Operations Research Letters 42 (2014) 278–283

Contents lists available at ScienceDirect

Operations Research Letters

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

Extended formulations for stochastic lot-sizing problems

Chaoyue Zhao, Yongpei Guan *

Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, United States

ARTICLE INFO

Article history:
Received 12 November 2013
Received in revised form
13 April 2014
Accepted 13 April 2014
Available online 19 April 2014

Keywords:
Stochastic programming
Lot-sizing
Integral solution

ABSTRACT

In this paper, extended formulations for stochastic uncapacitated lot-sizing problems with and without backlogging are developed in higher dimensional spaces that provide integral solutions. Moreover, physical meanings of the decision variables in the extended formulations are explored and special cases with more efficient formulations are studied.

1. Introduction

The lot-sizing (LS) problem, determining when to produce and how much to produce at each time period so as to minimize the total production cost, is fundamental in production and inventory management. Meanwhile, significant research results have been derived for this problem and its deterministic variants, including polynomial time algorithms [8,3,7], cutting planes describing the convex hull in the original space [2], and extended formulations for the problems with Wagner–Whitin costs [6].

Recently, with the consideration of cost and demand uncertainties, as well as dependency among different time periods, scenario-tree based stochastic lot-sizing is introduced in [4]. Under this setting, the uncertain problem parameters are assumed to follow a discrete-time stochastic process with finite probability space and a scenario tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ is utilized to describe the resulting information structure. Each node $i \in \mathcal{V}$ corresponds to a possible realization of uncertain problem parameters up to the time period this node belongs to. We denote the corresponding probability as p_i. Except the root node, i.e., node 1, each node $i \in \mathcal{V}$ has a unique parent i^-. In addition, we let $\mathcal{C}(i)$ represent the set of children of node i, and $\mathcal{V}(i)$ represent the set of descendants of node i (including itself). Moreover, corresponding to each node $i \in \mathcal{V}$, we let α_i, h_i, β_i, and d_i represent unit production cost, holding cost, fixed setup cost, and demand respectively. All these parameters are assumed nonnegative with p_i included. The corresponding stochastic uncapacitated lot-sizing problem (SULS) can be formulated as follows:

$$\min \sum_{i\in\mathcal{V}} (\alpha_i x_i + \beta_i y_i + h_i s_i)$$

$$\text{s.t. } s_{i^-} + x_i = d_i + s_i, \quad (1)$$

$$x_i \le M y_i, \quad (2)$$

$$x_i \ge 0,\ s_i \ge 0,\ y_i \in \{0,1\},\ \forall i \in \mathcal{V}, \quad (3)$$

where x_i, y_i, and s_i represent the production level, the setup decision, and the inventory left at the end of the time period corresponding to the state defined by node i. Constraints (1) and (2) indicate inventory balance and production capacity.

The extended formulation for SULS is first attempted in [1], in which a reformulation is introduced to reduce the LP relaxation gap. Later on, in [9], an extended formulation of SULS is provided for a special case in which demands are deterministic (although costs are uncertain) and Wagner–Whitin costs are assumed. In this paper, we derive extended formulations for general SULS problems in which both demands and costs are uncertain.

2. Extended formulation for SULS

We first review the optimality conditions and the dynamic programming algorithm for SULS (as described in [5]). Then, we derive the corresponding dual formulation in the dual space. Finally, we develop the extended formulation of SULS in a higher dimensional space. Without loss of generality, we assume the initial inventory level is zero. For notation brevity, we add a dummy node 0 which is the parent of node 1 and define $\bar{\mathcal{V}} = \mathcal{V} \cup \{0\}$. We let d_{1i} represent the cumulative demand from the root node 1 to node i and define $d_{10} = 0$. Accordingly we have $d_{ik} = d_{1k} - d_{1i^-}$ for each $k \in \mathcal{V}(i)$.

Optimality conditions and dynamic programming for SULS: As shown in [5], the key concept for the optimality conditions has two folds: (1) for each node $i \in \mathcal{V}$, if we choose to produce at this node, then the production amount plus the entering inventory to this node should be able to exactly cover the cumulative demand from node

* Corresponding author.
E-mail address: guan@ise.ufl.edu (Y. Guan).

http://dx.doi.org/10.1016/j.orl.2014.04.003
0167-6377/

i to a descendant of node i. That is, if (x^*, y^*, s^*) is the optimal solution to SULS and meanwhile for each node i, if

$$x_i^* > 0, \quad \text{then } x_i^* = d_{ik} - s_{i^-}^* \text{ for some node } k \in \mathcal{V}(i); \tag{4}$$

(2) the inventory level entering a node has a limited number of candidates based on (1). More specifically, the inventory level under the optimality conditions can be described as follows:

$$s_i^* = d_{1j} - d_{1i} \quad \text{for some node } j \in \mathcal{V} : d_{1j} \ge d_{1i}. \tag{5}$$

Based on the above two important insights for the problem, the backward induction dynamic programming framework can be explored based on if production setup is taken or not at a particular node. Let $K(i, j)$ represent the optimal value function for node i when the inventory left from the previous period i^- is $d_{1j} - d_{1i^-}$.

If a production is set up at node i, the total costs in this node contain the setup, production, and holding costs corresponding to this node and the costs for descendant nodes. Therefore, the corresponding value function, for which we denote it as $K_{\mathrm{P}}(i, j)$, is equal to

$$\min_{k\in\mathcal{V}(i):d_{1k}>d_{1j}} \left\{ \beta_i + \alpha_i(d_{1k} - d_{1j}) + h_i(d_{1k} - d_{1i}) + \sum_{l\in\mathcal{C}(i)} K(l, k) \right\}. \tag{6}$$

Otherwise, if no production occurs at this node, the total costs for this node are the holding cost corresponding to this node, together with the costs for descendant nodes. Thus, the corresponding value function, denoted as $K_{\mathrm{NP}}(i, j)$, is

$$h_i(d_{1j} - d_{1i}) + \sum_{l\in\mathcal{C}(i)} K(l, j). \tag{7}$$

Finally, if both options are available for node i, the actual value function $K(i, j)$ is the minimum of the two, i.e., $K(i, j) = \min\{K_{\mathrm{P}}(i, j), K_{\mathrm{NP}}(i, j)\}$. Meanwhile, for the special cases in which the production is necessary at node i (i.e., $d_{1j} - d_{1i^-} < d_i$) and no production is needed at node i (i.e., $d_{1j} - d_{1i^-} \ge \max_{l\in\mathcal{V}(i)} d_{il}$), $K(i, j)$ is equal to $K_{\mathrm{P}}(i, j)$ and $K_{\mathrm{NP}}(i, j)$ respectively.

Extended formulation of SULS: Now, we can formulate SULS as a linear program (denoted as SULS-L) incorporating dynamic programming Bellman equations as constraints:

$$\max K(1, 0)$$

$$\text{s.t. } K(1, 0) \le C_{10k} + \sum_{l\in\mathcal{C}(1)} K(l, k), \quad \forall k \in \mathcal{V}(1), \tag{8}$$

$$K(i, j) \le C_{ijk} + \sum_{l\in\mathcal{C}(i)} K(l, k), \quad \forall i \in \mathcal{V} \setminus \{1\}, \ \forall j \in \mathcal{V}, \text{ and } \forall k \in \mathcal{V}(i) : d_{1k} > d_{1j} \ge d_{1i^-}, \tag{9}$$

$$K(i, j) \le D_{ij} + \sum_{l\in\mathcal{C}(i)} K(l, j), \quad \forall i \in \mathcal{V} \setminus \{1\}, \ \forall j \in \mathcal{V} : d_{1j} \ge d_{1i}, \tag{10}$$

where the parameters

$C_{ijk} = \beta_i + \alpha_i(d_{1k} - d_{1j}) + h_i(d_{1k} - d_{1i})$ and
$D_{ik} = h_i(d_{1j} - d_{1i})$.

In the above formulation, constraints (8) correspond to (6) for the root node case and constraints (9) correspond to (6) for non-root nodes. Constraints (10) correspond to (7). For notation brevity, we define $\mathcal{I}_i = \{0\}$ if $i = 1$ and $\mathcal{I}_i = \mathcal{V}$ otherwise, and $\mathcal{V}_{\mathrm{T}}(t)$ as the node set that covers the nodes with the corresponding time periods larger than t. Moreover, we define two index sets:

$$\Pi = \{(i, j, k) | i \in \mathcal{V}, j \in \mathcal{I}_i, k \in \mathcal{V}(i) : d_{1k} > d_{1j} \ge d_{1i^-}\}$$
$$\Gamma = \{(i, j) | i \in \mathcal{V}, j \in \mathcal{I}_i : d_{1j} \ge d_{1i}\}.$$

Note here that Π represents the set of all possible production combinations and Γ represents the set of all possible non-production combinations. We can get the dual formulation (denoted as SULS-D) of SULS-L as follows:

$$\min_{w,u} \sum_{(i,j,k)\in\Pi} C_{ijk} w_{ijk} + \sum_{(i,j)\in\Gamma} D_{ij} u_{ij} \tag{11}$$

$$\text{s.t. } \sum_{k\in\mathcal{V}(1)} w_{10k} = 1, \tag{12}$$

$$\sum_{k\in\mathcal{V}(i)} w_{ijk} + u_{ij} - w_{10j} = 0, \quad \forall i \in \mathcal{C}(1), \ \forall j \in \mathcal{V}(1), \tag{13}$$

$$\sum_{k\in\mathcal{V}(i)} w_{ijk} + u_{ij} - \sum_{l\in\mathcal{V}} w_{i^-lj} - u_{i^-j} = 0, \quad \forall i \in \mathcal{V}_{\mathrm{T}}(2), \ \forall j \in \mathcal{V}(i^-), \tag{14}$$

$$\sum_{k\in\mathcal{V}(i)} w_{ijk} + u_{ij} - u_{i^-j} = 0, \quad \forall i \in \mathcal{V}_{\mathrm{T}}(2), \ \forall j \in \mathcal{V} \setminus \mathcal{V}(i^-), \tag{15}$$

where w_{ijk} and u_{ij} are dual variables corresponding to constraints (9) and (10) respectively.

In the remaining part of this paper, we demonstrate that the above SULS-D can automatically generate integral solutions for w and u. Furthermore, we provide physical implications of these two introduced variables. We first need to prove the following important lemma.

Lemma 1. *The extreme points of SULS-D are binary.*

Proof. To prove the proposition, it is equivalent to prove that for $\forall C_{ijk} \in (-\infty, +\infty)$ and $\forall D_{ij} \in (-\infty, +\infty)$, the optimal solutions of SULS-D are binary. First, by solving SULS using the dynamic programming approach with respect to C and D, we can obtain an optimal decision and accordingly we can introduce $\hat{w}$ and $\hat{u}$, which are binary, to represent the optimal decision. For a given optimal decision, if the inventory level entering node i is equal to $d_{1j} - d_{1i^-}$ and production amount at node i can exactly cover the cumulative demand from node i to its descendant node k, we let $\hat{w}_{ijk} = 1$. Otherwise, $\hat{w}_{ijk} = 0$. Similarly, if the inventory level entering node i is $d_{1j} - d_{1i^-}$ and no production occurs at node i, we let $\hat{u}_{ij} = 1$. Otherwise, $\hat{u}_{ij} = 0$. In the following, we prove that this generated $(\hat{w}, \hat{u})$ is an optimal solution to the dual formulation SULS-D, which is sufficient to prove our claim.

We first verify that $\hat{w}$ and $\hat{u}$ are feasible solutions to SULS-D, i.e., $\hat{w}$ and $\hat{u}$ satisfy constraints (12) to (15). For constraint (12), since the inventory level entering node 1 is 0, production has been set up in node 1 to cover the cumulative demand from node 1 to one of its descendant nodes. Constraint (13) indicates the optimality conditions applied to the children nodes of root node 1. For any pair $i \in \mathcal{C}(1), j \in \mathcal{V}(1)$, if $\hat{w}_{10j} = 1$, which indicates that the production amount at node 1 exactly covers the cumulative demand from node 1 to node j. Then, for node i, we can choose to produce to cover the cumulative demand from node i to some descendant node of node i or no production. That is, $\sum_{k\in\mathcal{V}(i)} \hat{w}_{ijk} + \hat{u}_{ij} = 1$. Otherwise, if $\hat{w}_{10j} = 0$, then the inventory level entering node i is not equal to $d_{1j} - d_1$, we have $\sum_{k\in\mathcal{V}(i)} \hat{w}_{ijk} + \hat{u}_{ij} = 0$, based on the definitions of $\hat{w}_{ijk}$ and $\hat{u}_{ij}$. Similar arguments can be applied for constraints (14) and (15), depending on whether $j \in \mathcal{V}(i^-)$ or not. If $j \in \mathcal{V}(i^-)$, i.e., corresponding to constraint (14), then it is possible to set up at node i^- and cover the cumulative demand from node i^- to node j, based on the optimality condition described in (4). Therefore, there are two possible ways to make the inventory level entering node i to be $d_{1j} - d_{1i^-}$: (1) production is set up at node i^- and cover the cumulative demand from node i^- to node j, or (2) no production is set up at node i^- and the entering inventory for node i^- covers the demand until node j. This indicates the equivalence between