t(j) = \psi(i) - 1$ and whose demand is not satisfied. Thus, s_i^* is not a feasible solution, which is a contradiction.

Case 1.2. If $s_i^* > \sum_{t=t(i)+1}^{\psi(i)-1} d_t$, then let $\hat{s}_i = \sum_{t=t(i)+1}^{\psi(i)-1} d_t - \varepsilon$, where $0 < \varepsilon \leq s_i^* - \sum_{t=t(i)+1}^{\psi(i)-1} d_t$. It can be observed that (\hat{s}, ℓ^*) is also a feasible solution and leads to an equal total cost or a smaller total cost, which is a contradiction. For the former case, we can increase ε such that (2) holds.

Case 2: (2) holds and (7) does not hold. We can give the similar proof as in Case 1 to find a contradiction.

Case 3: Neither (2) nor (7) holds. In order to satisfy the demand in each node in $\Phi(i)$, we can construct two feasible solutions for two-stage SULSB-WW.

Let $s_i^1 = s_i^* + \varepsilon$, where ε is a small positive number, and each node j in $\Lambda(i)$ produces ε less. Then $\ell_j^1 = \ell_j^* - \varepsilon$, $j \in \Phi(i) \setminus \Psi(i)$. The corresponding objective value is

$$F^1 = \sum_{i \in \mathcal{V}} (h_i s_i + b_i \ell_i + \beta_i y_i) + \sum_{j \in \Psi(i)} h_j \varepsilon - \sum_{j \in \Phi(i) \setminus \Psi(i)} b_j \varepsilon.$$

Let $s_i^2 = s_i^* - \varepsilon$, and each node j in $\Lambda(i)$ produces ε more. Then $\ell_j^2 = \ell_j^* + \varepsilon$, $j \in \Phi(i) \setminus \Psi(i)$. The corresponding objective value is

$$F^2 = \sum_{i \in \mathcal{V}} (h_i s_i + b_i \ell_i + \beta_i y_i) - \sum_{j \in \Psi(i)} h_j \varepsilon + \sum_{j \in \Phi(i) \setminus \Psi(i)} b_j \varepsilon.$$

If $\sum_{j \in \Psi(i)} h_j < \sum_{j \in \Phi(i) \setminus \Psi(i)} b_j$, then $F^1 < F^*$; if $\sum_{j \in \Psi(i)} h_j > \sum_{j \in \Phi(i) \setminus \Psi(i)} b_j$, then $F^2 < F^*$. This contradicts the assumption that F^* is the optimal objective value. Note here, if $\sum_{j \in \Psi(i)} h_j = \sum_{j \in \Phi(i) \setminus \Psi(i)} b_j$, we can increase (or decrease) s^* and decrease (or increase) ℓ^* to match the optimal form. Therefore, (7) holds.

Proof of (3): According to (7), s_i covers demands for nodes in $\Psi(i)$. In order to minimize the objective function, nodes in $\Psi(i)$ do not obtain backlogging. Thus, $g_j = 0$, $j \in \Psi(i)$. Therefore, $\ell_j = \max_{\{\tau: 1 \leq \tau \leq t(j)\}} \sum_{t=\tau}^{t(j)} d_t [g_{\eta(j,t)} - \sum_{r \in \mathcal{P}(\eta(j,t),j)} z_r]^+ = \sum_{t=\psi(i)}^{t(j)} d_t$. Therefore, (2) and (3) hold.

Second, we prove that conditions (4), (5), and (6) hold.

According to the definition of f_i and g_i , $i \in \mathcal{V}$, it is obvious that conditions (5) and (6) hold.

Now we prove that condition (4) holds by two cases:

Case 1. If $j \in \Psi(i)$, then the demand in node j is satisfied by inventory and $f_j = 1$. In order to keep the smallest production cost, $f_j + g_j + z_j = 1$.

Case 2. If $j \in [\Phi(i) \setminus \Psi(i)] \cup \Lambda(i)$, then $f_j = 0$ and we need to prove $g_j + z_j = 1$. In order to satisfy the demand, $g_j + z_j \geq 1$.

Case 2.1. If $j \in \Lambda(i)$, $z_j = 1$. If $g_j = 1$, then we can let node j produce more to cover the backlogging amount and reduce the objective value, which is a contradiction. Therefore, $g_j = 0$ and $f_j + g_j + z_j = 1$.

Case 2.2. If $j \in \Phi(i) \setminus \Psi(i)$, $g_j = 1$. According to (3), the demand of node j can be covered by backlogging from its children. In order to minimize the objective function, $z_j = 0$.

Therefore, based on Cases 1 and 2, $f_j + g_j + z_j = 1$, $j \in \Phi(i)$. □

Under (4), at most one of $g_{\eta(i,k)}$ and $z_{\eta(i,k)}$ equals 1. Then $g_{\eta(i,k)} - z_{\eta(i,k)} = 1$ if $g_{\eta(i,k)} = 1$; $g_{\eta(i,k)} - z_{\eta(i,k)} \leq 0$ if $g_{\eta(i,k)} = 0$. Thus,

$$\begin{aligned} \ell_j &= \max_{1 \leq \tau \leq t(j)} \sum_{t=\tau}^{t(j)} d_t \left[g_{\eta(j,t)} - \sum_{r \in \mathcal{P}(\eta(j,t),j)} z_r \right]^+ \\ &= \max_{1 \leq \tau \leq t(j)} \sum_{t=\tau}^{t(j)} d_t \left[g_{\eta(j,t)} - z_{\eta(j,t)} - \sum_{r \in \mathcal{P}(\eta(j,t),j) \setminus \{\eta(j,t)\}} z_r \right]^+ \\ &= \max_{1 \leq \tau \leq t(j)} \sum_{t=\tau}^{t(j)} d_t \left[g_{\eta(j,t)} - \sum_{r \in \mathcal{P}(\eta(j,t),j) \setminus \{\eta(j,t)\}} z_r \right]^+. \end{aligned} \tag{8}$$

From (4), we also have

$$f_i + g_i + z_i = f_{a(i)} + g_{a(i)} + z_{a(i)} = 1. \tag{9}$$

Thus, one of constraints (5) and (6) is redundant. That is, if constraint (5) holds, then (6) must hold based on (9). Now, we introduce three types of inequalities corresponding to the optimal inventory level for node i on the scenario tree.

1. Path I inequality. This type of inequality is for the second-stage nodes. For a given node i on the second stage, w is determined and

$$m_t^i \geq f_{\xi(i,t,w)} - \sum_{j \in \mathcal{P}_t^i(w) \setminus \{i, \xi(i,t,w)\}} z_j, \quad t(i) \geq t(p) + 1, \quad t \geq t(i) + 1. \quad (10)$$

That is, node i covers demands along its branch until next backlogging or production.

2. Path II inequality. This type of inequality is for the first-stage nodes (except node p). For a given node i on the first stage, node $i + 1$ is its child node and

$$m_t^i \geq f_{i+1}, \quad 1 \leq t(i) \leq t(p) - 1, \quad t = t(i) + 1, \quad (11)$$

$$m_t^i \geq m_t^{i+1} - z_{i+1}, \quad 1 \leq t(i) \leq t(p) - 1, \quad t \geq t(i) + 2. \quad (12)$$

Inequality (11) indicates that if the demand of child node $i + 1$ is satisfied by inventory, then the inventory left from node i covers the demand of node $i + 1$. Inequality (12) indicates that if node $i + 1$ is not set up and the inventory left from node $i + 1$ covers the demand up to time t , $t \geq t(i) + 2$, then the inventory left from node i also covers the demand up to time t .

3. Connection inequality. This type of inequality is for the branching node p ,

$$m_t^p \geq f_{q_w}, \quad q_w \in \mathcal{C}(p), \quad t = t(p) + 1, \quad (13)$$

$$m_t^p \geq m_t^{q_w} - z_{q_w}, \quad q_w \in \mathcal{C}(p), \quad t \geq t(p) + 2. \quad (14)$$

Inequalities (13) and (14) are similar to (11) and (12). The inventory left from node p will cover demands from period $t(p) + 1$ to $t - 1$ unless there is a set up or a backlogging, before or at time period t along each scenario path.

With the information of m_t^i , the inventory level left from each node i in the scenario tree is as follows:

$$s_i = \sum_{t=t(i)+1}^T d_t m_t^i, \quad i \in \mathcal{V}. \quad (15)$$

Because for a given node i , its ancestor at time period t , $\eta(i, t)$, is unique, the following two inequalities hold for each node $i \in \mathcal{V}$ based on (8):

$$n_t^i \geq g_{\eta(i,t)} - \sum_{j \in \mathcal{P}(\eta(i,t), i) \setminus \{\eta(i,t)\}} z_j, \quad i \in \mathcal{V} \text{ and } t \leq t(i). \quad (16)$$

$$\ell_i = \sum_{t=1}^{t(i)} d_t n_t^i, \quad i \in \mathcal{V}. \quad (17)$$

Constraints (4), (5), (10)–(17) guarantee the feasibility of the reformulation for two-stage SULSB-WW since the demand for each time period is covered. Now, we show these constraints provide an extended formulation for two-stage SULSB-WW, which accordingly provides an integral solution for the problem.

Proposition 2 Constraints (4), (5), and (10) to (17) provide an extended formulation for two-stage SULSB-WW.

Proof Because constraints (15) and (17) can be directly transferred to the objective function, we prove this proposition by showing the constraint matrix described by constraints (4), (5), (10)–(14), and (16) is totally unimodular. The detailed proof is shown in Appendix. \square

3 Projection to a lower dimensional space

Now, we study the integral polyhedron in the (f, g, z, s, ℓ) space to generate an extended formulation for the two-stage SULSB-WW problem. First, we define

$$Q_M = \left\{ (f, g, z, s, \ell) : (f, g, z, s, \ell) \text{ satisfies} \right.$$

$$s_i \geq \sum_{t=t(i)+1}^{\tau} d_t \left(f_{\xi(i,t,w_t)} - \sum_{j \in \mathcal{P}_t^i(w_t) \setminus \{i, \xi(i,t,w_t)\}} z_j \right),$$

$$w_t \in W(i), t(i) + 1 \leq \tau \leq T, i \in \mathcal{V}, \tag{18}$$

$$\ell_i \geq \sum_{t=\tau}^{t(i)} d_t \left(g_{\eta(i,t)} - \sum_{j \in \mathcal{P}(\eta(i,t),i) \setminus \{\eta(i,t)\}} z_j \right), 1 \leq \tau \leq t(i), i \in \mathcal{V}, \tag{19}$$

$$\left. (4) \text{ and } (5), 0 \leq f_i, g_i, z_i \leq 1, s_i, \ell_i \geq 0, i \in \mathcal{V} \right\}.$$

We prove that Q_M is an integral polyhedron for the two-stage SULSB-WW problem in the (f, g, z, s, ℓ) space, by showing that it is a projection of Q_H into the (f, g, z, s, ℓ) space, where Q_H records the polyhedron of the two-stage SULSB-WW problem in the (f, g, z, m, n, s, ℓ) space, i.e.,

$$Q_H = \left\{ (f, g, z, m, n, s, \ell) : (f, g, z, m, n, s, \ell) \text{ satisfies} \right.$$

$$(4), (5), (10) \text{ to } (17),$$

$$\left. 0 \leq f_i, g_i, z_i, m_i^t, n_i^{t'} \leq 1, s_i, \ell_i \geq 0, t(i) + 1 \leq t \leq T, 1 \leq t' \leq t(i), i \in \mathcal{V} \right\}.$$

Proposition 3 $Proj_{(f,g,z,s,\ell)} Q_H = Q_M$.

Proof We prove this proposition by showing the following two claims:

Claim 1. All inequalities in Q_M are valid for Q_H .

Claim 2. For an arbitrary extreme point $(f, g, z, s, \ell) \in Q_M$, there exists a $(f, g, z, m, n, s, \ell) \in Q_H$.

Proof of Claim 1. We prove Claim 1 by showing that

$$s_i \geq \sum_{t=t(i)+1}^T d_t \left[f_{\xi(i,t,w_t)} - \sum_{j \in \mathcal{P}_t^i(w_t) \setminus \{i, \xi(i,t,w_t)\}} z_j \right]^+, \quad w_t \in W(i), \tag{20}$$

$$\ell_i \geq \sum_{t=1}^{t(i)} d_t \left[g_{\eta(i,t)} - \sum_{j \in \mathcal{P}(\eta(i,t),i) \setminus \{\eta(i,t)\}} z_j \right]^+ \tag{21}$$

are valid for Q_H , because if (20) and (21) hold, then

$$s_i \geq \sum_{t=t(i)+1}^T d_t \left[f_{\xi(i,t,w_t)} - \sum_{j \in \mathcal{P}_t^i(w_t) \setminus \{i, \xi(i,t,w_t)\}} z_j \right]^+$$

$$\geq \sum_{t=t(i)+1}^{\tau} d_t \left(f_{\xi(i,t,w_t)} - \sum_{j \in \mathcal{P}_t^i(w_t) \setminus \{i, \xi(i,t,w_t)\}} z_j \right), \quad t(i) + 1 \leq \tau \leq T;$$

$$\begin{aligned} \ell_i &\geq \sum_{t=1}^{t(i)} d_t \left[g_{\eta(i,t)} - \sum_{j \in \mathcal{P}(\eta(i,t),i) \setminus \{\eta(i,t)\}} z_j \right]^+ \\ &\geq \sum_{t=\tau}^{t(i)} d_t \left(g_{\eta(i,t)} - \sum_{j \in \mathcal{P}(\eta(i,t),i) \setminus \{\eta(i,t)\}} z_j \right), \quad 1 \leq \tau \leq t(i). \end{aligned}$$

Based on (16), (17) and the nonnegativity of n_t^i , we have

$$\begin{aligned} \ell_i &= \sum_{t=1}^{t(i)} d_t n_t^i \geq \sum_{t=1}^{t(i)} d_t \max \left\{ 0, g_{\eta(i,t)} - \sum_{j \in \mathcal{P}(\eta(i,t),i) \setminus \{\eta(i,t)\}} z_j \right\} \\ &= \sum_{t=1}^{t(i)} d_t \left[g_{\eta(i,t)} - \sum_{j \in \mathcal{P}(\eta(i,t),i) \setminus \{\eta(i,t)\}} z_j \right]^+. \end{aligned}$$

Thus, (21) holds.

Now, we prove (20) by two conditions. (1) $t(i) \geq t(p) + 1$; (2) $t(i) \leq t(p)$.

For condition (1), $t(i) \geq t(p) + 1$, based on (10), (15), and the nonnegativity of m_t^i , we have

$$\begin{aligned} s_i &= \sum_{t=t(i)+1}^T d_t m_t^i \geq \sum_{t=t(i)+1}^T d_t \max \left\{ 0, f_{\xi(i,t,w_t)} - \sum_{j \in \mathcal{P}_t^i(w_t) \setminus \{i, \xi(i,t,w_t)\}} z_j \right\} \\ &= \sum_{t=t(i)+1}^T d_t \left[f_{\xi(i,t,w_t)} - \sum_{j \in \mathcal{P}_t^i(w_t) \setminus \{i, \xi(i,t,w_t)\}} z_j \right]^+, \end{aligned}$$

where w_t is the single element in $W(i)$ for each $t(p) + 1 \leq t \leq T$. Thus, (20) holds.

For condition (2), $t(i) \leq t(p)$, we have

$$\begin{aligned} s_i &= \sum_{t=t(i)+1}^T d_t m_t^i = d_{t(i)+1} m_{t(i)+1}^i + \sum_{t=t(i)+2}^T d_t m_t^i \\ &\geq \begin{cases} d_{t(i)+1} [f_{i+1}]^+ + \sum_{t=t(i)+2}^T d_t [m_t^{i+1} - z_{i+1}]^+, & \text{if } t(i) < t(p), \end{cases} \quad (22) \\ &\geq \begin{cases} d_{t(i)+1} [f_{q_{w_t(i)+1}}]^{+} + \sum_{t=t(i)+2}^T d_t [m_t^{q_{w_t}} - z_{q_{w_t}}]^{+}, & \text{if } t(i) = t(p), \end{cases} \quad (23) \end{aligned}$$

where (22) is based on (11), (12), and the nonnegativity of m_t^i , and (23) is based on (13), (14), and the nonnegativity of m_t^i .

If $t(i) = t(p)$, as shown in (23), based on (10)

$$\begin{aligned} [m_t^{q_{w_t}} - z_{q_{w_t}}]^+ &\geq \left[f_{\xi(q_{w_t},t,w_t)} - \sum_{j \in \mathcal{P}_t^{q_{w_t}}(w_t) \setminus \{q_{w_t}, \xi(q_{w_t},t,w_t)\}} z_j - z_{q_{w_t}} \right]^+ \\ &= \left[f_{\xi(q_{w_t},t,w_t)} - \sum_{j \in \mathcal{P}_t^{q_{w_t}}(w_t) \setminus \{\xi(q_{w_t},t,w_t)\}} z_j \right]^+ \\ &= \left[f_{\xi(p,t,w_t)} - \sum_{j \in \mathcal{P}_t^p(w_t) \setminus \{p, \xi(p,t,w_t)\}} z_j \right]^+, \end{aligned}$$

where the second equation holds due to $\xi(q_{w_t}, t, w_t) = \xi(p, t, w_t)$, $\mathcal{P}_t^{q_{w_t}}(w_t) \subseteq \mathcal{P}_t^p(w_t)$ and $\mathcal{P}_t^{q_{w_t}}(w_t) \setminus \xi(q_{w_t}, t, w_t) = \mathcal{P}_t^p(w_t) \setminus \{p, \xi(q_{w_t}, t, w_t)\}$, because w_t is the single element in $W(p)$ for each $t(p) + 1 \leq t \leq T$. Thus, (20) holds.

Now, we only need to show that, if $t(i) < t(p)$,

$$[m_t^{i+1} - z_{i+1}]^+ \geq \left[f_{\xi(i,t,w_t)} - \sum_{j \in \mathcal{P}_t^i(w_t) \setminus \{i, \xi(i,t,w_t)\}} z_j \right]^+ \tag{24}$$

By observation, if $t \leq t(p)$, (24) holds based on (11) and (12). Hence, we discuss $t = t(p) + 1$ and $t > t(p) + 1$.

(a) If $t = t(p) + 1$, then

$$[m_t^{i+1} - z_{i+1}]^+ \geq \left[m_t^p - \sum_{j=i+1}^p z_j \right]^+ \geq \left[f_{q_{w_t}} - \sum_{j=i+1}^p z_j \right]^+ = \left[f_{q_{w_t}} - \sum_{j \in \mathcal{P}_t^i(w_t) \setminus \{i, q_{w_t}\}} z_j \right]^+,$$

where the first inequality follows (12) and the second inequality follows (13).

(b) If $t > t(p) + 1$, then

$$\begin{aligned} [m_t^{i+1} - z_{i+1}]^+ &\geq \left[m_t^p - \sum_{j=i+1}^p z_j \right]^+ \geq \left[m_t^{q_{w_t}} - \sum_{j \in \mathcal{P}_{t(p)+1}^i(w_t) \setminus \{i\}} z_j \right]^+ \\ &\geq \left[f_{\xi(i,t,w_t)} - \sum_{j \in \mathcal{P}_{t(p)+1}^i(w_t) \setminus \{i\}} z_j - \sum_{j \in \mathcal{P}_{t(p)+1}^{q_{w_t}}(w_t) \setminus \{q_{w_t}, \xi(i,t,w_t)\}} z_j \right]^+ \\ &\geq \left[f_{\xi(i,t,w_t)} - \sum_{j \in \mathcal{P}_t^i(w_t) \setminus \{i, \xi(i,t,w_t)\}} z_j \right]^+, \end{aligned}$$

where the first inequality follows (12), the second one follows (14), and the third one follows (10). Thus, (20) holds.

Therefore, Claim 1 holds.

Now, we prove Claim 2 holds.

For any give extreme point $(f, g, z, s, \ell) \in Q_M$, we construct m and n such that $(f, g, z, m, n, s, \ell) \in Q_H$. That is, (f, g, z, m, n, s, ℓ) satisfies the conditions (4), (5), (10) to (17). Now, for a given extreme point $(\widehat{f}, \widehat{g}, \widehat{z}, \widehat{s}, \widehat{\ell}) \in Q_M$, let

$$\widehat{m}_t^i = \max_{w \in W(i)} \left[\widehat{f}_{\xi(i,t,w)} - \sum_{j \in \mathcal{P}_t^i(w) \setminus \{i, \xi(i,t,w)\}} \widehat{z}_j \right]^+ \tag{25}$$

$$\text{and } \widehat{n}_t^i = \left[\widehat{g}_{\eta(i,t)} - \sum_{j \in \mathcal{P}(\eta(i,t), i) \setminus \{\eta(i,t)\}} \widehat{z}_j \right]^+ \tag{26}$$

Since $(\widehat{f}, \widehat{g}, \widehat{z}, \widehat{s}, \widehat{\ell})$ is an extreme point in Q_M , we first observe that $(\widehat{f}, \widehat{g}, \widehat{z}, \widehat{s}, \widehat{\ell})$ satisfies equation (15) based on (25) and (18). Similarly, $(\widehat{f}, \widehat{g}, \widehat{z}, \widehat{s}, \widehat{\ell})$ satisfies equation (17) based on (26) and (19). It is also obvious that $(\widehat{f}, \widehat{g}, \widehat{z}, \widehat{m}, \widehat{n}, \widehat{s}, \widehat{\ell})$ satisfies (4) and (5). Based on (25), inequalities (10), (11), and (13) hold. Based on (26) and the nonnegativity of \widehat{n}_t^i , inequality (16) holds.

Now, we only need to show that (12) and (14) hold.

For (12), let w^* be the scenario where \widehat{m}_t^{i+1} achieves the maximum value. Then, we observe that \widehat{m}_t^i achieves the maximum value in the same scenario w^* . We prove (12) holds

when $\widehat{z}_{i+1} = 1$ and $\widehat{z}_{i+1} = 0$. If $\widehat{z}_{i+1} = 1$, then $\widehat{m}_t^i = 0$ based on (25). Then, $\widehat{m}_t^i = 0 \geq \widehat{m}_t^{i+1} - 1 = \widehat{m}_t^{i+1} - \widehat{z}_{i+1}$. If $\widehat{z}_{i+1} = 0$, then

$$\begin{aligned} \widehat{m}_t^i &= \left[\widehat{f}_{\xi}(i,t,w^*) - \sum_{j \in \mathcal{P}_t^i(w^*) \setminus \{i,\eta(i,t,w^*)\}} \widehat{z}_j \right]^+ \\ &= \left[\widehat{f}_{\xi}(i,t,w^*) - \sum_{j \in \mathcal{P}_t^i(w^*) \setminus \{i,i+1,\eta(i,t,w^*)\}} \widehat{z}_j \right]^+ \\ &= \left[\widehat{f}_{\xi}(i,t,w^*) - \sum_{j \in \mathcal{P}_t^{i+1}(w^*) \setminus \{i+1,\eta(i,t,w^*)\}} \widehat{z}_j \right]^+ \\ &= \widehat{m}_t^{i+1} = \widehat{m}_t^{i+1} - \widehat{z}_{i+1}. \end{aligned}$$

Thus, (12) holds.

For (14), according to (25),

$$\begin{aligned} \widehat{m}_t^p &= \max_{w_t \in W(p)} \left[\widehat{f}_{\xi}(p,t,w_t) - \sum_{j \in \mathcal{P}_t^p(w_t) \setminus \{p,\xi(p,t,w_t)\}} \widehat{z}_j \right]^+ \\ &\geq \left[\widehat{f}_{\xi}(p,t,w_t) - \sum_{j \in \mathcal{P}_t^p(w_t) \setminus \{p,\xi(i,t,w_t)\}} \widehat{z}_j \right]^+, \quad \text{for each } w_t \in W(p). \end{aligned}$$

Now, for each particular $w_t \in W(p)$,

$$\begin{aligned} &\left[\widehat{f}_{\xi}(p,t,w_t) - \sum_{j \in \mathcal{P}_t^p(w_t) \setminus \{p,\xi(p,t,w_t)\}} \widehat{z}_j \right]^+ \\ &= \left[\widehat{f}_{\xi}(p,t,w_t) - \sum_{j \in \mathcal{P}_t^p(w_t) \setminus \{p,q_{w_t},\xi(p,t,w_t)\}} \widehat{z}_j - \widehat{z}_{q_{w_t}} \right]^+ \\ &= \left[\widehat{f}_{\xi}(q_{w_t},t,w_t) - \sum_{j \in \mathcal{P}_t^{q_{w_t}}(w_t) \setminus \{q_{w_t},\xi(p,t,w_t)\}} \widehat{z}_j - \widehat{z}_{q_{w_t}} \right]^+ \\ &= \left[\widehat{m}_t^{q_{w_t}} - \widehat{z}_{q_{w_t}} \right]^+ \\ &\geq \widehat{m}_t^{q_{w_t}} - \widehat{z}_{q_{w_t}}, \end{aligned}$$

where the second equation follows $\mathcal{P}_t^p(w_t) \setminus \{p, q_{w_t}\} = \mathcal{P}_t^{q_{w_t}}(w_t) \setminus \{q_{w_t}\}$ and the third equation follows $\widehat{m}_t^{q_{w_t}} = \widehat{f}_{\xi}(q_{w_t},t,w_t) - \sum_{j \in \mathcal{P}_t^{q_{w_t}}(w_t) \setminus \{q_{w_t},\xi(p,t,w_t)\}} \widehat{z}_j$ for a given $w_t \in W(p)$. Then, (14) holds. Thus, $(\widehat{f}, \widehat{g}, \widehat{z}, \widehat{m}, \widehat{n}, \widehat{s}, \widehat{\ell}) \in Q_H$ and Claim 2 holds.

Therefore, the conclusion holds. □

4 Computational experiments

In this section, we report the computational results to mainly demonstrate the computational efficiency and stability of the extended formulation for SULTS with backloging. All algorithms are implemented in C++ and carried out on a Linux workstation with two AMD Opteron Quad-Core 2.2 GHz processors and 32 GB RAM. We use CPLEX 12.4 Callable

Table 2 The percentage value of the stochastic solution for SULSB: $T = 50$

	Ratio 1			Ratio 2		
	0.2	0.4	0.6	0.2	0.4	0.6
$W = 100$	4.33 %	2.34 %	1.79 %	4.39 %	2.29 %	1.78 %
$W = 200$	2.34 %	1.24 %	0.91 %	2.22 %	1.28 %	0.95 %
$W = 300$	1.55 %	0.77 %	0.65 %	1.51 %	0.77 %	0.66 %

Library to implement our extended formulation and the two-stage mixed-integer programming formulation. To fairly compare the performance of these two formulations, we restrict the CPLEX Callable Library to run on a single thread.

The computational experiments are designed for two purposes, examining the value of stochastic solution of SULSB (VSS_{SULSB}) and demonstrating the efficiency and stability of the extended formulation of SULSB. A number of SULSB instances are generated corresponding to different scenario tree structures and different ratios of the setup cost to the unit production cost. We consider the underlying two-stage scenario tree is balanced with T time periods and W branches. We consider 27 different tree structures for both purposes. To test the extended formulation, we consider $T \in \{50, 100, 150\}$, $W \in \{10, 20, 30\}$, and $p \in \{0.2T, 0.4T, 0.6T\}$. To examine VSS_{SULSB} , we consider a larger number of scenarios with $W \in \{100, 200, 300\}$, the same time horizon $T \in \{50, 100, 150\}$, and the branching time period $p \in \{0.4T, 0.5T, 0.6T\}$. We consider the setup cost to production cost ratios to be 25 and 50, respectively. Therefore, there are 54 combinations in total for each case.

For each combination, we generate 5 random instances as follows. For each node i in the scenario tree, the unit production cost α_i is uniformly distributed in the interval $[10, 30]$ with average production cost $\bar{\alpha} = 20$ and the setup cost β'_i is uniformly distributed in the interval $[0.8(\beta'/\alpha)\bar{\alpha}, 1.2(\beta'/\alpha)\bar{\alpha}]$. We let the demand d_i , the unit inventory cost h'_i , and the unit backlogging cost b'_i uniformly distribute in the intervals $[50, 100]$, $[2, 7]$, and $[2, 5]$, respectively. Finally, all W branches are assigned equal probabilities, i.e., $\rho_w = 1/W$. For each combination, we run 5 instances and report the average value.

We first test the mixed-integer programming formulation of SULSB and its corresponding expected (mean) value problem, where the expected value problem (EV_{SULSB}) is obtained by substituting all random variables in SULSB by their expected values. Then, the value of using the expected value solution, e.g., EEV_{SULSB} , can be obtained by solving SULSB in which the first-stage variables are replaced by the corresponding solution of EV_{SULSB} . In Tables 2, 3, 4, we report the percentage value of the stochastic solution of SULSB, i.e., VSS_{SULSB}/EEV_{SULSB} to demonstrate the needs to solve SULSB, where $VSS_{SULSB} = EEV_{SULSB} - Obj_{SULSB}$ with Obj_{SULSB} representing the optimal objective value of SULSB. As described in Birge and Louveaux (1997), VSS measures how good (bad) a decision based on the EV problem solution compared with the decision based on stochastic programming. We let “Ratio 1” and “Ratio 2” indicate the cases in which the ratios between the setup cost and the unit production cost β'/α are 25 and 50, respectively.

The results in Tables 2 to 4 indicate that under the same setting in terms of time horizon, scenario number, and β'/α ratio, the percentage VSS_{SULSB} increases when the branching node time period, p , decreases. In average, the values of percentage VSS_{SULSB} are 2.5 %, 1.5 %, and 1.2 % corresponding to branching node time periods $0.2T$, $0.4T$, and $0.6T$, respectively. Thus, the computational results demonstrate that the cost of ignoring uncertainties is higher when cost uncertainties happen at an earlier time period in the planning horizon.

Table 3 The percentage value of the stochastic solution for SULSB: $T = 100$

	Ratio 1			Ratio 2		
	0.2	0.4	0.6	0.2	0.4	0.6
$W = 100$	3.78 %	2.24 %	1.78 %	4.12 %	2.37 %	1.95 %
$W = 200$	2.05 %	1.16 %	1.07 %	2.13 %	1.23 %	0.98 %
$W = 300$	1.39 %	0.95 %	0.72 %	1.41 %	0.83 %	0.66 %

Table 4 The percentage value of the stochastic solution for SULSB: $T = 150$

	Ratio 1			Ratio 2		
	0.2	0.4	0.6	0.2	0.4	0.6
$W = 100$	4.04 %	2.38 %	2.04 %	3.73 %	2.37 %	2.02 %
$W = 200$	1.95 %	1.18 %	1.05 %	2.11 %	1.22 %	0.97 %
$W = 300$	1.27 %	0.83 %	0.71 %	1.44 %	0.84 %	0.67 %

Table 5 Computational time (in seconds) comparison between stochastic programming formulation and extended formulation of SULSB: $T = 50$

		$p = 0.2T$			$p = 0.4T$			$p = 0.6T$		
		ELP	MIP	GAP	ELP	MIP	GAP	ELP	MIP	GAP
Ratio 1	$W = 10$	0.20	0.31	55.00 %	0.09	0.16	77.78 %	0.07	0.14	100.00 %
	$W = 20$	0.30	1.53	410.00 %	0.20	0.63	215.00 %	0.14	0.22	57.14 %
	$W = 30$	0.51	4.20	723.53 %	0.36	1.55	330.56 %	0.23	0.95	313.04 %
Ratio 2	$W = 10$	0.15	1.13	653.33 %	0.09	1.00	1011.11 %	0.07	0.35	400.00 %
	$W = 20$	0.31	4.27	1277.42 %	0.20	2.73	1265.00 %	0.13	1.46	1023.08 %
	$W = 30$	0.48	5.72	1091.67 %	0.34	5.09	1397.06 %	0.22	1.99	804.55 %

Second, we test both the mixed-integer programming and extended linear programming formulations for SULSB, and evaluate the performance of these two formulations by comparing their running times to obtain an optimal solution. The running times (in seconds) of these two formulations are reported in Tables 5, 6, 7. We let “MIP”, “ELP” and “GAP” denote the mixed-integer programming formulation, the extended linear programming formulation, and the computational time difference between the mixed-integer programming formulation and the extended linear programming formulation, i.e., $GAP = (Time_{MIP} - Time_{ELP})/Time_{ELP}$. We also let “Ratio 1” and “Ratio 2” indicate the cases in which the ratios between the setup cost and the unit production cost β'/α are 25 and 50, respectively.

Tables 5 to 7 demonstrate that all these two formulations obtain an optimal solution within 30 seconds for all tested instances. In average, the time spent for the mixed-integer programming approach is 2.77 times of that of the extended formulation approach. Meanwhile, the performance of the extended formulation is stable for both “Ratio 1” and “Ratio 2” cases. But the mixed-integer programming approach takes more times to run “Ratio 2” than “Ratio 1” instances in average.

Table 6 Computational time (in seconds) comparison between stochastic programming formulation and extended formulation of SULSB: $T = 100$

		$p = 0.2T$			$p = 0.4T$			$p = 0.6T$		
		ELP	MIP	GAP	ELP	MIP	GAP	ELP	MIP	GAP
Ratio 1	$W = 10$	1.04	1.80	73.08 %	0.69	1.49	115.94 %	0.32	0.99	209.38 %
	$W = 20$	2.17	4.96	128.57 %	1.16	2.98	156.90 %	0.69	1.91	176.81 %
	$W = 30$	3.38	5.64	66.86 %	1.95	4.26	118.46 %	1.07	2.86	167.29 %
Ratio 2	$W = 10$	1.03	3.37	227.18 %	0.68	2.47	263.24 %	0.35	1.91	445.71 %
	$W = 20$	2.14	8.42	293.46 %	1.29	9.24	616.28 %	0.74	3.30	345.95 %
	$W = 30$	3.31	13.94	321.15 %	1.91	8.22	330.37 %	1.16	5.01	331.90 %

Table 7 Computational time (in seconds) comparison between stochastic programming formulation and extended formulation of SULSB: $T = 150$

		$p = 0.2T$			$p = 0.4T$			$p = 0.6T$		
		ELP	MIP	GAP	ELP	MIP	GAP	ELP	MIP	GAP
Ratio 1	$W = 10$	2.94	3.73	26.87 %	1.65	3.51	112.73 %	1.04	2.36	126.92 %
	$W = 20$	5.71	7.52	31.70 %	3.48	4.68	34.48 %	2.12	3.93	85.38 %
	$W = 30$	9.16	10.68	16.49 %	5.56	7.45	33.99 %	3.49	6.04	73.07 %
Ratio 2	$W = 10$	3.08	13.27	330.84 %	1.76	4.28	143.18 %	1.08	3.59	232.41 %
	$W = 20$	5.87	20.77	253.83 %	3.60	11.27	213.06 %	2.14	7.10	231.78 %
	$W = 30$	9.29	29.01	212.27 %	5.70	13.92	144.21 %	3.42	8.52	149.12 %

The above computational results show that the extended formulation approach has a better chance to be more efficient and stable, as compared to the mixed-integer programming formulation approach.

5 Conclusion

In this research, we studied the two-stage stochastic lot-sizing problem under cost uncertainty and developed an extended linear programming formulation for the problem that can provide an integral optimal solution. In future research, a part of our results can be applied to a more general multi-stage stochastic programming setting. For instance, under the multi-stage setting, Proposition 1 still holds. The relationship between nodes in the stochastic scenario tree is similar to the relationship described in constraints (10) to (14). It is possible that similar reformulation can provide an extended formulation for the multi-stage stochastic lot-sizing problem under cost uncertainty in a higher dimensional space. In addition, we will target to solve large scale production planning problems under cost uncertainty, in which the extended formulation for each single-item problem can be embedded into a decomposition framework (e.g., the Lagrange relaxation decomposition). We will evaluate how this approach helps solve large scale practical production planning under uncertainty problems.

Acknowledgements The authors would like to thank the editor and the two anonymous referees for their helpful suggestions on improving the quality of this paper. This research was partially supported by the U.S. National Science Foundation under Award CMMI0700868 and under CAREER Award CMMI0748204.

Appendix: Proof of Proposition 2

To show that the constraint matrix described by constraints (4), (5), (10)–(14), and (16) is totally unimodular, we order variables $f_i, g_i,$ and z_i with loop i ranging from 1 to $|\mathcal{V}|$. Variable m_t^i is ordered with an outer loop i ranging from 1 to $|\mathcal{V}|$, and an inner loop t ranging from $t(i) + 1$ to T . Variable n_t^i is ordered with an outer loop i ranging from 1 to $|\mathcal{V}|$, and an inner loop t ranging from 1 to $t(i)$. Table 8 shows the constraint matrix corresponding to Fig. 2.

As the submatrix corresponding to variable n_t^i is an identity matrix for $i \in \mathcal{V}$ and $t \leq t(i)$, we do not need to consider n_t^i in our construction. As the submatrix corresponding to variable m_t^i is an identity matrix for $t(i) \geq t(p) + 2, i \in \mathcal{V},$ and $t \geq t(i) + 1$, we only need to consider $m_t^i, 1 \leq t(i) \leq t(p) + 1, t \geq t(i) + 1$ in our construction. In the following, we only study the constraint submatrix for the variables we need to consider, denoted as matrix A .

We show that for any column subset J of matrix A , there exist partitions J_1 and J_2 of J such that

$$\left| \sum_{j \in J_1} a_{ij} - \sum_{j \in J_2} a_{ij} \right| \leq 1 \tag{27}$$

for all i . We partition variables f, g, z, m in J starting from branching node p and then, extend it in both directions to nodes after p and before p .

First, we define $M(i)$ as the closest ancestor of node i such that $z_{M(i)} \in J$ and $m(i)$ as the closest descendant of node i such that $z_{m(i)} \in J$.

In the following Steps 1 to 5, we allocate the decision variables m and z to J_1 and J_2 :

- Step 1. Allocate m_t^p to J_1, z_p to $J_1,$ and z_{q_w} to $J_2,$ where $t \geq t(p) + 1, q_w \in \mathcal{C}(p)$.
- Step 2. Allocate $m_t^{q_w}, t \geq t(p) + 2$ to the same set as z_{q_w} (if $z_{q_w} \in J$), or to the same set as m_t^p (if $z_{q_w} \notin J$ and $m_t^p \in J$), or to J_1 (if $z_{q_w} \notin J$ and $m_t^p \notin J$).
- Step 3. Allocate $z_i, t(i) \geq t(p) + 2,$ to the opposite set of $z_{M(i)},$ if $M(i)$ exists and $t(M(i)) \geq t(p)$. Otherwise, allocate z_i to the opposite set of m_t^p if $m_t^p \in J$. If $m_t^p \notin J$, then allocate z_i to J_2 .
- Step 4. Allocate $z_i, 1 \leq i \leq p - 1,$ to the opposite set of $z_{m(i)},$ if $m(i)$ exists and $t(m(i)) \leq t(p)$. Otherwise, allocate z_i to J_1 .
- Step 5. Allocate $m_t^i, 1 \leq i \leq p - 1,$ to the opposite set of $z_{m(i)},$ if $m(i)$ exists and $t(m(i)) \leq t(p)$. Otherwise, allocate m_t^i to J_1 .

In the following Steps 6 and 7, we allocate the decision variables f and g to J_1 and J_2 :

- Step 6. Allocate f_i to the same set of $z_{M(i)}$ if $M(i)$ exists; allocate f_i to the opposite set of z_i if $M(i)$ does not exist and $z_i \in J$; allocate f_i to the opposite set of $z_{m(i)}$ if $m(i)$ exists, $M(i)$ does not exist, and $z_i \notin J$; allocate f_i to J_1 if $m(i)$ and $M(i)$ do not exist and $z_i \notin J$.
- Step 7. Allocate g_i to the same set of $z_{m(i)}$ if $m(i)$ exists; allocate g_i to the opposite set of z_i if $m(i)$ does not exist and $z_i \in J$; allocate g_i to the opposite set of $z_{M(i)}$ if $m(i)$ does not exist, $M(i)$ exists, and $z_i \notin J$; allocate g_i to J_2 if $m(i)$ and $M(i)$ do not exist and $z_i \notin J$.

Following the above partition steps, we observe the following two properties:

Claim 1 If $z_i \in J$ and $M(i)$ exists, z_i goes to the opposite set of $z_{M(i)}$ for all $i \in \mathcal{V} \setminus \{1\}$.

Proof of Claim 1 If $1 \leq t(M(i)) \leq t(i) \leq t(p),$ because the closest descendant of $M(i)$ is i and $t(i) \leq t(p), m(M(i)) = i, z_{M(i)}$ goes to the opposite set of z_i based on Step 4.

Table 8 The matrix of constraints (4), (5), (10) to (14), and (16) for the example in Fig. 2

	z_1	z_2	z_3	z_4	z_5	f_1	f_2	f_3	f_4	f_5	g_1	g_2	g_3	g_4	g_5	m_2^1	m_3^1	m_3^2	m_3^3	n_1^2	n_1^3	n_2^4	n_1^4	n_2^5	n_1^5	
1	1					1																				
2		1					1					1														
3			1					1					1													
4				1					1					1												
5					1					1					1											
6		1									-1	1														
7			1									-1	1													
8				1										1												
9					1										1											
10										-1						1										
11																	1									
12																		1								
13																										
14																										
15																										
16																										
17																										
18																										
19																										
20																										
21																										

If $1 \leq t(M(i)) < t(p) < t(i) \leq T$, $z_{M(i)}$ goes to J_1 based on Step 4 and z_i goes to J_2 based on Step 3 (if $t(i) \geq t(p) + 2$) or based on Step 1 (if $t(i) = t(p) + 1$).

If $t(p) = t(M(i)) < t(i) \leq T$, $z_{M(i)}$ goes to J_1 based on Step 1 (i.e., $M(i) = p$), z_i goes to J_2 based on Step 1 (if $i = q_w$) or Step 3 (if $t(i) \geq t(p) + 2$). Thus, z_i goes to the opposite set of $z_{M(i)}$.

If $t(p) + 1 \leq t(M(i)) < t(i) \leq T$, z_i goes to the opposite set of $z_{M(i)}$ based on Step 3.

Therefore, Claim 1 holds. □

Claim 2 If $j(w_{k_1})$ and $j(w_{k_2})$ are the first second-stage nodes corresponding to scenarios w_{k_1} and w_{k_2} such that $z_{j(w_{k_1})}, z_{j(w_{k_2})} \in J$, then $z_{j(w_{k_1})}$ and $z_{j(w_{k_2})}$ go to the same set.

Proof of Claim 2 For any first second-stage node $j(w)$, if $j(w) = q_w$, then $z_{j(w)}$ goes to J_2 based on Step 1, otherwise $j(w)$ goes to J_2 based on Step 3 due to the fact that $j(w)$ is the first second-stage node in J and on the branch corresponding to w and $t(p) \leq t(j(w)) \leq T$. Thus, the claim holds. □

Now, we verify that (27) holds for constraints (4), (5), (10)–(14), and (16) under the above partition. First, corresponding to each row, if J contains at most one decision variable in A , then it is obvious that (27) holds. In the following, we consider the case in which J contains at least two decision variables in each row of A .

1. For constraint (4), we discuss the following four cases:

1.1 $\{f_i, g_i\} \subseteq J$ and $z_i \notin J$

1.1.1 If both $M(i)$ and $m(i)$ exist, f_i goes to the same set of $z_{M(i)}$ based on Step 6; g_i goes to the same set of $z_{m(i)}$ based on Step 7. Because $M(m(i)) = M(i)$ (due to $z_i \notin J$), $z_{m(i)}$ goes to the opposite set of $z_{M(i)}$ based on Claim 1. Thus, f_i goes to the opposite set of g_i . Then, (27) holds.

1.1.2 If $M(i)$ exists and $m(i)$ does not exist, f_i goes to the same set of $z_{M(i)}$ based on Step 6; g_i goes to the opposite set of $z_{M(i)}$ based on Step 7. Thus, f_i goes to the opposite set of g_i .

1.1.3 If $m(i)$ exists and $M(i)$ does not exist, g_i goes to the same set of $z_{m(i)}$ based on Step 7; f_i goes to the opposite set of $z_{m(i)}$ based on Step 6. Thus, f_i goes to the opposite set of g_i .

1.1.4 If neither $m(i)$ nor $M(i)$ exists, f_i and g_i go to J_1 and J_2 respectively based on Steps 6 and 7.

1.2 $\{g_i, z_i\} \subseteq J$ and $f_i \notin J$

1.2.1 If $m(i)$ exists, then g_i goes to the same set of $z_{m(i)}$ based on Step 7. Because $M(m(i)) = i$, $z_{m(i)}$ goes to the opposite set of z_i based on Claim 1. Thus, g_i goes to the opposite set of z_i . Then, (27) holds.

1.2.2 If $m(i)$ does not exist, g_i goes to the opposite set of z_i based on Step 7. Thus, (27) holds.

1.3 $\{f_i, z_i\} \in J$ and $g_i \notin J$

1.3.1 If $M(i)$ exists, f_i goes to the same set of $z_{M(i)}$ based on Step 6; z_i goes to the opposite set of $z_{M(i)}$ based on Claim 1. Then, f_i goes to the opposite set of z_i . Thus, (27) holds.

1.3.2 If $M(i)$ does not exist, f_i goes to the opposite set of z_i based on Step 6. Thus, (27) holds.

1.4 $\{f_i, g_i, z_i\} \in J$. This conclusion directly follows 1.3, because f_i goes to the opposite set of z_i . Then, (27) holds no matter where g_i goes.

2. For constraint (5), we discuss the following four cases:

2.1 If $\{g_i, z_i\} \in J$ and $g_{a(i)} \notin J$ this condition is the same as 1.2. Thus, (27) holds.

2.2 If $\{g_{a(i)}, z_i\} \in J$ and $g_i \notin J$, $g_{a(i)}$ goes to the same set as z_i due to $z_i = z_{m(a(i))}$ based on Step 7. Thus, (27) holds.

2.3 If $\{g_i, g_{a(i)}\} \in J$ and $z_i \notin J$, we first have $m(i) = m(a(i))$. If (a) $m(i) = m(a(i))$ exists, then g_i and $g_{a(i)}$ go to the same set as $z_{m(i)}$ based on Step 7. If (b) $m(i) = m(a(i))$ does not exist and $z_{a(i)} \in J$, then $g_{a(i)}$ goes to the opposite set of $z_{a(i)}$ based on Step 7. Also, because $z_{a(i)} \in J$, we have $z_{M(i)} = z_{a(i)}$. Then, g_i goes to the opposite set of $z_{a(i)}$ based on Step 7. Thus, g_i and $g_{a(i)}$ go to the same set. If (c) $m(i) = m(a(i))$ does not exist, $z_{a(i)} \notin J$ and $M(a(i))$ exists, then $z_{M(i)} = z_{M(a(i))}$. Because $z_i \notin J$, g_i and $g_{a(i)}$ go to the opposite set of $z_{M(i)} = z_{M(a(i))}$ based on Step 7. Thus, g_i and $g_{a(i)}$ go to the same set. If (d) $m(i) = m(a(i))$, $M(a(i))$ does not exist, $z_{a(i)} \notin J$, then both g_i and $g_{a(i)}$ go to J_2 based on Step 7.

2.4 If $\{g_i, g_{a(i)}, z_i\} \in J$, this conclusion directly follows 2.2, because $g_{a(i)}$ and z_i go to the same set. Then, (27) holds no matter where g_i goes.

3. For constraint (10), we consider two cases:

3.1 $t(i) \geq t(p) + 2$. For this case, we do not need to consider m_i^i based on the identity matrix argument at the beginning of the proof. Then, based on Step 6, $f_{\xi(i,t,w)}$ goes to the same set of $z_{M(\xi(i,t,w))}$, where $z_{M(\xi(i,t,w))} \in J$ and the corresponding $M(\xi(i,t,w))$ is the largest-index such node in path $\mathcal{P}_t^i(w)$ with $\mathcal{P}_t^i(w)$ representing the set of nodes on the path from node i to a node at time period t and on the branch corresponding to scenario w . Note that there must exist at least one such $M(\xi(i,t,w))$ based on the assumption that we have at least two elements in J for each constraint. Based on Step 4, z_j alternatively goes to J_1 and J_2 based on Step 4, where $t(p) + 2 \leq t(j) \leq t$. Thus, (27) holds.

3.2 $t(i) = t(p) + 1$. For this case, $i = q_w$, for some $1 \leq w \leq W$. We discuss the following two cases:

3.2.1 If $m_t^{q_w} \notin J$, this argument is the same as 3.1. Thus, (27) holds.

3.2.2 If $m_t^{q_w} \in J$, under this condition, we discuss $f_{\xi(q_w,t,w)} \notin J$ and $f_{\xi(q_w,t,w)} \in J$ respectively.

3.2.2.1 $f_{\xi(q_w,t,w)} \notin J$. Under this case, if $z_{q_w} \in J$, $m_t^{q_w}$ and $z_{m(q_w)}$ go to the same set and the opposite set of z_{q_w} based on Steps 2 and 3 respectively; if $z_{q_w} \notin J$, $m_t^{q_w}$ goes to the same set as m_t^p (i.e., J_1) if $m_t^p \in J$ or J_1 if $m_t^p \notin J$ based on Step 2 and similarly $z_{m(q_w)}$ goes to J_2 based on Step 3. Thus, $m_t^{q_w}$ and $z_{m(q_w)}$ go to the opposite sets. Besides these, $z_j \in J$ alternatively goes to J_1 and J_2 where $t(p) + 2 \leq t(j) \leq t$ based on Step 3. Thus, (27) holds.

3.2.2.2 $f_{\xi(q_w,t,w)} \in J$. Under this case, we discuss two cases depending on if there exists a node $j \in \mathcal{P}_t^i(w) \setminus \{i, \xi(i,t,w)\}$ such that $z_j \in J$.

3.2.2.2.1 If no such node j exists, then $\{f_{\xi(q_w,t,w)}, m_t^{q_w}\} \in J$, based on our assumption that at least two elements in each constraint in matrix A . If $z_{q_w} \in J$, $M(\xi(q_w,t,w)) = q_w$ due to $z_j \notin J$ for each $j \in \mathcal{P}_t^i(w) \setminus \{i, \xi(i,t,w)\}$. Based on Steps 6 and 2, $f_{\xi(q_w,t,w)}$ and $m_t^{q_w}$ go to the same set of z_{q_w} . If $z_{q_w} \notin J$, based on Step 2, $m_t^{q_w}$ goes to J_1 . In the following, we prove that $f_{\xi(q_w,t,w)}$ goes to J_1 in this case. Based on Step 6, $f_{\xi(q_w,t,w)}$ goes to (a) the same set of $z_{M(\xi(q_w,t,w))}$ if $M(\xi(q_w,t,w))$ exists, (b) the opposite set of $z_{\xi(q_w,t,w)}$ if $M(\xi(q_w,t,w))$ does not exist and $z_{\xi(q_w,t,w)} \in J$, (c) the opposite set of $z_{m(\xi(q_w,t,w))}$ if $m(\xi(q_w,t,w))$ exists, $M(\xi(q_w,t,w))$

does not exist, and $z_{\xi(q_w,t,w)} \notin J$, or (d) J_1 if $M(\xi(q_w, t, w))$ and $m(\xi(q_w, t, w))$ do not exist, and $z_{\xi(q_w,t,w)} \notin J$. For (a), $M(\xi(q_w, t, w))$ exists and $z_{q_w} \notin J$. Based on Step 4, $z_{M(\xi(q_w,t,w))}$ goes to J_1 , because $1 \leq t(M(\xi(q_w, t, w))) \leq t(p)$. For (b), $z_{\xi(q_w,t,w)} \in J$, $z_{q_w} \notin J$, and $M(\xi(q_w, t, w))$ does not exist. Then, $z_{\xi(q_w,t,w)}$ goes to J_2 based on Step 3 and accordingly $f_{\xi(q_w,t,w)}$ goes to J_1 . For (c), $z_{m(\xi(q_w,t,w))} \in J$, and $z_{\xi(q_w,t,w)}, z_{M(\xi(q_w,t,w))}, z_{q_w} \notin J$. Then, $z_{m(\xi(q_w,t,w))}$ goes to J_2 based on Step 3. Thus, accordingly $f_{\xi(q_w,t,w)}$ goes to J_1 . Then, $f_{\xi(q_w,t,w)}$ and $m_t^{q_w}$ go to the same set.

3.2.2.2.2 If there exists such a node j , then $f_{\xi(q_w,t,w)}$ goes to the same set of $z_{M(\xi(q_w,t,w))}$ based on Step 6. If $z_{q_w} \in J$, $m_t^{q_w}$ goes to the same set of z_{q_w} based on Step 2, and $z_{m(q_w)}$ goes to the opposite set of z_{q_w} based on Step 3. If $z_{q_w} \notin J$, $m_t^{q_w}$ goes to J_1 based on Step 2 and $z_{m(q_w)}$ goes to J_2 based on Step 3. Thus, for both $z_{q_w} \in J$ and $z_{q_w} \notin J$, $m_t^{q_w}$ goes to the opposite set of $z_{m(q_w)}$. Besides these, z_j alternatively goes to J_1 and J_2 , $j \in \mathcal{P}_t^i(w) \setminus \{i, \xi(i, t, w)\}$. Thus, (27) holds.

4. For constraint (11), we only need to consider the case in which $\{m_t^i, f_{i+1}\} \in J$.
 - 4.1 If $M(i + 1)$ exists, f_{i+1} goes to the same set as $z_{M(i+1)}$ based on Step 6. If $m(i)$ exists and $t(m(i)) \leq t(p)$, m_t^i goes to the opposite set of $z_{m(i)}$ based on Step 5. Because $m(i) = m(M(i + 1))$, $z_{M(i+1)}$ and $z_{m(i)}$ go to the opposite set based on Step 4. Thus, f_{i+1} and m_t^i go to the same set. If $m(i)$ does not exist, or $m(i)$ exists and $t(m(i)) > t(p)$, then $m(M(i + 1))$, $t(m(M(i + 1))) \leq t(p)$, does not exist. Thus, $z_{M(i+1)}$ and m_t^i go to J_1 based on Steps 4 and 5. Thus, m_t^i and f_{i+1} go to the same set. Therefore, (27) holds.
 - 4.2 If $M(i + 1)$ does not exist and $z_{i+1} \in J$, then $m(i) = i + 1$. Thus, m_t^i and f_{i+1} go to the opposite set of z_{i+1} based on Steps 5 and 6. Thus, (27) holds.
 - 4.3 If $M(i + 1)$ does not exist, $z_{i+1} \notin J$, and $m(i + 1)$ exists, then, $m(i) = m(i + 1)$. If $1 \leq t(m(i + 1)) \leq t(p)$, m_t^i and f_{i+1} go to the opposite set of $z_{m(i+1)}$ based on Steps 5 and 6; otherwise, if $t(m(i + 1)) > t(p)$, m_t^i goes to J_1 and f_{i+1} goes to the opposite set of $z_{m(i+1)}$. Since $z_{m(i+1)}$ goes to J_2 based on Step 1 (if $t(m(i + 1)) = t(p) + 1$) or Step 3 (under this case, $M(m(i + 1))$ does not exist), f_{i+1} and m_t^i go to the same set. Then, (27) holds.
 - 4.4 If neither $M(i + 1)$ nor $m(i + 1)$ exists and $z_{i+1} \notin J$, then m_t^i and f_{i+1} go to J_1 based on Steps 5 and 6. Then, (27) holds.
5. For constraint (12), we discuss the following four cases:
 - 5.1 $\{m_t^i, z_{i+1}\} \subseteq J$, and $m_t^{i+1} \notin J$. Based on Step 5, m_t^i goes to the opposite set of z_{i+1} since z_{i+1} is the closest descendant of node i . Thus, (27) holds.
 - 5.2 $\{m_t^{i+1}, z_{i+1}\} \subseteq J$, and $m_t^i \notin J$. Under this case, m_t^{i+1} and z_{i+1} will go to the same set, because both z_{i+1} and m_t^{i+1} are in the opposite set of $z_{m(i+1)}$, if $m(i + 1)$ exists and $t(m(i + 1)) \leq t(p)$ or in J_1 otherwise, based on Steps 4 and 5.
 - 5.3 $\{m_t^i, m_t^{i+1}\} \subseteq J$, and $z_{i+1} \notin J$. Because $z_{i+1} \notin J$, m_t^i and m_t^{i+1} are in the same set based on Step 5.
 - 5.4 $\{m_t^i, m_t^{i+1}, z_{i+1}\} \subseteq J$. The conclusion follows from Step 5. Variable m_t^i goes to the opposite set of z_{i+1} , since z_{i+1} is the closest descendant of node i . Then, (27) holds no matter where m_t^{i+1} goes.
6. For constraint (13), we only need to consider the case in which $\{f_{q_w}, m_t^p\} \subseteq J$. Based on Step 1, m_t^p goes to J_1 . In the following, we show that f_{q_w} goes to J_1 .
 - 6.1 If $M(q_w)$ exists, then $z_{M(q_w)}$ goes to J_1 based on Step 1 if $M(q_w) = p$, or Step 4 if $t(m(M(q_w))) \leq t(p) - 1$. Thus, based on Step 6, f_{q_w} goes to the same set of $z_{M(q_w)}$, which is J_1 .

- 6.2 If $z_{q_w} \in J$, $M(q_w)$ does not exist, then z_{q_w} goes to J_2 based on Step 1. Based on Step 6, f_{q_w} goes to the opposite set of z_{q_w} , which is J_1 .
- 6.3 If $m(q_w)$ exists, $z_{q_w} \notin J$ and $M(q_w)$ does not exist, then $t(m(q_w)) \geq t(p) + 2$ and $z_{m(q_w)}$ goes to the opposite set of m_i^p based on Step 3. Based on Step 6, we have f_{q_w} goes to the opposite set of $z_{m(q_w)}$. Thus, f_{q_w} and m_i^p go to the same set, which is J_1 .
- 6.4 If $m(q_w)$ and $M(q_w)$ do not exist and $z_{q_w} \notin J$, then f_{q_w} goes to J_1 based on Step 6.
7. For constraint (14), we need to consider the following four cases:
- 7.1 $\{m_i^p, z_{q_w}\} \subseteq J$ and $m_i^{q_w} \notin J$. Based on Step 1, m_i^p and z_{q_w} go to J_1 and J_2 , respectively. Thus, (27) holds.
- 7.2 $\{m_i^{q_w}, z_{q_w}\} \subseteq J$ and $m_i^p \notin J$. $m_i^{q_w}$ and z_{q_w} go to the same set based on Step 2.
- 7.3 $\{m_i^p, m_i^{q_w}\} \subseteq J$ and $z_{q_w} \notin J$. Under this case, m_i^p and $m_i^{q_w}$ go to the same set based on Step 2.
- 7.4 $\{m_i^p, m_i^{q_w}, z_{q_w}\} \subseteq J$. Under this case, m_i^p and z_{q_w} go to J_1 and J_2 , respectively based on Step 1. Then, (27) holds no matter where $m_i^{q_w}$ goes.
8. For constraint (16), we do not need to consider n_i^t based on the identity matrix argument at the beginning of the proof. Based on Step 7, $g_{\eta(i,t)}$ goes to the same set of $z_{m(\eta(i,t))}$. Note here, there must exist $m(\eta(i,t))$ based on the assumption that we have at least two elements in J . If $t(m(\eta(i,t))) \leq t(p)$ or $t(\eta(i,t)) \geq t(p) + 1$, then $\eta(i,t)$ and $m(\eta(i,t))$ are one-to-one corresponding. Otherwise, all $z_{m(\eta(i,t))}$ in different scenarios go to the same set based on Claim 2 because $t(m(\eta(i,t))) \geq t(p) + 1$. Besides these, z_j alternatively goes to J_1 and J_2 based on Claim 1. Thus, (27) holds.

Therefore, the desired property (27) holds for constraints (4), (5), (10)–(14), and (16), and the corresponding constraint matrix is totally unimodular.

References

- Ahmed, S., & Garcia, R. (2003). Dynamic capacity acquisition and assignment under uncertainty. *Annals of Operations Research*, 124, 267–283.
- Ahmed, S., King, A. J., & Parija, G. (2003). A multi-stage stochastic integer programming approach for capacity expansion under uncertainty. *Journal of Global Optimization*, 26, 3–24.
- Barany, I., Van Roy, T. J., & Wolsey, L. A. (1984). Uncapacitated lot-sizing: the convex hull of solutions. *Mathematical Programming Studies*, 22, 32–43.
- Birge, J., & Louveaux, F. (1997). *Introduction to stochastic programming*. Berlin: Springer.
- Campa, J., & Goldberg, L. (2005). Exchange rate pass through into import prices. *Review of Economics and Statistics*, 87, 679–690.
- Di Summa, M., & Wolsey, L. A. (2008). Lot-sizing on a tree. *Operations Research Letters*, 36, 7–13.
- Feiringa, B. R., & Sastri, T. (1990). Improving production planning by utilizing stochastic programming. *Computers & Industrial Engineering*, 19, 53–56.
- Guan, Y., & Miller, A. J. (2008). Polynomial-time algorithms for stochastic uncapacitated lot-sizing problems. *Operations Research*, 56, 1172–1183.
- Guan, Y., Ahmed, S., Miller, A. J., & Nemhauser, G. L. (2006a). On formulations of the stochastic uncapacitated lot-sizing problem. *Operations Research Letters*, 34, 241–250.
- Guan, Y., Ahmed, S., Nemhauser, G. L., & Miller, A. J. (2006b). A branch-and-cut algorithm for the stochastic uncapacitated lot-sizing problem. *Mathematical Programming*, 105, 55–84.
- Heikkilä, J. (2002). From supply to demand chain management: efficiency and customer satisfaction. *Journal of Operations Management*, 20, 747–767.
- Huang, K., & Ahmed, S. (2009). The value of multistage stochastic programming in capacity planning under uncertainty. *Operations Research*, 57, 893–904.
- Huang, K., & Küçükyavuz, S. (2008). On stochastic lot-sizing problems with random lead times. *Operations Research Letters*, 36, 303–308.
- Jiang, R., & Guan, Y. (2011). An $\mathcal{O}(n^2)$ -time algorithm for the stochastic uncapacitated lot-sizing problem with random lead times. *Operations Research Letter*, 39, 74–77.

- Küçükyavuz, S., & Pochet, Y. (2007). Uncapacitated lot sizing with backlogging: the convex hull. *Mathematical Programming*, *118*, 151–175.
- Lee, Y. M., & Chen, E. J. (2005). Case studies: supply chain optimization models in a chemical company. In J. Geunes, E. Akcali, P. M. Pardalos, H. E. Romeijn, & Z. J. Shen (Eds.), *Applications of supply chain management and E-commerce research* (pp. 453–477). Berlin: Springer.
- Narasimhan, C. (1988). Competitive promotional strategies. *The Journal of Business*, *61*, 427–449.
- Nemhauser, G. L., & Wolsey, L. A. (1999). *Integer and combinatorial optimization*. New York: Wiley.
- Pochet, Y., & Wolsey, L. A. (1988). Lot-size models with backlogging: strong reformulations and cutting planes. *Mathematical Programming*, *40*, 317–335.
- Pochet, Y., & Wolsey, L. A. (1994). Polyhedra for lot-sizing with Wagner-Whitin costs. *Mathematical Programming*, *67*, 297–323.
- Pochet, Y., & Wolsey, L. (2006). *Production planning using mixed integer programming*. Berlin: Springer.
- Simester, D. (1997). Optimal promotion strategies: a demand-sided characterization. *Management Science*, *43*, 251–256.
- Terkaj, W., & Tolio, T. (2006). A stochastic approach to the FMS loading problem. *Journal of Manufacturing Systems*, *35*, 481–490.
- Wagelmans, A., van Hoesel, C., & Kolen, A. (1992). Economic lot sizing: an $O(n \log n)$ algorithm that runs in linear time in the Wagner-Whitin case. *Operations Research*, *40*, 145–156.
- Wagner, H. M., & Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management Science*, *5*, 89–96.
- Zanjania, M. K., Nourelfatha, M., & Ait-Kadia, D. (2010). A multi-stage stochastic programming approach for production planning with uncertainty in the quality of raw materials and demand. *International Journal of Production Research*, *48*, 4701–4723.