A STUDY OF THE LOT–SIZING POLYTOPE

ALPER ATAMTÜRK AND JUAN CARLOS MUÑOZ

Abstract. The lot–sizing polytope is a fundamental structure contained in many practical production planning problems. Here we study this polytope and identify facet–defining inequalities that cut off all fractional extreme points of its linear programming relaxation, as well as liftings from those facets. We give a polynomial–time combinatorial separation algorithm for the inequalities when capacities are constant. We also report on an extensive computational study on solving the lot–sizing problem for instances up to 365 time periods with varying cost and capacity characteristics.

September 2002

1. Introduction

Given the demand, production cost, and inventory holding cost for a product and production capacities and production setup cost for each time period over a finite discrete–time horizon, the lot–sizing problem is to determine how much to produce and hold as inventory in each time period so that the sum of production, inventory holding, and setup costs over the horizon is minimized. The lot–sizing problem (LSP) is $\mathcal{NP}$–hard [8]. Several special cases, including the uncapacitated and constant–capacity cases, of the problem are solved in polynomial time; see [5, 6, 7, 17, 18, 19].

Many practical multi–item, multi–stage production planning problems over a finite discrete–time horizon contain the lot–sizing problem as a substructure. Strong inequalities and reformulations for the lot–sizing problem often form the basis of branch–and–cut algorithms and good models for those more complicated production planning problems; see for instance [3, 4, 15, 21]. Therefore, a good understanding of the lot–sizing polytope has immediate implications for many practical production problems. Here we investigate this basic polytope.

A. Atamtürk: Department of Industrial Engineering and Operations Research, 4135 Etcheverry Hall, University of California, Berkeley 94720–1777. atamturk@ieor.berkeley.edu. Supported, in part, by NSF grants DMII–0070127 and DMII–0218265, and a grant from ILOG, Inc.
J.C. Muñoz: Department of Civil and Environmental Engineering, 416G McLaughlin Hall, University of California, Berkeley 94720–1720. juanca@uclink4.berkeley.edu. Instructor at the Pontificia Universidad Católica de Chile.

The uncapacitated lot–sizing polytope is described completely by Barany et al. [2]. The constant–capacity lot–sizing polytope is studied by Leung et al. [10] and Pochet and Wolsey [16]. When there are no restrictions on the capacity Pochet [14] gives valid inequalities for LSP based on single node flow relaxations, Miller et al. [12] describe inequalities from its continuous knapsack relaxations. Loparic et al. [11] study a dynamic knapsack set relaxation of LSP. Here we study the lot–sizing polytope directly, and identify facet–defining inequalities that cut–off all of the fractional vertices of its linear programming relaxation as well as liftings from those facets.

Let p_t, h_t, and s_t denote the production, holding, and setup costs in period t, and i_t, w_t, and z_t denote the incoming inventory, production, and setup variables in period $t,\ t \in \{1, 2, \ldots, n\}$, respectively. Then LSP can be formulated as

$$\min \left\{ \sum_{i=1}^{n} (h_t i_t + p_t w_t + s_t z_t) : (w, i, z) \in E \right\}$$

where

$$E := \left\{ z \in \{0,1\}^n, w \in \mathbb{R}^n_+, i \in \mathbb{R}^{n+1}_+ : i_t + w_t - i_{t+1} = d_t,\ w_t \le c_t z_t,\ t \in [1,n],\ i_{n+1} = 0 \right\},$$

d_t is the demand and c_t is the production capacity in period t.

Eliminating the inventory variables, by substituting $i_t = d_t - w_t + i_{t+1}$ in $i_t \ge 0$ and reindexing the variables in the reverse order, gives the following equivalent model of the feasible solutions of LSP:

$$F := \left\{ x \in \{0,1\}^n, y \in \mathbb{R}^n_+ : \begin{array}{r} y_1 \le u_1 \\ y_1 + y_2 \le u_2 \\ \vdots \\ y_1 + y_2 + \cdots + y_n \le u_n \\ y_1 \le a_1 x_1,\ \cdots,\ y_n \le a_n x_n \end{array} \right\}$$

where $u_i = \sum_{j=i}^{n} d_{n-j+1}$ and $a_i = c_{n-i+1}$ for $i \in [1, n]$, that is $u_1 = d_n$, $u_2 = d_n + d_{n-1}$, and so on. We refer to F as the *bottleneck flow model* of LSP. The full–dimensional bottleneck flow model reveals more of the structure of the lot–sizing problem than the standard formulation E; therefore, in the rest of the paper we will work on F. Throughout we let $[i, k] := \{j \in \mathbb{Z} : i \le j \le k\}$, $a^+ := \max\{a, 0\}$, and e_i denote the ith unit vector.

Section 2 is devoted to the analysis of the lot–sizing polytope. The inequalities given in Section 2 are specialized for the uncapacitated and constant–capacity cases in Section 3. The computational studies with the new inequalities when used as cutting planes are described in Section 4. We conclude with Section 5.

2. Polyhedral Analysis

Throughout the paper we assume that the data of the bottleneck flow model F consists of rational numbers and *without loss of generality* satisfy the following:

Assumptions.

(A1) $0 < a_i \le u_i$ for all $i \in [1, n]$,
(A2) $0 \le u_i - u_{i-1} \le a_i$ for all $i \in [1, n]$, where $u_0 = 0$.

If $a_i \le 0$, then (x_i, y_i) can be dropped. If $u_i < a_i$, then a_i can be reduced as u_i without changing F. If $u_i < u_{i-1}$, then u_{i-1} can be reduced to u_i, and if $u_i > u_{i-1} + a_i$, then u_i can be reduced to $u_{i-1} + a_i$ without changing F.

Proposition 1. *Dimension & trivial facets.*

1. *$Conv(F)$ is full–dimensional,*
2. *$y_i \ge 0$, $x_i \le 1$, $y_i \le a_i x_i$, $i \in [1, n]$, define facets of $conv(F)$,*
3. *$y_1 + \cdots + y_i \le u_i$, $i \in [2, n]$, defines a facet of $conv(F)$ if and only if $u_i \le u_{k-1} + \sum_{j=k+1}^{i} a_j$ for all $k \in [1, i]$ and $u_i < u_{i+1}$, where $u_{n+1} = \infty$.*

Proof. Parts 1 and 2 follow from **(A1)** immediately. We prove part 3. If $u_i > u_{k-1} + \sum_{j=k+1}^{i} a_j$ for some $k \in [1, i]$, then inequality (2), introduced in Section 2.2, dominates $y_1 + \cdots + y_i \le u_i$. Else, $y_1 + \cdots + y_i + y_{i+1} \le u_i$ dominates it whenever $u_i = u_{i+1}$. For sufficiency, we give $2n$ affinely independent points (x^k, y^k) of F satisfying $y_1 + \cdots + y_i = u_i$. The first $2i$ points are: $y^k_j > 0$ and $x^k_j = 1$ for $j \in [1, k-1]$ such that $\sum_{j=1}^{k-1} y^k_j = u_{k-1}$; $y^k_k = 0$ and $x^k_k \in \{0, 1\}$; and $y^k_j > 0$ and $x^k_j = 1$ such that $\sum_{j=k+1}^{i} y^k_j = u_i - u_{k-1}$; and $y^k_j = x^k_j = 0$ for $j \in [i+1, n]$ and $k \in [1, i]$. The remaining $2(n - i)$ points are (y^i, e_k) and $(y^i + \min\{a_k, u_{i+1} - u_i\}e_k, e_k)$ for $k \in [i + 1, n]$. □

2.1. **A min–max relationship.** Let $S = \{s_1, s_2, \ldots, s_p\}$ be a subset of $[1, n]$ indexed in increasing order of s_i, and, for simplicity of notation, let $s_0 = 0$. The following min–max relationship, which follows from the lower–triangular structure of the first set of constraints of F, is key for understanding the structure of $conv(F)$:

$$\zeta(S) := \max\left\{\sum_{i \in S} y_i : (x, y) \in F\right\} = \min_{0 \le i \le p}\left\{u_{s_i} + \sum_{k=i+1}^{p} a_{s_k}\right\}. \tag{1}$$

Definition 1. The smallest minimizer in (1) is called the *bottleneck of the set S* and is denoted as $b_S (\in [0, p])$. Let $S_i := \{s_1, s_2, \ldots s_i\}$ for $i \in [1, p]$. $B_S := \{b_{S_i} : i \in [1, p]\}$ is called the *bottleneck set of S*. For a given S and $i \in [1, p]$, we say that the *bottleneck of the ith element*, denoted as b_i, is the bottleneck of the *set* S_{i-1}.

Example 1. Consider an instance of F with $u = (5, 8, 11, 13)$ and $a = (5, 9, 7, 12)$. For $S = \{s_1, s_2\} = \{3, 4\}$, we have $B_S = \{0, 2\}$ and $b_1 = b_2 = 0$, since $u_{s_0} + a_{s_1} = 0 + 7 \leq u_{s_1} = 11$. On the other hand, for $S = \{s_1, s_2, s_3, s_4\}$, we have $B_S = \{0, 2, 3, 4\}$ and $b_1 = b_2 = 0$, whereas $b_3 = 2$ and $b_4 = 3$. □

Observation 1. If p is the bottleneck of $S = \{s_1, s_2, \ldots, s_p\}$; i.e., $b_S = p$, then

1. $\zeta(S \setminus \{s_k\}) = \min\left\{u_{s_p}, u_{s_{b_k}} + \sum_{i=b_k+1}^{p} a_{s_i} - a_{s_k}\right\}$,
2. $\zeta(S \cup \{k\}) = \begin{cases} u_{s_p} & \text{if } k \leq s_p, \\ \min\{u_{s_p} + a_k, u_k\} & \text{if } k > s_p. \end{cases}$

Observation 1 is a consequence of the min–max relationship (1) and will be useful in proving validity of the inequalities in Sections 2.2 and 2.3.

2.2. **Bottleneck covers.**

Definition 2. Given $S = \{s_1, s_2, \ldots, s_p\} \subseteq [1, n]$, let $\lambda_i := \sum_{j=b_i+1}^{p} a_{s_j} - (u_{s_p} - u_{s_{b_i}})$ for $i \in [1, p]$. S is called a *bottleneck cover* if $\lambda_i > 0$ for *some* $i \in [1, p]$.

Observation 2. For any $S = \{s_1, s_2, \ldots, s_p\} \subseteq [1, n]$ the following statements are true:

1. $\lambda_i \geq \lambda_k$ for $i, k \in [1, p]$ with $i \leq k$,
2. $\lambda_i = \lambda_k$ if and only if $b_i = b_k$ for $i, k \in [1, p]$,
3. $\lambda_i > \lambda_{i+1}$ if and only if $b_{i+1} = i$ for $i \in [1, p-1]$,
4. $\lambda_p > 0$ if and only if $b_S = p$.

For a bottleneck cover S, we define the *bottleneck cover inequality* as

$$\sum_{i=1}^{p} \min\{a_{s_i}, (a_{s_i} - \lambda_i)^+\}(1 - x_{s_i}) + \sum_{i=1}^{p} y_{s_i} \leq u_{s_p}. \tag{2}$$

Example 1 (cont.) For $S = \{s_1, s_2, s_3, s_4\} = \{1, 2, 3, 4\}$, we have $\lambda_1 = \lambda_2 = 20$, $\lambda_3 = 14$, and $\lambda_4 = 10$, and thus inequality (2) is

$$2(1 - x_4) + y_1 + y_2 + y_3 + y_4 \leq 13.$$

For $S = \{s_1, s_2\} = \{2, 3\}$, $\lambda_1 = 5$ and $\lambda_2 = 4$, and inequality (2) becomes

$$4(1 - x_2) + 3(1 - x_3) + y_2 + y_3 \leq 11.$$

□

Remark 1. Observe that if $u_{s_i} = u_{s_p}$ for all $i \in [1, p]$, then $b_i = 0$ and $\lambda_i = \lambda_p$ for all $i \in [1, p]$; consequently, the bottleneck cover inequality reduces to the flow cover inequality [13] for constraint $y_1 + y_2 + \cdots + y_{s_p} \leq u_{s_p}$. Otherwise, bottleneck cover inequalities dominate all flow cover inequalities from this constraint.

Proposition 2. *Let LF denote the linear programming relaxation of F.*

1. *Every fractional extreme point of LF is defined by a bottleneck cover.*
2. *Bottleneck cover inequalities* (2) *cut off all fractional extreme points of LF.*

Proof. 1. Let (x, y) be an extreme point of LF. Observe that x_i equals either 1 or y_i/a_i; and $0 < x_i < 1$ implies that $0 < y_i < a_i$ and $y_i = u_i - \sum_{j=1}^{i-1} y_j$. Now let $S = \{s_1, s_2, \ldots, s_p\} = \{i \in [1, n] : y_i > 0\}$ such that $0 < x_{s_p} < 1$ and let $k = \max\{i \in [0, p-1] : x_{s_i} < 1\}$. Because either $k = 0$, or $k \geq 1$ and $y_{s_k} = u_{s_k} - \sum_{i=1}^{k-1} y_{s_i}$, we have $y_{s_p} = u_{s_p} - u_{s_k} - \sum_{i=k+1}^{p-1} a_{s_i}$. Since $y_{s_p} < a_{s_p}$, we have $u_{s_p} < u_{s_k} + \sum_{i=k+1}^{p} a_{s_i}$, and since (x, y) is feasible, we have $u_{s_k} + \sum_{i=k+1}^{j} a_{s_i} \leq u_{s_j}$ for all $j \in [k+1, p-1]$. Then, by induction, S is a bottleneck cover and k is the bottleneck of p. Hence, $\{i \in [1, p] : 0 < x_{s_i} < 1\}$ is precisely the bottleneck set of S, B_S. 2. Since $y_{s_p} > 0$, we have $\lambda_p = u_{s_k} + \sum_{i=k+1}^{p} a_{s_i} - u_{s_p} < a_{s_p}$. Then inequality (2) defined by $S = \{i \in [1, n] : y_i > 0\}$ cuts off (x, y) as $\sum_{i=1}^{p} y_{s_i} = u_{s_p}$ and $0 < x_{s_p} < 1$. □

Proposition 3. *The bottleneck cover inequality* (2) *is valid for F.*

Proof. The statement holds trivially if $a_{s_i} \leq \lambda_i$ for all $i \in [1, p]$. Otherwise, for $(x, y) \in F$ let $\{z_1, z_2, \ldots, z_\ell\} := \{i \in [1, p] : x_{s_i} = 0\}$, indexed in increasing order of i, and let $k = \min\{j \in [1, \ell] : a_{s_{z_j}} > \lambda_{z_j}\}$. Then

$$\sum_{i=1}^{p} \min\{a_{s_i}, (a_{s_i} - \lambda_i)^+\}(1 - x_{s_i}) = \sum_{j=k}^{\ell} \min\{a_{s_{z_j}}, (a_{s_{z_j}} - \lambda_{z_j})^+\}$$

$$\leq a_{s_{z_k}} - \lambda_{z_k} + \sum_{i=k+1}^{\ell} a_{s_{z_i}} \leq \max_{j \in [1,\ell]} \left\{ (\sum_{i=j}^{\ell} a_{s_{z_i}} - \lambda_{z_j})^+ \right\}. \tag{3}$$

On the other hand, we also have

$$\sum_{i=1}^{p} y_{s_i} \leq \min \left\{ u_{s_p}, \min_{j \in [1,\ell]} \{ u_{s_{b_{z_j}}} + \sum_{k=b_{z_j}+1}^{p} a_{s_k} - \sum_{k=j}^{\ell} a_{s_{z_k}} \} \right\}$$

$$= u_{s_p} - \max_{j \in [1,\ell]} \left\{ (u_{s_p} - u_{s_{b_{z_j}}} - \sum_{k=b_{z_j}+1}^{p} a_{s_k} + \sum_{k=j}^{\ell} a_{s_{z_k}})^+ \right\}$$

$$= u_{s_p} - \max_{j \in [1,\ell]} \left\{ (\sum_{k=j}^{\ell} a_{s_{z_k}} - \lambda_{z_j})^+ \right\}. \tag{4}$$

Adding (3) and (4) shows that inequality (2) is satisfied by $(x, y) \in F$. □

Theorem 4. *The bottleneck cover inequality* (2) *defines a facet of conv*(F) *if and only if* $\lambda_p > 0$, $max_{i\in[1,p]}\{a_{s_i} - \lambda_i\} > 0$, $[1, s_{b_r}] \subset S$, $u_{s_{b_r}} < u_{s_{b_r}+1}$, *where* $r = min\{i \in [1,p] : \lambda_i < a_{s_i}\}$, *and* $K = \{k \in [1,n] \setminus S : u_{b_r} \le u_k \le u_t\} = \emptyset$, *where* $t = max\{i \in [b_r, r-1] : u_{s_{b_r}} + \sum_{k=b_r+1}^{i} a_{s_k} = u_{s_i}\}$.

Proof. Necessity. Let $q \in [1,p]$ be the highest index such that $\lambda_q > 0$. Since S is a bottleneck cover, q exists. Then by updating S as $\{s_1, s_2, \ldots, s_q\}$, we get an inequality dominating (2), since $\sum_{i=q+1}^{p}(a_{s_i}(1-x_{s_i}) + y_{s_i}) \le \sum_{i=q+1}^{p} a_{s_i} \le u_{s_p} - u_{s_q}$. If $\max_{i\in[1,p]}\{a_{s_i} - \lambda_i\} \le 0$, then $y_1 + \cdots + y_{s_p} \le u_{s_p}$ dominates (2). If $k \in [1, s_{b_r}] \setminus S$, since b_r is the bottleneck of $[1, s_{b_r}]$, by Observation 1, augmenting S with k, does not change λ_i for $i \in [b_r+1, p]$ and cannot decrease λ_i for $i \in [1, b_r]$. Since the coefficients of all x_{s_i} with $i \in [1, r-1]$ are zero, the inequality with S equal to $\{s_1, \ldots, s_p\} \cup \{k\}$ dominates (2). If $u_{s_{b_r}} = u_{s_{b_r}+1}$, then the inequality obtained by augmenting S with $s_{b_r} + 1$ dominates (2). Finally, note that if $t > b_r$, then t is an alternative minimizer in (1), achieving the value $\zeta(\{s_1, \ldots, s_{r-1}\})$. Thus, S can be augmented with $k \in K$ without changing λ_i for $i \in [r,p]$ to get an inequality stronger than (2).

Sufficiency. The following $2n$ points (x^k, y^k) are affinely independent points of the face of *conv*(F) generated by inequality (2). For $k \in [1,p]$ such that $0 < \lambda_k < a_{s_k}$, let $y^{s_k}_{s_i} > 0$ and $x^k_{s_i} = 1$ for $i \in [1, b_k]$ such that $\sum_{i=1}^{b_k} y_{s_i} = u_{s_{b_k}}$, $y^{s_k}_{s_k} = 0$ and $x^{s_k}_{s_k} = 0$, and $y^{s_k}_{s_i} = a_{s_j}$ and $x^{s_k}_{s_i} = 1$ for $i \in [b_k+1, p] \setminus \{k\}$; and $y^{s_k}_i = x^{s_k}_i = 0$ for $i \in [1,n] \setminus S$; and let $y^{s_k}_{s_i} > 0$ and $x^k_{s_i} = 1$ for $i \in [1, b_k]$ such that $\sum_{i=1}^{b_k} y_{s_i} = u_{s_{b_k}}$, $y^{s_k}_{s_k} = a_{s_k} - \lambda_k$ and $x^{s_k}_{s_k} = 1$, and $y^{s_k}_{s_i} = a_{s_i}$ and $x^{s_k}_{s_i} = 1$ for $i \in [b_k+1, p] \setminus \{k\}$; and $y^{s_k}_i = x^{s_k}_i = 0$ for $i \in [1,n] \setminus S$.

For $k \in [1,p]$ such that $\lambda_k \ge a_{s_k}$, let $y^{s_k}_{s_i} > 0$ and $x^k_{s_i} = 1$ for $i \in [1, b_k]$ such that $\sum_{i=1}^{b_k} y_{s_i} = u_{s_{b_k}}$, $y^{s_k}_{s_k} = 0$ and $x^{s_k}_{s_k} \in \{0,1\}$, and $y^{s_k}_{s_i} > 0$ and $x^{s_k}_{s_i} = 1$ for $i \in [b_k+1, p] \setminus \{k\}$ such that $\sum_{i=b_k+1}^{p} y_{s_i} = u_{s_p} - u_{s_{b_k}}$ and $y^{s_k}_i = x^{s_k}_i = 0$ for $i \in [1,n] \setminus S$.

Let $\bar{y} = \sum_{i=1}^{b_r} \epsilon_i e_{s_i} + \sum_{i=b_r+1}^{p} a_{s_i} e_{s_i} - a_{s_r} e_{s_r}$ and $\bar{x} = \sum_{i=1}^{p} e_{s_i} - e_{s_r}$ such that $a_i \ge \epsilon_i > 0$ and $\sum_{i=1}^{b_r} \epsilon_i = u_{b_r}$. We see that $\bar{y}$ has a positive slack for constraints $y_1 + \cdots + y_i \le u_i$ with $s_{b_r}+1 \le i < s_{b_r+1}$ since $u_{s_{b_r}} < u_{s_{b_r}+1}$; with $s_{b_r+1} \le i < s_r$ since $K = \emptyset$; with $i \ge s_r$ since $\bar{y}$ has a slack of $a_{s_r} - \lambda_r$ for constraint $y_1 + \cdots + y_{s_p} \le u_{s_p}$ and p is the bottleneck of S. Then the remaining $2(n-p)$ points are $(\bar{y}, \bar{x} + e_k)$ and $(\bar{y} + \epsilon e_k, \bar{x} + e_k)$ for $k \in [s_{b_r}+1, n] \setminus S$ with small $\epsilon > 0$. □

Remark 2. The necessity part of the proof of Theorem 4 shows that if any of the facet conditions is not satisfied by a bottleneck flow cover inequality (2), then a stronger inequality is immediately available. Observe that both of the inequalities in Example 1 satisfy the conditions of Theorem 4.

As pointed out to us by L. A. Wolsey, the bottleneck cover inequalities are part of the much general submodular inequalities for fixed charge networks [20]. The min–max relationship (1) allows us to characterize these inequalities explicitly for the lot–sizing problem and lift them to derive further strong inequalities.

2.3. **Lifting bottleneck covers.** In this section we generalize bottleneck cover inequalities (2) by introducing pairs of variables $(x_i, y_i),\ i \subseteq [1,n] \setminus S$ to them. Let $T \subset [1,n]$ and consider the restriction $F_T := \{(x,y) \in F : x_i = y_i = 0 \text{ for } i \in T\}$ of F and a facet–defining bottleneck inequality

$$\sum_{i=1}^{p}((a_{s_i} - \lambda_i)^+(1 - x_{s_i}) + y_{s_i}) \leq u_{s_p} \tag{5}$$

defined by some $S = \{s_1, s_2, \ldots, s_p\} \subseteq [1,n] \setminus T$ for $conv(F_T)$. We will derive a new inequality of the form

$$\sum_{i=1}^{p}((a_{s_i} - \lambda_i)^+(1 - x_{s_i}) + y_{s_i}) + \sum_{i \in T}(\pi_i x_i + \mu_i y_i) \leq u_{s_p} \tag{6}$$

starting from (5).

Let $F_T(u)$ be the set F_T as a function of the right hand side vector $u \in \mathbb{R}^n$. Let $\Phi : \mathbb{R}^n_+ \mapsto \mathbb{R} \cup \{+\infty\}$ be defined as

$$\Phi(v) = u_{s_p} - \max\left\{\sum_{i=1}^{p}((a_{s_i} - \lambda_i)^+(1 - x_{s_i}) + y_{s_i}) : (x,y) \in F_T(u - v)\right\}.$$

By definition of Φ, inequality (6) is valid for F if and only if

$$\sum_{i \in T}(\pi_i x_i + \mu_i y_i) \leq \Phi(\sum_{i \in T} y_i g_i) \tag{7}$$

for all $(x,y) \in F$, where $g_i = \sum_{k=i}^{n} e_k$, $i \in [1,n]$ and e_k is the kth unit vector in $\mathbb{R}^n$.

Rather than characterizing $\Phi(\sum_{i \in T} y_i g_i)$ for all $(x,y) \in F$, we will describe a lower bound on $\Phi(a g_\ell)$ for $0 \leq a \leq u_\ell$ and $\ell \in T$, which suffices to prove the validity of the inequalities introduced in this section. For some $\ell \in T$ let $P_{\Phi(a g_\ell)}$ denote the problem of computing $\Phi(a g_\ell)$. Since $(x_i, y_i),\ i \in [1,n] \setminus S$ do not appear in inequality (5), we

may ignore them when computing $\Phi(ag_\ell)$. Thus, $P_{\Phi(ag_\ell)}$ can be stated as

$$\begin{aligned}
\Phi(ag_\ell) = u_{s_p} - \max \sum_{i=1}^{p} ((a_{s_i} - \lambda_i)^+ (1 - x_{s_i}) + y_{s_i}) \\
\text{s.t.:} \qquad\qquad y_{s_1} &\le u_{s_1} \\
y_{s_1} + y_{s_2} &\le u_{s_2} \\
&\vdots \\
y_{s_1} + y_{s_2} + \cdots + y_{s_p} &\le u_{s_p} \qquad (8) \\
&\vdots \\
y_{s_1} + y_{s_2} + \cdots + y_{s_p} &\le u_\ell - a \\
&\vdots \\
y_{s_1} + y_{s_2} + \cdots + y_{s_p} &\le u_n - a \\
y_{s_i} \le a_{s_i} x_{s_i},\ y_{s_i} \in \mathbb{R}_+,\ x_{s_i} &\in \{0,1\} \quad i \in [1,p].
\end{aligned}$$

Proposition 5. *Problem $P_{\Phi(ag_\ell)}$ has an optimal solution (x, y) such that*

1. *$x_{s_i} = 1$ for all $i \in [1,p]$ with $a_{s_i} \le \lambda_i$,*
2. *$a_{s_k} > y_{s_k} > (a_{s_k} - \lambda_k)^+$ for at most one $k \in [1,p]$,*
3. *$y_{s_i} \in \{0, a_{s_i}\}$ for all $i \in [1,p] \setminus \{k\}$ with $a_{s_i} > \lambda_i$, and*
4. *if $\sum_{i=1}^{h} y_{s_i} = u_{s_h}$ for $h \in [1,p]$, then $y_{s_i} > 0$ for all $i \in [1,h]$ with $a_{s_i} > \lambda_i$.*

Proof. Part 1 is immediate. For part 2, suppose $a_{s_i} > y_{s_i} > (a_{s_i} - \lambda_i)^+$ and $a_{s_j} > y_{s_j} > (a_{s_j} - \lambda_j)^+$ for $i < j$. Increasing y_{s_j} and decreasing y_{s_i} by the same amount sufficiently we satisfy either $y_{s_j} = a_{s_j}$ or $y_{s_i} = (a_{s_i} - \lambda_i)^+$ and do not degrade the objective function value. To see part 3, observe that if $y_{s_i} \le a_{s_i} - \lambda_i$, then the objective function improves by $a_{s_i} - \lambda_i - y_{s_i}$ by setting $y_{s_i} = x_{s_i} = 0$.

Part 4 is a consequence of feasibility. By definition of b_h, $u_{s_h} + \sum_{i=h+1}^{p} a_{s_i} \ge u_{b_h} + \sum_{i=b_h+1}^{p} a_{s_i} = u_{s_p} + \lambda_h$. Since for any $j \in [1,h]$, we also have $u_{s_{b_j}} + \sum_{i=b_j+1}^{p} a_{s_i} = u_{s_p} + \lambda_j$, it follows that $u_{s_h} \ge u_{s_{b_j}} + \sum_{i=b_j+1}^{h} a_{s_i} - (\lambda_j - \lambda_h)$ for $j \le h$. Therefore, if $a_{s_j} > \lambda_j$, since $\lambda_h \ge 0$ (as (5) is facet–defining for $conv(F_T)$), we have $u_{s_h} > u_{s_{b_j}} + \sum_{i=b_j+1}^{h} a_{s_i} - a_{s_j}$ and $y_{s_j} = 0$, which contradicts $\sum_{i=1}^{h} y_{s_i} = u_{s_h}$. □

First suppose that $\ell > s_p$; we will remove this restriction later. Since (5) is facet–defining for $conv(F_T)$, we have $\Phi(0) = 0$; consequently, $\Phi(ag_\ell) \ge 0$ for $a \ge 0$. From Observation 1, we see that $\Phi(0) = 0$ is achieved by (x, y) such that $x_{s_i} = 1$ for all $i \in [1,p] \setminus \{k\}$ and $x_{s_k} = 0$ for any $k \in [1,p]$ with $a_{s_k} > \lambda_k$, and $\sum_{i=1}^{p} y_{s_i} = u_{s_p} - (a_{s_k} - \lambda_k)$. Since (8) has a slack of $a_{s_k} - \lambda_k$ for this solution for all $a \le \delta_\ell + a_{s_k} - \lambda_k$,

where $\delta_\ell := (u_\ell - u_{s_p})^+$, we have $\Phi(ag_\ell) = 0$ for $0 \le a \le \delta_\ell + a_{s_h} - \lambda_h$, where $h = \text{argmax}_{i\in[1,p]}\{a_{s_i} - \lambda_i\}$.

Then for $a = \delta_\ell + a_{s_h} - \lambda_h$, problem $P_{\Phi(ag_\ell)}$ has an optimal solution $(\bar{x}, \bar{y})$ such that $\sum_{i=1}^{b_h} \bar{y}_{s_i} = u_{s_{b_h}}$, $\bar{y}_{s_i} = a_{s_i}$ for all $i \in [b_h + 1, p] \setminus \{h\}$, and $\bar{y}_{s_h} = 0$, so that $\sum_{i=1}^{p} \bar{y}_{s_i} = u_{s_p} - a_{s_h} + \lambda_h$. Since constraint (8) is tight at this point, given that $\bar{x}_{s_h} = 0$, increasing a beyond $\delta_\ell + a_{s_h} - \lambda_h$ requires reduction in some $\bar{y}_{s_i}$, $i \in [1, p]\setminus\{h\}$, which will increase $\Phi(ag_\ell)$ at the same rate.

If $a_{s_i} \le \lambda_i$ for all $i \in [1, p]\setminus\{h\}$, then $\Phi(ag_\ell) = a - (\delta_\ell + a_{s_h} - \lambda_h)$ for $\delta_\ell + a_{s_h} - \lambda_h < a \le u_\ell$ by reducing $\bar{y}_{s_i}$, $i \in [1, p] \setminus \{h\}$ in increasing order of i; and we are done with characterizing $\Phi(ag_\ell)$. Otherwise, since when $\bar{y}_{s_i}$ reduces to $a_{s_i} - \lambda_i (> 0)$, we can set $y_{s_i} = x_{s_i} = 0$ without changing the objective function, and introduce a slack of $a_{s_i} - \lambda_i$ to constraint (8), the variable to reduce is y_{s_d}, where $d = \text{argmin}_{i\in[1,p]\setminus\{h\}:\ a_{s_i}>\lambda_i}\{\bar{y}_{s_i} - a_{s_i} + \lambda_i\}$ (ties broken by selecting one with the largest a_{s_i}). Thus, $\Phi(ag_\ell)$ will increase by $\bar{y}_{s_d} - a_{s_d} + \lambda_d$ at the rate of one and then stay constant for $a_{s_d} - \lambda_d$ as a increases further. The following Lemma, which is slightly more general than what is needed at this point, shows, by letting $Q = \{h\}$, that $\bar{y}_{s_d} - a_{s_d} + \lambda_d$ equals λ_p.

Lemma 6. *Let $P_{\Phi(ag_\ell)|_Q}$ be the restriction of $P_{\Phi(ag_\ell)}$ with $x_i = y_i = 0$ for all $i \in Q$ for some $Q \subseteq \{i \in [1, p] : a_{s_i} > \lambda_i\}$ with $k \in Q$ and with $a = \delta_\ell + \sum_{i\in Q} a_{s_i} - \lambda_k$. Then $P_{\Phi(ag_\ell)|_Q}$ has an optimal solution (x, y) such that*

$$\min_{i\in[1,p]}\{y_i - a_{s_i} + \lambda_i :\ y_{s_i} > a_{s_i} - \lambda_i > 0\} = \min\left\{\lambda_k, \min_{i\in[k+1,p]}\{\lambda_i :\ y_{s_i} > a_{s_i} - \lambda_i > 0\}\right\}$$

and $\Phi(ag_\ell)|_Q = \sum_{i\in Q\setminus\{k\}} \lambda_i$.

Proof. Since $u_{s_{b_k}} + \sum_{i=b_k+1}^{p} a_{s_i} = u_{s_p} + \lambda_k$, we have $y_{s_i} = a_{s_i}$ for all $i \in [b_k + 1, p] \setminus Q$ and $\sum_{i=1}^{b_k} y_{s_i} = u_{s_{b_k}}$. Suppose $b_k \ge 1$. Then, from Proposition 5, we may assume that $a_{s_i} > y_{s_i} > a_i - \lambda_i > 0$ for at most one $i \in [1, b_k]$.

Suppose $a_{s_j} > y_{s_j} > a_{s_j} - \lambda_j > 0$ for some $j \in [1, b_k]$. Without loss of generality, we may assume that $\sum_{i=1}^{b_j} y_{s_i} = u_{s_{b_j}}$, since if $\sum_{i=1}^{b_j} y_{s_i} < u_{s_{b_j}}$, we can increase $\sum_{i=1}^{b_j} y_{s_i}$ (as from Proposition 5.1 and 5.4, $x_i = 1$ for all $i \le k$ and $Q \subseteq [k + 1, p]$) and decrease y_{s_j} by the same amount until either $\sum_{i=1}^{b_j} y_{s_i} = u_{s_{b_j}}$ holds or $y_{s_j} = a_{s_j} - \lambda_j$ without changing the objective value. Then, since $\lambda_k = u_{s_{b_k}} + \sum_{i=b_k+1}^{p} a_{s_i} - u_{s_p}$ and $\lambda_j = u_{s_{b_j}} + \sum_{i=b_j+1}^{p} a_{s_i} - u_{s_p}$, and $\sum_{i=1}^{b_k} y_{s_i} = u_{s_{b_k}}$ as well, we see that $\sum_{i=b_j+1}^{b_k} y_{s_i} = \lambda_k - \lambda_j + \sum_{i=b_j+1}^{b_k} a_{s_i}$. Therefore, $y_{s_i} = a_{s_i}$ for $i \in [b_j + 1, b_k] \setminus \{j\}$ and $y_{s_j} = a_{s_j} - (\lambda_j - \lambda_k)$. Also since $y_{s_i} \in \{0, a_{s_i}\}$ for all $i \ne j$ such that $a_{s_i} > \lambda_i$ and $\lambda_i \ge \lambda_k$ for all $i \in [1, k - 1]$, the result follows. Furthermore, since $\lambda_j > \lambda_k$, j exists; and we are done when $b_k \ge 1$. Finally, if $b_k = 0$, then $\lambda_i = \lambda_k$ for all $i \in [1, k - 1]$, which

completes the proof of the claim on (x, y). Evaluating the objective function for this point gives $\Phi(ag_\ell)|_Q = \sum_{i\in Q\setminus\{k\}} \lambda_i$. □

From Lemma 6, for $a > \delta_\ell + a_{s_h} - \lambda_h$, $\Phi(ag_\ell)$ increases at the rate of one by reducing y_{s_d} from $\bar{y}_{s_d}$ to $\bar{y}_{s_d} - \lambda_p$ and then reaches to a flat region again because of the slack $a_{s_d} - \lambda_d$ introduced in constraint (8) by setting $y_{s_d} = x_{s_d} = 0$. Let $\alpha_1 := \max_{i\in[1,p]}\{a_{s_i} - \lambda_i\} = a_{s_h} - \lambda_h$ and $\beta_1 := \min_{i\in[1,p]}\{\lambda_i\} = \lambda_p$. Hence

$$\Phi(ag_\ell) = \begin{cases} 0 & \text{if } 0 \le a \le \delta_\ell + \alpha_1, \\ a - \delta_\ell - \alpha_1 & \text{if } \delta_\ell + \alpha_1 \le a \le \delta_\ell + \alpha_1 + \beta_1, \\ \beta_1 & \text{if } \delta_\ell + \alpha_1 + \beta_1 \le a \le \delta_\ell + \alpha_1 + \beta_1 + a_{s_d} - \lambda_d. \end{cases}$$

Example 2. Let $u = (4, 8, 11, 12, 13)$ and $a = (4, 4, 4, 2, 8)$. For $S = \{s_1, s_2, s_3, s_4\} = \{1, 2, 3, 4\}$, we have $\lambda_1 = \lambda_2 = \lambda_3 = 2$ and $\lambda_4 = 1$ and the corresponding bottleneck cover inequality

$$2(1 - x_1) + 2(1 - x_2) + 2(1 - x_3) + 1(1 - x_4) + y_1 + y_2 + y_3 + y_4 \le 12.$$

Let us compute $\Phi(ag_5)$. We have $\delta_5 = (u_5 - u_{s_4})^+ = 1$, $\alpha_1 = \max_{i\in[1,4]}\{a_{s_i} - \lambda_i\} = 2$ and $\beta_1 = \lambda_4 = 1$ (with $d = 4$). Thus

$$\Phi(ag_5) = \begin{cases} 0 & \text{if } 0 \le a \le 3, \\ a - 3 & \text{if } 3 \le a \le 4, \\ 1 & \text{if } 4 \le a \le 5. \end{cases}$$

Unfortunately characterizing $\Phi(ag_\ell)$ for $a > \delta_\ell + \alpha_1 + \beta_1 + a_{s_d} - \lambda_d$ is much harder, because reducing a variable with small λ_i first, without considering $a_{s_i} - \lambda_i$, may not be the best for larger a.

From Lemma 6 we know that, given that a set of variables $\{y_{s_i}\}_{i\in Q}$, $Q\{i \in [1, p] a_{s_i} > \lambda_i\}$ is reduced to zero (in some order) and $a = \delta_\ell + \sum_{i\in Q} a_{s_i} - \lambda_k$; increasing a further, increases $\Phi(ag_\ell)|_Q$ at the same rate by some λ_q, after which $\Phi(ag_\ell)$ will be constant for some $a_{s_t} - \lambda_t$, $q, t \in [1, p]$.

However, Figure 1 illustrates that there is no single sequence in which to reduce y_{s_i} for computing $\Phi(ag_\ell)$ as a increases. Here we plot two upper bounds on $\Phi(ag_\ell)$ obtained by reducing y_i in two different sequences. The plot with dashed line is obtained by reducing y_i in the order 1,2,3,4, whereas the plot with dotted line is obtained when y_i is reduced in the order 1,4,2,3. $\Phi(ag_\ell)$ is the point–wise minimum of these two lines (not drawn in the figure). □

In order to overcome the difficulty illustrated in Example 2, we resort to an easily computable lower bound on $\Phi(ag_\ell)$ by sorting $a_{s_i} - \lambda_i$ in nonincreasing order, and λ_i in nondecreasing order separately. Formally, let α_k be the kth largest $a_{s_i} - \lambda_i$, $i \in [1, p]$ such that $a_{s_i} > \lambda_i$ and β_k be the kth smallest λ_i, $i \in [1, p]$ such that $a_{s_i} > \lambda_i$, and $\alpha_0 = \beta_0 = 0$. Define $\bar{\alpha}_i := \sum_{k=0}^{i} \alpha_k$ and $\bar{\beta}_i := \sum_{k=0}^{i} \beta_k$ for $i \in [0, r]$,

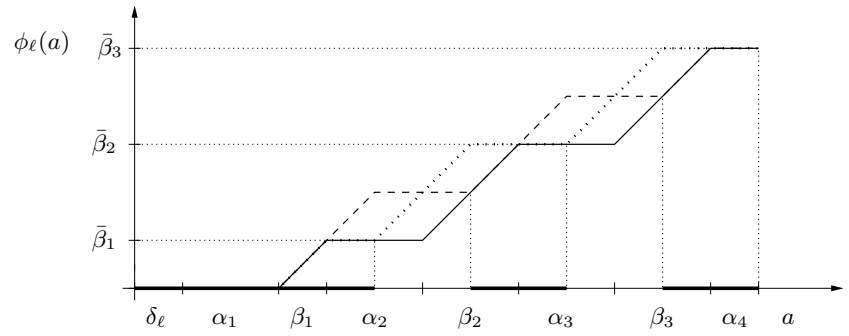

FIGURE 1. Illustration of $\phi_\ell(a)$ for Example 2.

where $r = |\{i \in [1,p] : a_{s_i} > \lambda_i\}|$, and $\phi_\ell : \mathbb{R}_+ \mapsto \mathbb{R}_+$ as

$$\phi_\ell(a) = \begin{cases} 0 & \text{if } 0 \le a \le \delta_\ell, \\ \bar\beta_i & \text{if } \delta_\ell + \bar\alpha_i + \bar\beta_i \le a \le \delta_\ell + \bar\alpha_{i+1} + \bar\beta_i, \quad i \in [0, r-1], \\ a - \delta_\ell - \bar\alpha_i & \text{if } \delta_\ell + \bar\alpha_i + \bar\beta_{i-1} \le a \le \delta_\ell + \bar\alpha_i + \bar\beta_i, \quad i \in [1, r-1], \\ a - \delta_\ell - \bar\alpha_r & \text{if } \delta_\ell + \bar\alpha_r + \bar\beta_{r-1} \le a. \end{cases}$$

Since, by Lemma 6, the plot of $\Phi(ag_\ell)$ has alternating sloped regions of length λ_q and constant regions of length $a_{s_t} - \lambda_t$, lining up longest constant regions and shortest sloped regions as in ϕ_ℓ gives a lower bound on $\Phi(ag_\ell)$.

The solid line in Figure 1 shows $\phi_5(a)$ for Example 2. The bold line segments on the horizontal axis indicate the regions where the $\phi_5(a)$ equals $\Phi(ag_\ell)$, the point–wise minimum of the dashed and dotted lines.

Finally, we consider the case $\ell < s_p$. Notice that by increasing u_ℓ and u_{s_i} for $i \in [1,p]$ with $\ell < s_i < s_p$ to u_{s_p}, we relax the problem $P_{\Phi(ag_\ell)}$. Hence, $\phi_\ell(a)$ with $\delta_\ell := (u_\ell - u_{s_p})^+$ is indeed a lower bound on $\Phi(ag_\ell)$ for $\ell < s_p$ as well. This completes the characterization of a lower bound on $\Phi(ag_\ell)$ for a particular $\ell \in T$; and from the preceding we have the following proposition.

Proposition 7. *Given a facet–defining inequality* (5) *for conv*(F_T), $T \subset [1,n]$

1. $\phi_\ell(a) \le \Phi(ag_\ell)$ *for* $0 \le a \le u_\ell$ *and all* $\ell \in T$,
2. $\phi_\ell(a) = \Phi(ag_\ell)$ *for* $0 \le a \le \delta_\ell + \alpha_1 + \beta_1 + a_{s_d} - \lambda_d$ *and* $\ell \in T \cap [s_p + 1, n]$,
3. $\phi_\ell(a) = \Phi(ag_\ell)$ *for* $0 \le a \le u_\ell$ *and* $\ell \in T \cap [s_p + 1, n]$ *if*
   (*i*) *either* $a_i = a_k$ *for all* $i, k \in S$,
   (*ii*) *or* $\lambda_i = \lambda_k$ *for all* $i, k \in [1,p]$.

Next we show how to obtain valid coefficients (π_i, μ_i) for all $i \in T$ in (6).

Lemma 8. *$\Phi(\sum_{i\in T} y_i g_i) \geq \sum_{i\in T} \phi_i(y_i)$ for all $T \subseteq [1,n] \setminus S$.*

Proof. For any $k \in T$ let $\psi : \mathbb{R} \mapsto \mathbb{R}_+$ be defined as

$$\psi(a) = \begin{cases} \phi_k(a+\delta_k) & \text{if } a > 0 \\ 0 & \text{if } a \leq 0, \end{cases}$$

or equivalently

$$\psi(a) = \begin{cases} 0 & \text{if } a \leq 0 \\ \bar\beta_i & \text{if } \bar\alpha_i + \bar\beta_i \leq a \leq \bar\alpha_{i+1} + \bar\beta_i, \quad i \in [0, r-1], \\ a - \bar\alpha_i & \text{if } \bar\alpha_i + \bar\beta_{i-1} \leq a \leq \bar\alpha_i + \bar\beta_i, \quad i \in [1, r-1], \\ a - \bar\alpha_r & \text{if } \bar\alpha_r + \bar\beta_{r-1} \leq a. \end{cases}$$

Since $\bar\alpha_i$ is partial sum of nonincreasing terms and $\bar\beta_i$ is partial sum of nondecreasing terms, it follows that ψ is superadditive; i.e., $\psi(a_1)+\psi(a_2) \leq \psi(a_1+a_2)$ for $a_1, a_2 \in \mathbb{R}$.

The lemma is trivially true if T is empty or singleton. Suppose it is true for all strict subsets of T and let $\ell = \max\{i : i \in T\}$. Then, since Φ is nondecreasing and the constraint matrix is lower–triangular, we have

$$\begin{aligned} \Phi(\sum_{i\in T} y_i g_i) &\geq \max\left\{ \Phi(\sum_{i\in T\setminus\{\ell\}} y_i g_i),\ \Phi((\sum_{i\in T} y_i) g_\ell) \right\} \\ &\geq \max\left\{ \sum_{i\in T\setminus\{\ell\}} \phi_i(y_i),\ \phi_\ell(\sum_{i\in T} y_i) \right\} \qquad \text{(Proposition 7 and induction)} \\ &= \max\left\{ \sum_{i\in T\setminus\{\ell\}} \psi(y_i - \delta_i),\ \psi(\sum_{i\in T} y_i - \delta_\ell) \right\} \\ &\geq \sum_{i\in T\setminus\{\ell\}} \psi(y_i - \delta_i) + \psi(y_\ell - \delta_\ell) = \sum_{i\in T} \phi_i(y_i). \end{aligned}$$

In order to see the last inequality, observe that if $\sum_{i\in T\setminus\{\ell\}}(y_i - \delta_i) \geq \sum_{i\in T} y_i - \delta_\ell$, we have $y_\ell - \delta_\ell \leq 0$, and hence $\psi(y_\ell - \delta_\ell) = 0$. Otherwise, since ψ is nondecreasing and superadditive, we have $\psi(\sum_{i\in T} y_i - \delta_\ell) \geq \psi(\sum_{i\in T\setminus\{\ell\}}(y_i - \delta_i)) \geq \sum_{i\in T\setminus\{\ell\}} \psi(y_i - \delta_i)$. So $\psi(\sum_{i\in T} y_i - \delta_\ell) \geq \sum_{i\in T\setminus\{\ell\}} \psi(y_i) + \psi(y_\ell - \delta_\ell) \geq \sum_{i\in T\setminus\{\ell\}} \psi(y_i - \delta_i) + \psi(y_\ell - \delta_\ell)$. □

Valid coefficients (π_ℓ, μ_ℓ) for (x_ℓ, y_ℓ) are obtained by ensuring that

$$(H) \qquad \begin{aligned} h(a) = \max\ & \pi_\ell x_\ell + \mu_\ell y_\ell \\ \text{s.t. } & y_\ell = a \\ & 0 \leq y_\ell \leq a_\ell x_\ell,\ x_\ell \in \{0,1\} \end{aligned}$$

is no more than $\phi_\ell(a)$ and equals $\phi_\ell(a)$ for two linearly independent solutions, as suggested in Gu et al. [9]. It is seen above that $h(a) = \pi_\ell + \mu_\ell a$ for $a > 0$. Thus π_ℓ and μ_ℓ are intercept and slope of an affine function that supports $\phi_\ell(a)$ at two linearly independent solutions of (H). Defining $\gamma_i = \delta_\ell + \bar{\beta}_i + \bar{\alpha}_{i+1}$, it is easy to verify that $(\pi_\ell, \mu_\ell) \in H_\ell = \{(0,0)\} \cup H_\ell^1 \cup H_\ell^2$, where

$$H_\ell^1 = \left\{ (\bar{\beta}_{i-1} - \frac{\beta_i \gamma_{i-1}}{\beta_i + \alpha_{i+1}}, \frac{\beta_i}{\beta_i + \alpha_{i+1}}) : a_\ell \geq \gamma_i,\ i \in [1, |S| - 1] \right\} \text{ and}$$

$$H_\ell^2 = \left\{ \begin{array}{ll} (-\delta_\ell - \bar{\alpha}_i, 1) & \text{if } \gamma_{i-1} < a_\ell \leq \gamma_{i-1} + \beta_i,\ i \in [1, |S|], \\ (\bar{\beta}_{i-1} - \frac{\beta_i \gamma_{i-1}}{a_\ell - \gamma_{i-1}}, \frac{\beta_i}{a_\ell - \gamma_{i-1}}) & \text{if } \gamma_{i-1} + \beta_i < a_\ell < \gamma_i,\ i \in [1, |S| - 1] \end{array} \right\}.$$

Theorem 9. *Inequality* (6) *with* $(\pi_i, \mu_i) \in H_i$ *for* $i \in T$, *where* S, $T \subset [1, n]$ *and* $S \cap T = \emptyset$ *is valid for* F. *It defines a facet of* $conv(F)$ *if inequality* (5) *defines a facet of* $conv(F_T)$, $T \subseteq [s_p + 1, n]$, *and* $a_i \leq \delta_i + \alpha_1 + \beta_1$ *for all* $i \in T$.

Proof. The validity of (6) is a consequence of (7), which follows from Lemma 8 and that $\pi_i + \mu_i a \leq \phi_i(a)$ for $a \geq 0$ when $(\pi_i, \mu_i) \in H_i$, $i \in T$. The facet condition follows from the fact that $\phi_i(a) = \Phi(a g_i)$ for $0 \leq a \leq \delta_i + \alpha_1 + \beta_1$ and $i > s_p$, and that $\pi_i + \mu_i a$ supports $\phi_i(a)$ at two linearly independent solutions of (H) for all $i \in T$. □

Example 1 (cont.) Earlier we have seen that for $S = \{s_1, s_2\} = \{2, 3\}$, inequality

$$4(1 - x_2) + 3(1 - x_3) + y_2 + y_3 \leq 11$$

defines a facet of $conv(F)$. Lifting it with (x_i, y_i), $i \in T = \{1, 4\}$, we have $\alpha_1 = 4$, $\beta_1 = 4$, $\alpha_2 = 3$, $\delta_1 = 0$, and $\delta_4 = 2$. Hence, $H_1^1 = \emptyset$, $H_1^2 = (1, -4)$, $H_2^1 = \emptyset$, $H_2^2 = (\frac{2}{3}, -4)$; consequently, we obtain

$$-4x_1 + 4(1 - x_2) + 3(1 - x_3) - 4x_4 + y_1 + y_2 + y_3 + \frac{2}{3} y_4 \leq 11.$$

This inequality and similar ones

$$-4x_1 + 4(1 - x_2) + 3(1 - x_3) + y_1 + y_2 + y_3 \leq 11,$$

$$4(1 - x_2) + 3(1 - x_3) - 4x_4 + y_2 + y_3 + \frac{2}{3} y_4 \leq 11,$$

by taking $T = \{1\}$ and $T = \{4\}$ are easily verified to be facet–defining for $conv(F)$. The inequality with $T = \{1\}$ illustrates that lifted inequalities (6) may define facets also when $T \not\subseteq [s_p + 1, n]$. □

3. Special cases

3.1. **Uncapacitated case.** We obtain the uncapacitated lot–sizing polytope when $a_i = u_i$ for all $i \in [1, n]$. In this case, inequalities (6) for $S = [1, \ell],\ \ell \in [0, n-1]$, and $T \subseteq [\ell+1, n]$ reduce to

$$\sum_{i=1}^{\ell} y_i + (u_\ell - u_{\ell-1})(1 - x_\ell) + \sum_{i \in T} y_i \leq u_\ell + \sum_{i \in T} (u_i - u_{\ell-1}) x_i,$$

or

$$\sum_{i=1}^{\ell-1} y_i + \sum_{i \in T \cup \{\ell\}} y_i \leq u_{\ell-1} + \sum_{i \in T \cup \{\ell\}} (u_i - u_{\ell-1}) x_i,$$

which are equivalent to the uncapacitated lot–sizing inequalities

$$\sum_{t \in T} w_t \leq \sum_{t \in T} d_{t\ell} z_t + i_\ell$$

for $T \subseteq [1, \ell],\ \ell \in [1, n]$. These inequalities are sufficient to describe the lot–sizing polytope in the uncapacitated case and can be separated in polynomial time [2].

3.2. **Constant capacity case.** When the capacities are constant; i.e., $a_i = a$ for all $i \in [1, n]$, inequalities (2) reduce to

$$\sum_{i=1}^{p} \min\{a, (u_{s_p} - u_{s_{b_i}} - (p - b_i - 1)a)^+\}(1 - x_{s_i}) + \sum_{i=1}^{p} y_{s_i} \leq u_{s_p} \tag{9}$$

for $S = \{s_1, s_2, \ldots, s_p\} \subseteq [1, n]$. Observe from Theorem 4 that for the constant capacity case, as $\alpha_1 := \max_{k \in [1,p]}\{a_{s_k} - \lambda_k\} = u_{s_p} - u_{s_{b_p}} - (p - b_p - 1)a$, inequalities (6) with $(\pi_i, \mu_i) = (u_{s_{b_p}} + (p - b_p - 1)a - u_i, 1),\ i \in T \subseteq [s_p + 1, n]$ such that $u_i - u_{s_p} < (p - b_p)a$ define facets of $conv(F)$ whenever (9) defines a facet of $conv(F_T)$, because $\phi_i(a) = \Phi(ag_i)$ as $a \leq \alpha_1 + \beta_1$.

We show here that these inequalities can be separated in polynomial time. Note that they form a strong, strict subset of the (k, ℓ, S, I) inequalities given by Pochet and Wolsey [16], for which no polynomial–time separation algorithm is known.

First we describe the separation algorithm for inequalities (9) and then extend it for the lifted inequalities. Given a point (x, y) to separate, for each fixed $s_p \in [1, n]$, we define a directed network on which a longest path corresponds to an inequality (9) with the largest left–hand–side value for (x, y). Let $N = (V, A)$ be an acyclic directed graph, where each vertex in V is a triple (t, d, r) such that $t \in [0, s_p]$; $d \in [0, t-1]$ if $t \in [1, s_p - 1]$ (d is undefined if $t = 0$); and $r \in [0, s_p - t]$. For a vertex (t, d, r), t denotes an element that may possibly be included in S, d denotes a possible bottleneck for t, and r denotes $|\{s_i \in S : s_i > t\}|$. A tuple $((t_i, d_i, r_i), (t_k, d_k, r_k))$, or simply (i, k), is an arc in A if and only if it satisfies $t_i < t_k$, $r_i = r_k + 1$, and $d_k = t_i$ or $d_k = d_{t_i}$.

Thus N is an acyclic graph with source vertices $\{(0,-,0),(0,-,1),\ldots,(0,-,s_p)\}$ and sink vertices $\{(s_p,0,0),(s_p,1,0),\ldots,(s_p,s_p-1,0)\}$. The set of vertices on a path from a source vertex to a sink vertex represents S and an arc (i,k) on such a path denotes that t_i and t_k are two consecutive elements in S. Observe that in (1) the bottleneck of t_k is either t_i or the bottleneck of t_i; therefore, G, with $O(n^4)$ arcs, can be built easily with a forward pass from the sources to the sinks.

Next we assign lengths on the arcs. Given a point (x,y) to separate, the length of arc $(i,k)\in A$ equals the contribution of (x_k,y_k) to the left hand side of inequality (9). In order to do that we define the length of an arc (i,k) as

$$c_{ik}=\begin{cases} y_{t_k}+\min\{a,(u_{s_p}-r_k a)^+\}(1-x_{t_k}) & \text{if } t_i=0,\\ y_{t_k}+\min\{a,(u_{s_p}-u_{b_i}-(\texttt{pos[i]}-\texttt{pos[d}_\texttt{k}\texttt{]}+r_k)a)^+\}(1-x_{t_k}) & \text{if } t_i\geq 1,\end{cases}$$

where pos[i] is the number of vertices in a longest path from any source vertex to vertex i. Since the longest path algorithm on an acyclic directed network proceeds in topological ordering of the vertices (see for instance [1]), pos[i] and pos[$\texttt{d}_\texttt{k}$] are determined before the arc (i,k) is used in the algorithm. Therefore c_{ik} is computed when running the longest path algorithm as it is needed. Given a longest path with arc (i,k), since pos[i] refers to the position of t_i in S, $p=|S|$, and $r_k=|\{s_i\in S: s_i>t_k\}|$, we see that $\texttt{pos[i]}-\texttt{pos[d}_\texttt{k}\texttt{]}+r_k=p-b_{pos[k]}-1$. Hence the length of a longest path from a source vertex to a sink vertex $(s_p,i,0)$, $i\in[0,p-1]$, all of which can be computed simultaneously in $O(n^4)$, equals the maximum left hand side value for any inequality (9) under the assumption that i is the bottleneck of p. Longest of these s_p paths gives the desired inequality (9).

Now it is easy to extend this algorithm to find a most violated *lifted* bottleneck cover inequality. We augment N with a super sink vertex ν and an arc from each sink vertex $(s_p,i,0)$ to ν with length equal to

$$\sum_{k\in T_i}(\bar{y}_k-\min\{a,u_{s_i}+(p-b_p-1)a-u_k\}\bar{x}_k), \tag{10}$$

where $T_i=\{k\in[s_p+1,n]:\min\{a,u_{s_i}+(p-b_p-1)a-u_k\}\bar{x}_k<\bar{y}_k\}$. Note that T_i is the index set of variables (x_i,y_i) that has a positive contribution to the left hand side of the inequality, given that i is the bottleneck of s_p and the summand (10) can be computed in linear time. Therefore a longest path from a source to the super sink ν gives a lifted bottleneck cover inequality with the largest left hand side value. Hence for the constant capacity case, separation problem for the lifted bottleneck cover inequalities (6) with $T\subseteq[s_p+1,n]$ can be solved in $O(n^5)$ by running the linear–time longest path algorithm for each $s_p\in[1,n]$.

4. Computations

In this section we describe our computational experience on using the inequalities introduced in Section 2 as cutting planes when solving lot–sizing problems with a branch–and–cut algorithm. All experiments are done on a 2GHz Intel Pentium4/Linux workstation with 1GB RAM using the callable libraries of CPLEX[1] Version 8.1 Beta with one hour time limit.

For our experiments we created a data set of capacitated lot–sizing instances with varying cost and capacity characteristics. Our preliminary experience has shown that two main characteristics play a major role in influencing the integrality gap and hence the difficulty of solving the problem instances. The first one is the tightness of the capacities with respect to the demand. The second one is the ratio between the setup cost and the inventory holding cost. Therefore, the instances are generated for varying average capacity/demand ratios $\mathtt{c} \in \{3, 6, 9, 12\}$ and setup/holding cost ratios $\mathtt{f} \in \{100, 200, 500\}$. Capacity c_t is drawn from integer uniform $[0.75\mathtt{c}\bar{d}, 1.25\mathtt{c}\bar{d}]$, setup cost s_t is drawn from integer uniform $[0.90\mathtt{f}\bar{h}, 1.10\mathtt{f}\bar{h}]$, where $\bar{d}$ and $\bar{h}$ are the averages for demand and holding cost. For all instances h_t equals 10, and p_t and d_t are drawn from uniform integer [81,119] and [1,19], respectively. Here we report a summary of the solution statistics for instances with number of time periods 60, 90, 120, and 365.

We use a heuristic separation algorithm in order to find violated cutting planes. Given a fractional solution (x, y), for each $j \in [1, n]$ we sequentially let $S = [1, j]$, $S = \{i \in [1, j] : x_i > 0\}$, and $S = \{i \in [1, j] : 1 > x_i > 0\}$ and then find $T \subseteq [1, n] \setminus S$ that maximizes the left–hand–side value for (6) for each such S. Observe that given S, finding $T \subseteq [1, n] \setminus S$ that maximizes $\sum_{i \in T}(\pi_i x_i + \mu_i y_i)$, where $(\pi_i, \mu_i) \in H_i$, can be done easily by testing $\pi_i x_i + \mu_i y_i > 0$ for $(\pi_i, \mu_i) \in H_i$ separately for each $i \in [1, n] \setminus S$. If the inequality with $T = \emptyset$ is violated by (x, y), it is added to the formulation also.

Our experiments are summarized in Tables 1–4. On every two consecutive lines in these tables, we report the statistics from experiments with default CPLEX and with default CPLEX plus the lifted bottleneck cover cuts (6) and a combination of parameters `c` and `f`. Each entry in the tables corresponds to the average for 5 random instances. The first line is with default CPLEX, the second one is with the lifted bottleneck cover cuts. Default CPLEX MIP solver adds several classes of generic cutting planes, including the flow cover inequalities mentioned in Remark 1, to the formulation.

In these tables we report the averages for the percentage integrality gap of the formulation before cuts are added ($\mathtt{initgap} = 100 \times (\mathtt{bestub} - \mathtt{initlb})/\mathtt{bestub}$), the percentage improvement in the integrality gap after adding cuts before branching ($\mathtt{gapimp} = 100 \times (\mathtt{rootlb} - \mathtt{initlb})/(\mathtt{bestub} - \mathtt{initlb})$), and the percentage gap

[1]CPLEX is a trademark of ILOG, Inc.

c	f	initgap	gapimp	endgap	cuts	nodes	time
3	100	5.93	80.39	0.00	35	1613	2
			90.72	0.00	178	99	1
3	200	5.77	65.28	0.00	21	16477	12
			76.08	0.00	568	1533	14
3	500	4.65	54.35	0.00	264	232745	386
			66.14	0.00	1406	7872	144
6	100	10.29	92.26	0.00	32	64	0
			99.64	0.00	78	0	0
6	200	11.79	77.46	0.00	28	1681	2
			97.33	0.00	150	22	1
6	500	11.00	61.86	0.00	79	10486	9
			82.98	0.00	574	855	10
9	100	13.86	92.78	0.00	46	120	0
			99.72	0.00	87	0	0
9	200	16.59	65.10	0.00	20	10765	7
			99.26	0.00	86	2	0
9	500	17.01	57.09	0.00	42	8445	6
			93.77	0.00	277	74	2
12	100	15.68	91.46	0.00	51	342	1
			99.95	0.00	98	0	0
12	200	19.44	67.10	0.00	44	20894	14
			99.64	0.00	108	0	0
12	500	20.83	52.01	0.00	18	9630	6
			98.13	0.00	134	19	1

TABLE 1. 60 time periods

between the lowest upper bound and the highest lower bound on the optimal value at termination (`endgap` $= 100 \times$ (`bestub` $-$ `bestlb`)/`bestub`), where `initlb, rootlb, bestlb, bestub` are the objective function values of the initial LP relaxation, LP relaxation after all cuts are added before branching, the lowest lower bound for all unexplored nodes of the search tree, and finally the best feasible solution. We report also the averages for the number of cuts added in the search tree (`cuts`), the number of nodes explored (`nodes`) and the elapsed CPU time in seconds (`time`). We report the number of unsolved instances on the right of `engdap` in brackets if some of the instances of an experiment could not be solved to optimality within the one hour time limit. Consequently, we infer, for instance, that the `endgap` for the unsolved instance in group `c` $= 12$ of Table 2 is 2.15%.

c	f	initgap	gapimp	endgap	cuts	nodes	time
3	100	6.11	72.18	0.00	32	547439	775
			86.92	0.00	536	1882	35
3	200	5.74	64.86	0.00	17	746093	1013
			77.67	0.00	1040	9022	183
3	500	4.28	64.76	0.00	22	162234	187
			74.34	0.00	2178	13831	470
6	100	10.75	88.36	0.00	52	1556	3
			99.25	0.00	132	11	1
6	200	11.81	69.64	0.00	18	216147	266
			94.14	0.00	317	215	5
6	500	10.97	57.99	0.00	12	633771	731
			81.77	0.00	1990	12643	621
9	100	14.11	90.76	0.00	77	1826	4
			99.78	0.00	140	2	1
9	200	16.41	77.03	0.21[1]	92	681471	735
			99.77	0.00	135	0	1
9	500	16.75	54.08	0.23[1]	20	918048	915
			96.15	0.00	480	149	7
12	100	16.05	89.02	0.00	102	5805	14
			99.88	0.00	152	0	1
12	200	19.70	66.29	0.43[1]	76	1653136	1775
			99.91	0.00	161	0	1
12	500	21.39	45.56	0.00	17	1363485	1513
			98.83	0.00	229	25	3

TABLE 2. 90 time periods

An immediate observation that can be made from the Tables 1–4 is that the initial integrality gap increases with capacity. On the other hand, as setup/holding cost ratio increases, it decreases for tight capacities, but increases for high capacities.

The lifted bottleneck cover inequalities reduce the integrality gaps at the root node of the search tree quite favorably compared with CPLEX cuts. The gap improvement increases as the capacities get higher, which is an expected result since the facets of the uncapacitated case is defined completely by a special case of the cuts as explained in Section 3. The gap improvement decreases as the setup/holding cost ratio increases. On the average over about 70% of the integrality gap is closed by CPLEX cuts, where as this improvement increases to over 90% when lifted bottleneck flow cover cuts are added as well.

c	f	initgap	gapimp	endgap	cuts	nodes	time
3	100	5.94	79.59	0.00	85	352212	1046
			91.30	0.00	588	959	37
3	200	5.90	64.05	0.75[5]	58	1971093	3613
			74.79	0.25[2]	2392	63839	3157
3	500	4.42	59.26	0.49[4]	36	2068223	3456
			64.70	0.57[3]	3710	56611	3337
6	100	11.07	87.19	0.00	84	88726	252
			98.89	0.00	184	11	2
6	200	12.06	69.64	0.52[4]	59	1531901	3367
			93.57	0.00	664	797	29
6	500	11.16	54.03	1.86[5]	21	2686541	3617
			83.32	0.11[1]	3110	25055	1646
9	100	14.37	89.61	0.00	110	20373	70
			99.67	0.00	177	1	1
9	200	16.55	64.77	1.53[4]	70	2124499	3091
			99.50	0.00	209	6	2
9	500	16.32	53.14	1.52[4]	17	2424953	3318
			96.52	0.00	543	264	15
12	100	16.35	86.12	0.00	124	205615	682
			99.89	0.00	212	0	2
12	200	19.75	65.67	2.19[4]	132	1878952	3113
			99.92	0.00	241	2	3
12	500	21.25	44.02	4.07[5]	15	3403311	3620
			98.95	0.00	326	26	6

TABLE 3. 120 time periods

The improvement in the integrality gaps has profound effect in reducing the number of nodes explored in the search tree and elapsed CPU time. All instances up to 90 time periods are solved to optimality within the one hour time limit. For instances that are not solved to optimality, the maximum optimality gap at termination is under 1% for instances up to 120 time periods and under 3% for instances with 365 periods. Average optimality gap at termination over all instances is 0.18%.

All instances with mean capacity/demand ratio 9 or larger are solved very easily in a few nodes. The instances with very tight capacities and high setup costs are the hardest ones to solve with our cuts. Even though they have smaller integrality gaps, the percentage gap improvement by the cuts is smallest for them.

c	f	initgap	gapimp	endgap	cuts	nodes	time
3	100	6.29	70.42	1.69[5]	190	413932	3610
			86.68	0.62[5]	1916	10518	3630
3	200	6.38	52.22	3.11[5]	77	543234	3610
			70.39	1.69[5]	3909	8095	3689
3	500	5.30	40.99	3.17[5]	20	780071	3610
			51.70	2.39[5]	5588	7589	3760
6	100	11.28	87.82	0.90[5]	261	382649	3609
			99.01	0.00	670	148	83
6	200	12.35	66.35	3.79[5]	127	484482	3609
			94.67	0.28[5]	2296	11691	3639
6	500	11.94	66.35	3.79[5]	127	484482	3609
			77.71	2.35[5]	5273	7023	3779
9	100	14.77	90.05	0.90[5]	321	371756	3610
			99.93	0.00	596	1	57
9	200	17.49	76.00	3.48[5]	345	295057	3608
			99.52	0.00	705	43	64
9	500	17.77	44.46	10.07[5]	43	722353	3610
			94.92	0.45[4]	4050	7915	3569
12	100	16.97	89.24	1.23[5]	312	373887	3609
			99.92	0.00	638	1	82
12	200	20.92	72.61	4.86[5]	387	358506	3609
			99.89	0.00	692	2	94
12	500	22.89	38.14	14.07[5]	21	788805	3610
			99.18	0.00	1129	249	195

TABLE 4. 365 time periods

5. Conclusions

We identified facets of the lot–sizing polytope using its bottleneck structure. These facets are then generalized by simultaneous lifting with pairs of variables.

The computational experiments with the lifted bottleneck cover inequalities suggest that they are quite effective in solving lot–sizing problems when used as cutting planes. One may pursue several directions for improving the computations further. The separation heuristic used for finding violated cuts in our computations is quite simple. Other construction and exchange heuristics may be developed to find more violated cuts. Identifying stronger lower bounds on the lifting function Φ than ϕ would reduce the integrality gap further.

A complete description of the lot–sizing polytope for the constant capacity is unknown. Investigation of Φ for the constant capacity special case merits attention. Preliminary observations in this direction indicate that the constant–capacity lot–sizing polytope has facets that are not described by $(k\ell SI)$ inequalities [16]; but are much harder to identify explicitly than the ones listed in this paper.

The bottleneck cover inequalities and adaptations of them may have the potential of speeding up computations for more complicated production planning problems that contain the lot–sizing problem as a substructure.

Acknowledgments

We are grateful to Ed Rothberg for making changes in the CPLEX code that allowed us to add a large number of cuts efficiently and reduced the computation times significantly.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs NJ, 1993.

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

[3] G. Belvaux and L. A. Wolsey. bc–prod: a specialized branch–and–cut system for lot–sizing problems. *Management Science*, 46:724–738, 2000.

[4] G. Belvaux and L. A. Wolsey. Modelling practical lot-sizing problems as mixed integer programs. *Management Science*, 47:993–1007, 2001.

[5] G. R. Bitran and H. H. Yanasse. Computational complexity of the capacitated lot size problem. *Management Science*, 28:1174–1186, 1982.

[6] A. Federgruen and M. Tzur. A simple forward algorithm to solve general dynamic lot sizing models with n periods in $O(n\,logn)$ or $O(n)$ time. *Management Science*, 37:909–925, 1991.

[7] M. Florian and M. Klein. Deterministic production planning with concave costs and capacity constraints. *Management Science*, 18:12–20, 1971.

[8] M. Florian, J. K. Lenstra, and H. G. Rinnooy Kan. Deterministic production planning: Algorithms and complexity. *Management Science*, 26:669–679, 1980.

[9] Z. Gu, G. L. Nemhauser, and M. W. P. Savelsbergh. Sequence independent lifting in mixed integer programming. *Journal of Combinatorial Optimization*, 4:109–129, 2000.

[10] J. M. Y. Leung, T. L. Magnanti, and R. Vachani. Facets and algorithms for capacitated lot sizing. *Mathematical Programming*, 45:331–359, 1989.

[11] M. Loparic, H. Marchand, and L. A. Wolsey. Dynamic knapsack sets and capacitated lot–sizing. *Mathematical Programming*, xx:xx–xx, 2002.

[12] A. Miller, G. L. Nemhauser, and M. W. P. Savelsbergh. On the capacitated lot–sizing and continuous 0–1 knapsack polyhedra. *European Journal of Operational Research*, 125:298–315, 2000.

[13] M. W. Padberg, T. J. Van Roy, and L. A. Wolsey. Valid linear inequalities for fixed charge problems. *Operations Research*, 32:842–861, 1984.

[14] Y. Pochet. Valid inequalities and separation for capacitated economic lot sizing. *Operations Research Letters*, 7:109–115, 1988.

[15] Y. Pochet and L. A. Wolsey. Solving multi–item lot–sizing problems using strong cutting planes. *Management Science*, 37:53–67, 1991.
[16] Y. Pochet and L. A. Wolsey. Lot–sizing with constant batches: Formulation and valid inequalities. *Mathematics of Operations Research*, 18:767–785, 1993.
[17] C. P. M. Van Hoesel and A. P. M. Wagelmans. An $O(T^3)$ algorithm for the economic lot-sizing problem with constant capacities. *Management Science*, 42:142–150, 1996.
[18] A. Wagelmans, S. Van Hoesel, and A. Kolen. Economic lot sizing: An $O(n \log n)$ algorithm that runs in linear time in the Wagner-Whitin case. *Operations Research*, 40:S145–S156, 1992.
[19] H. M. Wagner and T. M. Whitin. Dynamic version of the economic lot size model. *Management Science*, 5:89–96, 1958.
[20] L. A. Wolsey. Submodularity and valid inequalities in capacitated fixed charge networks. *Operations Research Letters*, 8:119–124, 1989.
[21] L. A. Wolsey. Solving multi–item lot–sizing problems with an MIP solver using classification and reformulation. CORE Discussion Paper 2002/12, 2002.