Math. Program., Ser. A 99: 443–465 (2004)

Digital Object Identifier (DOI) 10.1007/s10107-003-0465-8

Alper Atamtürk · Juan Carlos Muñoz

A study of the lot–sizing polytope

Received: October 8, 2002 / Accepted: June 25, 2003
Published online: August 18, 2003 –

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 computational experiments on solving the lot–sizing problem with varying cost and capacity characteristics.

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 (Florian et al. [9]). Several special cases, including the uncapacitated and constant–capacity cases, of the problem are solved in polynomial time; see Bitran and Yanasse [5], Federgruen and Tzur [7], Florian and Klein [8], van Hoesel and Wagelmans [20], Wagelmans et al. [21], and Wagner and Whitin [22].

Many practical multi–item, multi–stage production planning problems over a finite discrete–time horizon contain the lot–sizing problem as a substructure. Strong valid inequalities and reformulations for the lot–sizing problem often form the basis of branch–and–cut algorithms and effective models for those more complicated problems; see, for instance, Belvaux and Wolsey [3, 4], Pochet and Wolsey [17], and Wolsey [24]. Therefore, a good understanding of the lot–sizing polytope has immediate implications for many practical production problems.

A complete linear description of the uncapacitated lot–sizing polytope is given by Barany et al. [2]. The constant–capacity lot–sizing polytope is studied by Leung et al. [11] and Pochet and Wolsey [18]. For the general case with no restrictions on capacities Pochet [16] gives valid inequalities for LSP based on surrogate single node flow relaxations, Miller et al. [14] describe inequalities from its continuous knapsack relaxations. Loparic et al. [12] study a dynamic knapsack set relaxation of LSP.

A. Atamtürk: Department of Industrial Engineering and Operations Research, University of California, 4135 Etcheverry Hall, Berkeley CA, 94720–1777 USA. e-mail: atamturk@ieor.berkeley.edu

Supported, in part, by NSF Grants 0070127 and 0218265, and a grant from ILOG, Inc.

J. C. Muñoz: Department of Civil and Environmental Engineering, University of California, 416G McLaughlin Hall, Berkeley CA, 94720–1720 USA. Currently in Pontificia Universidad Católica de Chile.
e-mail: jcm@ing.puc.cl

Here we study the lot–sizing polytope directly. In particular, we define a notion of *bottleneck covers* and show the correspondence between bottleneck covers and the fractional vertices of the linear programming relaxation of the lot–sizing problem. We identify facet–defining inequalities that cut off all of the fractional vertices of the linear programming relaxation as well as liftings from those facets.

Throughout we let $[i, k] := \{j \in \mathbb{Z} : i \leq j \leq k\}$, $a^+ := \max\{a, 0\}$, and e_i denote the ith unit vector. Let p_t, h_t, and s_t denote the production, holding, and setup costs in period t, and w_t, i_t, and z_t denote the production, incoming inventory, and setup variables in period $t \in [1, n]$, respectively. Then LSP can be formulated as

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

where

$$E := \left\{ (z, w, i) \in \{0, 1\}^n \times \mathbb{R}^n_+ \times \mathbb{R}^{n+1}_+ : \begin{array}{r} i_t + w_t - i_{t+1} = d_t \quad t \in [1, n] \\ w_t \leq c_t z_t \; t \in [1, n] \\ i_{n+1} = 0 \end{array} \right\}$$

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 $i_t \geq 0$ and reindexing the variables in the reverse order, gives the following equivalent *bottleneck flow model* of the feasible solutions of LSP:

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

where $a_i = c_{n-i+1}$ and $u_i = \sum_{j=1}^{i} d_{n-j+1}$ for $i \in [1, n]$, that is, $u_1 = d_n$, $u_2 = d_n + d_{n-1}$, and so on. 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.

Section 2 is devoted to the analysis of the convex hull of F. The results are specialized for the uncapacitated and constant–capacity cases in Section 3. The computational experiments with the new inequalities when used as cutting planes are described in Section 4. We conclude with Section 5.

2. Polyhedral analysis

Assumptions. *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:*

(A1) $0 < a_i \leq u_i$ *for all* $i \in [1, n]$,

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

If $a_i \leq 0$, then (x_i, y_i) can be dropped. If $u_i < a_i$, then a_i can be reduced to 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. $y_i \geq 0$*,* $x_i \leq 1$*,* $y_i \leq a_i x_i$*,* $i \in [1, n]$*, define facets of* $conv(F)$*,*
3. $y_1 + \cdots + y_i \leq u_i$*,* $i \in [2, n]$*, defines a facet of* $conv(F)$ *if and only if* $u_i \leq u_{k-1} + \sum_{j=k+1}^{i} a_j$ *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 $y_1 + \cdots + y_i \leq u_i$. Else, $y_1 + \cdots + y_i + y_{i+1} \leq u_i$ 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: $y_j^k > 0$ and $x_j^k = 1$ for $j \in [1, k-1]$ such that $\sum_{j=1}^{k-1} y_j^k = u_{k-1}$; $y_k^k = 0$ and $x_k^k \in \{0, 1\}$; and $y_j^k > 0$ and $x_j^k = 1$ such that $\sum_{j=k+1}^{i} y_j^k = u_i - u_{k-1}$; and $y_j^k = x_j^k = 0$ for $j \in [i+1, n]$ and $k \in [1, i]$. The remaining $2(n-i)$ points are (e_k, y^i) and $(e_k, y^i + \min\{a_k, u_{i+1} - u_i\} e_k)$ 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 $[1, n]$ such that $s_1 < s_2 < \cdots < s_p$ and, for simplicity of notation, let $s_0 = 0$. The following min–max relationship, which holds due to the lower–triangular structure of the first set of constraints of F, is the key for understanding the bottleneck structure of $conv(F)$

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

Definition 1. *The smallest minimizer in* (1) *is called the* bottleneck of the set S *and is denoted as* $b_S (\in [0, p],\ b_\emptyset = 0)$*. The index set* $B_S := \{b_{S_i} : i \in [1, p]\}$ *is called the* bottleneck set of S*, where* $S_i := \{s_1, s_2, \ldots, s_i\}$ *for* $i \in [1, p]$*. For a given* S *and* $i \in [1, p]$*, 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 $b_4 = 3$. □

Proposition 2. *If* p *is the bottleneck of* $S = \{s_1, s_2, \ldots, s_p\}$*, i.e.,* $b_S = p$*, then*

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

Proof. The proposition follows from the min–max relationship (1).

1. If $b_{S\setminus\{s_k\}} > k$, then

$$\zeta(S \setminus \{s_k\}) = \min_{k+1 \le i \le p} \left\{ u_{s_i} + \sum_{t=i+1}^{p} a_{s_t} \right\} = u_{s_p} \text{ (by assumption).}$$

If $b_{S\setminus\{s_k\}} < k$, then

$$\zeta(S \setminus \{s_k\}) = \min_{0 \le i \le k-1} \left\{ u_{s_i} + \sum_{t=i+1}^{p} a_{s_t} - a_k \right\} = \zeta(S_{k-1}) + \sum_{t=k+1}^{p} a_{s_t}.$$

2. Suppose that $s_j < s_k < s_{j+1}$ for $s_j, s_{j+1} \in S$. Then

$$\zeta(S \cup \{s_k\}) = \min \left\{ \zeta(S_j) + a_{s_k} + \sum_{t=j+1}^{p} a_{s_t}, u_{s_k} + \sum_{t=j+1}^{p} a_{s_t}, \zeta(S) \right\} = u_{s_p}$$

The last equation follows from $u_{s_k} + \sum_{t=j+1}^{p} a_{s_t} \ge u_{s_j} + \sum_{t=j+1}^{p} a_{s_t} \ge \zeta(S) = u_{s_p}$. On the other hand, if $s_k > s_p$, $\zeta(S \cup \{s_k\}) = \min\left\{\zeta(S) + a_{s_k}, u_{s_k}\right\}$. □

2.2. *Bottleneck covers*

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

Proposition 3. *For any $S = \{s_1, s_2, \dots, s_p\} \subseteq [1, n]$ the following statements hold:*

1. $\lambda_i \ge \lambda_k$ for $i, k \in [1, p]$ such that $i < k$,
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 $b_S = p$.

Proof. The proof follows from the definition that b_k is the bottleneck of S_{k-1}.

1. For $i, k \in [1, p]$ such that $i < k$

$$u_{s_{b_k}} + \sum_{j=b_k+1}^{k-1} a_{s_j} = \zeta(S_{k-1}) \le \zeta(S_{i-1}) + \sum_{j=i}^{k-1} a_{s_j} = u_{s_{b_i}} + \sum_{j=b_i+1}^{k-1} a_{s_j},$$

which gives $\lambda_i \ge \lambda_k$.

2. $b_i = b_k$ implies $\lambda_i = \lambda_k$ from the definition of λ_i. Now suppose $b_i < b_k$ for $i < k$. Then $u_{s_{b_i}} + \sum_{j=b_i+1}^{k-1} a_{s_j} < u_{s_{b_k}} + \sum_{j=b_k+1}^{k-1} a_{s_j}$, since b_k is the smallest minimizer, or equivalently, $\lambda_i > \lambda_k$.

3. If $\lambda_i = \lambda_{i+1}$, from part *2*, we have $b_i = b_{i+1}$, but $b_i < i$. If $\lambda_i > \lambda_{i+1}$, from part *2*, $b_i < b_{i+1}$ (as $b_i \le b_{i+1}$ in general). We also have

$$\zeta(S_i) = \min\left\{\zeta(S_{i-1}) + a_{s_i}, u_{s_i}\right\},$$

but $b_i < b_{i+1}$ implies that $\zeta(S_{i-1}) + a_{s_i} > u_{s_i} = \zeta(S_i)$, which gives $b_{i+1} = i$.

4. $b_S = p$ if and only if $u_{s_p} < \zeta(S_{p-1}) + a_{s_p}$ if and only if $\lambda_p > 0$. □

For a bottleneck cover $S = \{s_1, s_2, \dots, s_p\}$, let the *bottleneck cover inequality* be defined as

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

Observe that if p is the bottleneck of S, or equivalently, $\lambda_p > 0$, then $\zeta(S) = u_{s_p}$; consequently, when $x_{s_i} = 0$, the slack of the constraint $\sum_{i=1}^{p} y_{s_i} \le u_{s_p}$ is at least $\zeta(S) - \zeta(S \setminus \{s_i\}) = (a_{s_i} - \lambda_i)^+$. In Theorem 2 we show that inequality (2) is weak unless $\lambda_p > 0$.

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 $\lambda_4 = 10$. So the corresponding bottleneck cover inequality (2) is

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

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

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

□

Remark 1. Observe that the bottleneck cover inequality (2) is at least as strong as the flow cover inequality (Padberg et al. [15]) $\sum_{i=1}^{p}(a_{s_i} - \lambda)^+(1 - x_{s_i}) + \sum_{i=1}^{p} y_{s_i} \le u_{s_p}$, where $\lambda = \sum_{i=1}^{p} a_{s_i} - u_{s_p}$, since $\lambda = \lambda_1 \ge \lambda_i$ for all $i \in [1, p]$. If $b_i = 0$ for all $i \in [1, p]$, then $\lambda_i = \lambda_1$ for all $i \in [1, p]$ and the bottleneck cover inequality reduces to the flow cover inequality. □

Remark 2. Bottleneck cover inequality (2) is also at least as strong as the surrogate flow cover inequality (Pochet [16])

$$\sum_{s \in C} (y_s + (\min\{a_s, d_{s\ell}\} - \lambda)^+(1 - x_s)) \le d_{k\ell} + i_\ell \qquad (3)$$

for the lot–sizing problem, where $\lambda = \sum_{s \in C} \min\{a_s, d_{s\ell}\} - d_{k\ell}$ for $C \subseteq [\ell, k]$. In order to see this, using $d_{s\ell} = u_s - u_{\ell-1}$ and $i_\ell = u_{\ell-1} - \sum_{i=1}^{\ell-1} y_i$, we rewrite (3) as

$$\sum_{s=1}^{\ell-1} y_s + \sum_{s \in C} (y_s + (\min\{a_s, u_s - u_{\ell-1}\} - \lambda)^+(1 - x_s)) \le u_k \qquad (4)$$

and compare (4) with the bottleneck cover inequality (2) where $S = [1, \ell - 1] \cup C \cup \{k\}$; thus $s_p = k$. To verify that for $s_t \in C$, $\min\{a_{s_t}, u_{s_t} - u_{\ell-1}\} - \lambda = u_k - u_{\ell-1} - \sum_{s_i \in C} \min\{a_{s_i}, u_{s_i} - u_{\ell-1}\} + \min\{a_{s_t}, u_{s_t} - u_{\ell-1}\} \le a_{s_t} - \lambda_t = u_k - u_{s_{b_t}} - \sum_{j=b_t+1}^{p} a_{s_j} + a_{s_t}$, let $h = \max\{s_i \in C : a_{s_i} > u_{s_i} - u_{\ell-1}\}$ and observe that

$$u_{\ell-1} + \sum_{s_i \in C} \min\{a_{s_i}, u_{s_i} - u_{\ell-1}\} \ge u_h + \sum_{s_i \in C : s_i > h} a_{s_i}$$
$$\ge \min_{0 \le i \le p} \left\{ u_{s_i} + \sum_{j=i+1}^{p} a_{s_i} \right\} = u_{s_{b_t}} + \sum_{j=b_t+1}^{p} a_{s_j}.$$

□

Proposition 4. *The bottleneck cover inequality* (2) *is valid for* F.

Proof. 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. The statement holds trivially if $a_{s_i} \le \lambda_i$ for all $i \in [1, p]$. So let $k = \min\{j \in [1, \ell] : a_{s_{z_j}} > \lambda_{z_j}\}$. Then

$$\sum_{i=1}^{p} \min\{a_{s_i}, (a_{s_i} - \lambda_i)^+\}(1 - x_{s_i}) = \sum_{j=k}^{\ell} \min\{a_{s_{z_j}}, (a_{s_{z_j}} - \lambda_{z_j})^+\}$$
$$\le a_{s_{z_k}} - \lambda_{z_k} + \sum_{i=k+1}^{\ell} a_{s_{z_i}}$$
$$\le \max_{j \in [1,\ell]} \left\{ \left\{ \sum_{i=j}^{\ell} a_{s_{z_i}} - \lambda_{z_j} \right\}^+ \right\}. \qquad (5)$$

On the other hand, we also have

$$\sum_{i=1}^{p} y_{s_i} \le \min \left\{ u_{s_p}, \min_{j \in [1,\ell]} \left\{ u_{s_{b_{z_j}}} + \sum_{k=b_{z_j}+1}^{p} a_{s_k} - \sum_{k=j}^{\ell} a_{s_{z_k}} \right\} \right\}$$
$$= u_{s_p} - \max_{j \in [1,\ell]} \left\{ \left\{ u_{s_p} - u_{s_{b_{z_j}}} - \sum_{k=b_{z_j}+1}^{p} a_{s_k} + \sum_{k=j}^{\ell} a_{s_{z_k}} \right\}^+ \right\}$$
$$= u_{s_p} - \max_{j \in [1,\ell]} \left\{ \left\{ \sum_{k=j}^{\ell} a_{s_{z_k}} - \lambda_{z_j} \right\}^+ \right\}. \qquad (6)$$

Adding (5) and (6) shows that inequality (2) is satisfied by $(x, y) \in F$. □

Theorem 1. *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 $k = \max\{i \in [1, p-1] : x_{s_i} < 1\}$ (if no such k exists, let $k = 0$). Because either $k \ge 1$ and $y_{s_k} = u_{s_k} - \sum_{i=1}^{k-1} y_{s_i}$, or $k = 0$, 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 $u_{s_k} + \sum_{i=k+1}^{j} a_{s_i} \le u_{s_j}$ 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 $B_S \setminus \{0\}$.
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 $0 < x_{s_p} < 1$.
□

Theorem 2. *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* $K = \{k \in [1, n] \setminus S : u_{b_r} \leq u_k \leq u_t\} = \emptyset$, *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 $\{s_1, s_2, \dots, s_q\}$, we get an inequality dominating (2), since $\sum_{i=q+1}^{p}(a_{s_i}(1 - x_{s_i}) + y_{s_i}) \leq \sum_{i=q+1}^{p} a_{s_i} \leq u_{s_p} - u_{s_q}$. If $\max_{i\in[1,p]}\{a_{s_i} - \lambda_i\} \leq 0$, then $y_1 + \cdots + y_{s_p} \leq u_{s_p}$ dominates (2). If $k \in [1, s_{b_r}] \setminus S$, since b_r is the bottleneck of $[1, s_{b_r}]$, by Proposition 2, 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 $\{s_1, s_2, \dots, s_p\} \cup \{k\}$ 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 $\zeta(\{s_1, \dots, s_{r-1}\})$. 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 $y_{s_i}^{s_k} > 0$ and $x_{s_i}^{k} = 1$ 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 $y_{s_i}^{s_k} = a_{s_j}$ and $x_{s_i}^{s_k} = 1$ for $i \in [b_k + 1, p] \setminus \{k\}$; and $y_i^{s_k} = x_i^{s_k} = 0$ for $i \in [1, n] \setminus S$; and let $y_{s_i}^{s_k} > 0$ and $x_{s_i}^{k} = 1$ 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 $y_{s_i}^{s_k} = a_{s_i}$ and $x_{s_i}^{s_k} = 1$ for $i \in [b_k + 1, p] \setminus \{k\}$; and $y_i^{s_k} = x_i^{s_k} = 0$ for $i \in [1, n] \setminus S$.

For $k \in [1, p]$ such that $\lambda_k \geq a_{s_k}$, let $y_{s_i}^{s_k} > 0$ and $x_{s_i}^{s_k} = 1$ 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 $y_{s_i}^{s_k} > 0$ and $x_{s_i}^{s_k} = 1$ 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 $y_i^{s_k} = x_i^{s_k} = 0$ 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 $a_i \geq \epsilon_i > 0$ and $\sum_{i=1}^{b_r} \epsilon_i = u_{b_r}$. We see that $\bar{y}$ has a positive slack for constraints $y_1 + \cdots + y_i \leq u_i$ with $s_{b_r} + 1 \leq i < s_{b_r+1}$ since $u_{s_{b_r}} < u_{s_{b_r}+1}$; with $s_{b_r+1} \leq i < s_r$ since $K = \emptyset$; with $i \geq s_r$ since $\bar{y}$ has a slack of $a_{s_r} - \lambda_r$ for constraint $y_1 + \cdots + y_{s_p} \leq u_{s_p}$ 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 3. The necessity part of the proof of Theorem 2 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 2. □

As pointed out to us by L. A. Wolsey, the bottleneck cover inequalities are part of the more general submodular inequalities for capacitated fixed–charge networks (Wolsey [23]). The min–max relationship (1) and Propositions 2–3 allow us to characterize these inequalities explicitly for the lot–sizing problem and to lift them for deriving other strong inequalities. Interval submodular inequalities of Constantino [6] are more general than the bottleneck inequalities in the sense that they examine the effects of closing several production arcs in an interval simultaneously.

2.3. Lifting bottleneck covers

In this section we generalize bottleneck cover inequalities (2) by introducing pairs of variables (x_i, y_i) $i \in [1, n] \setminus S$ into 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

$$\sum_{i=1}^{p}((a_{s_i} - \lambda_i)^+(1 - x_{s_i}) + y_{s_i}) \leq u_{s_p} \tag{7}$$

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

$$\sum_{i=1}^{p}((a_{s_i} - \lambda_i)^+(1 - x_{s_i}) + y_{s_i}) + \sum_{i \in T}(\pi_i x_i + \mu_i y_i) \leq u_{s_p} \tag{8}$$

starting from (7).

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

$$\Phi(v) = u_{s_p} - \max\left\{\sum_{i=1}^{p}((a_{s_i} - \lambda_i)^+(1 - x_{s_i}) + y_{s_i}) : (x, y) \in F_T(u - v)\right\}, \tag{9}$$

where we let $\Phi(v) = \infty$ if $F_T(u - v) = \emptyset$, i.e., $v \nleq u$. By definition of Φ, inequality (8) is valid for F if and only if

$$\sum_{i \in T}(\pi_i x_i + \mu_i y_i) \leq \Phi(\sum_{i \in T} y_i g_i) \tag{10}$$

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 (7), we may ignore them when computing $\Phi(a g_\ell)$. Then, from assumption **(A2)**, all constraints $y_{s_1} + y_{s_2} + \cdots + y_{s_p} \leq u_i - a$ with $i > \max\{\ell, s_p\}$ are redundant and can be dropped from the problem. Thus, $P_{\Phi(a g_\ell)}$ can be stated as

$$\Phi(a g_\ell) = u_{s_p} - \max \sum_{i=1}^{p}((a_{s_i} - \lambda_i)^+(1 - x_{s_i}) + y_{s_i})$$

$$\begin{aligned} \text{s.t.} \qquad\qquad y_{s_1} &\leq u_{s_1} \\ y_{s_1} + y_{s_2} &\leq u_{s_2} \\ &\vdots \end{aligned}$$

$$
\begin{aligned}
y_{s_1} + y_{s_2} + \cdots + y_{s_p} &\leq u_{s_p} \\
&\vdots \\
y_{s_1} + y_{s_2} + \cdots + y_{s_p} &\leq u_\ell - a \\
y_{s_i} \leq a_{s_i} x_{s_i},\ y_{s_i} \in \mathbb{R}_+,\ x_{s_i} &\in \{0,1\}\ \ i \in [1,p]. \qquad (11)
\end{aligned}
$$

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* $a_{s_i} \leq \lambda_i$*,*
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* $i \in [1, p] \setminus \{k\}$*,*
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* $a_{s_i} > \lambda_i$*,*
5. $y_{s_i} = 0$ *for some* $i \in [1, p]$ *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 value. To see part *3*, observe that if $0 < y_{s_i} \leq a_{s_i} - \lambda_i$, then the objective value 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, $u_{s_h} + \sum_{i=h+1}^{p} a_{s_i} \geq u_{b_h} + \sum_{i=b_h+1}^{p} a_{s_i} = u_{s_p} + \lambda_h$. 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 $u_{s_h} \geq u_{s_{b_j}} + \sum_{i=b_j+1}^{h} a_{s_i} - (\lambda_j - \lambda_h)$ for $j \leq h$. Therefore, if $a_{s_j} > \lambda_j$, since $\lambda_h \geq 0$ (as (7) 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 $\sum_{i=1}^{h} y_{s_i} = u_{s_h}$.

Suppose (x, y) is a counterexample for part *5* with the smallest $y_{s_i} > 0$. We must have $\sum_{j=1}^{b_i} y_{s_j} = u_{s_{b_i}}$, $y_{s_j} = a_{s_j}$ for all $j \in [b_i, p] \setminus \{i\}$, since y_{s_i} can be reduced otherwise. Also from parts *3* and *4*, we have $y_{s_i} > a_{s_i} - \lambda_i$. However, this contradicts the feasibility of (x, y), since (x, y) does not satisfy constraint $\sum_{j=1}^{p} y_{s_j} \leq u_{s_p} - a$ as $a_{s_i} > \lambda_i = u_{s_{b_i}} + \sum_{j=b_i+1}^{p} a_{s_j} - u_{s_p}$. □

Since (7) is facet–defining for $conv(F_T)$, we have $\Phi(0) = 0$; consequently, $\Phi(ag_\ell) \geq 0$ for $a \geq 0$. Let $\delta_\ell := (u_\ell - u_{s_p})^+$. Since any optimal solution for $a = 0$ is feasible for $0 \leq a \leq \delta_\ell$, we also have $\Phi(ag_\ell) = 0$ for $0 \leq a \leq \delta_\ell$.

First suppose that $\ell > s_p$; we will remove this restriction later. From Proposition 2, 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 (11) has a slack of $a_{s_k} - \lambda_k$ for this solution for all $a \leq \delta_\ell + a_{s_k} - \lambda_k$, we have $\Phi(ag_\ell) = 0$ for $0 \leq a \leq \delta_\ell + a_{s_h} - \lambda_h$, where $h = \text{argmax}_{i \in [1,p]}\{a_{s_i} - \lambda_i : a_{s_i} > \lambda_i\}$ (since (7) is facet–defining, h exists).

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 (11) 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 $i \in [1, p] \backslash \{h\}$, 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 (11), the variable to reduce is y_{s_d}, where $d = \text{argmin}_{i \in [1,p] \backslash \{h\}:\, a_{s_i} > \lambda_i} \{\bar{y}_{s_i} - (a_{s_i} - \lambda_i)\}$ (ties broken by selecting one with the largest $a_{s_i} - \lambda_i$). 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. By letting $k = h$, the following lemma shows that $\bar{y}_{s_d} - (a_{s_d} - \lambda_d)$ indeed equals one of the λ_i's, which is not obvious unless $\bar{y}_{s_d} = a_{s_d}$.

Lemma 1. *Let $k \in [1, p]$ be such that $a_{s_k} > \lambda_k$. $\bar{P}_{\Phi(ag_\ell)}$ be the restriction of $P_{\Phi(ag_\ell)}$ with $x_k = y_k = 0$ and $a = \delta_\ell + a_{s_k} - \lambda_k$. Then $\bar{P}_{\Phi(ag_\ell)}$ has an optimal solution (x, y) such that*

$$\min_{i \in [1,p]} \left\{ y_i - (a_{s_i} - \lambda_i) : \; y_{s_i} > a_{s_i} - \lambda_i > 0 \right\} = \min_{i \in [1,p]} \left\{ \lambda_i : a_{s_i} > \lambda_i \right\}.$$

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 $i \in [b_k + 1, p]$ and $\sum_{i=1}^{b_k} y_{s_i} = u_{s_{b_k}}$. Then if $b_k = 0$, the result follows trivially. On the other hand, if $b_k \ge 1$, then, from Proposition 5.2, 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 $j \in [1, b_k]$. If $y_{s_j} - (a_{s_j} - \lambda_j) \ge y_i - (a_{s_i} - \lambda_i)$ for all $i \in [b_k + 1, p]$ such that $a_{s_i} > \lambda_i$, then we are done, since $y_{s_i} = a_{s_i}$ for all $i \in [b_k + 1, p]$. In the following we show that $y_{s_j} - (a_{s_j} - \lambda_j) = a_{s_j} - (\lambda_j - \lambda_k)$ and then the result follows from $a_{s_k} > \lambda_k$.

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 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 $y_{s_j} = a_{s_j} - (\lambda_j - \lambda_k)$. Furthermore, since $\lambda_j > \lambda_k$ (as $j \le b_k$), j exists; and we are done. □

Let $\alpha_1 := \max_{i \in [1,p]} \{a_{s_i} - \lambda_i : a_{s_i} > \lambda_i\}$ and $\beta_1 := \min_{i \in [1,p]} \{\lambda_i : a_{s_i} > \lambda_i\}$. From Lemma 1, for $a > \delta_\ell + \alpha_1$, $\Phi(ag_\ell)$ increases at the rate of one by reducing y_{s_d} from $\bar{y}_{s_d}$ to $\bar{y}_{s_d} - \beta_1$ and then reaches a flat region again because of the slack $a_{s_d} - \lambda_d$ introduced in constraint (11) by setting $y_{s_d} = x_{s_d} = 0$. 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 $\delta_5 = (u_5 - u_{s_4})^+ = 1$, $\alpha_1 = a_{s_1} - \lambda_1 = 2$, 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.

In Figure 1 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. Here observe that $\Phi(ag_\ell)$ is the point–wise minimum of these two lines (not drawn in the figure). □

The next proposition shows that the observation on $\Phi(ag_\ell)$ in Example 2 holds in general.

Proposition 6. *Let S and Φ be defined as in* (7) *and* (9)*. If $\ell > s_p$, then for $a > \delta_\ell$, $\Phi(ag_\ell)$ is the point–wise minimum of $\Phi_I(ag_\ell)$ for all I, where I is a permutation of S and $\Phi_I(ag_\ell)$ is the upper bound on $\Phi(ag_\ell)$, obtained by reducing the continuous variables in the order of I.*

Proof. Let (x, y) be an optimal solution for $P_{\Phi(ag_\ell)}$ such that $y_{s_i} = 0$ for some $i \in [1, p]$ with $a_{s_i} > \lambda_i$. By Proposition 5.5 such a solution exists. Let $Q = \{i \in [1, p] : y_{s_i} = 0\}$, $k = \min\{i \in Q : a_{s_i} > \lambda_i\}$, and $R = [b_k + 1, p] \setminus Q$.

Suppose constraint (11) has a slack $\rho > 0$ for (x, y). First observe that $\rho \le (a_{s_i} - \lambda_i)^+$ for all $i \in Q$, since otherwise letting $y_{s_i} = a_{s_i}$ improves the objective. Second, $\sum_{i=1}^{b_k} y_{s_i} = u_{s_{b_k}}$, because otherwise, since b_k is the bottleneck of s_k, $\sum_{i=1}^{b_k} a_{s_i} > u_{s_{b_k}}$, and $x_i = 1$ for all $i \in [1, b_k]$ (by assumption on k and Proposition 5.1), and $\rho > 0$, the objective can be improved by increasing some y_{s_i} $i \in [1, b_k]$. Also $y_{s_i} = a_{s_i}$ for all $i \in R$, since otherwise these variables can be increased to improve the objective

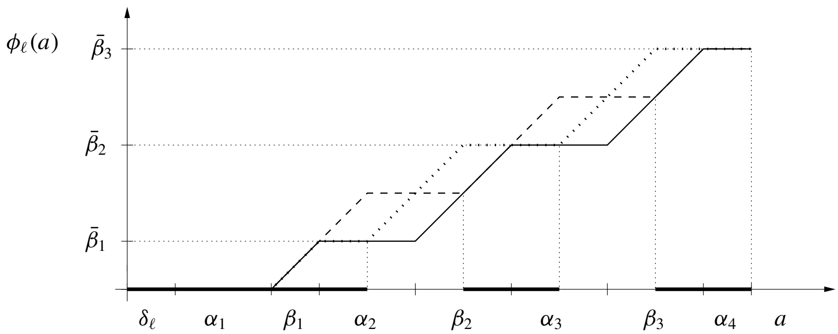

Fig. 1. Illustration of $\phi_\ell(a)$ in Example 2.

without violating feasibility as $\rho > 0$ and $\bar{y}_{s_k} = 0$, $\sum_{i=1}^{b_k} \bar{y}_{s_i} = u_{s_{b_k}}$, and $\bar{y}_{s_i} = a_{s_i}$ for all $i \in [b_k, p] \setminus \{k\}$ is a feasible (and optimal) solution when $a = 0$. Therefore,

$$u_{s_p} - (a - \delta_\ell) = \sum_{i=1}^{p} y_{s_i} + \rho = u_{s_{b_k}} + \sum_{i \in R} a_{s_i} + \rho.$$

Then

$$\Phi(ag_\ell) = u_{s_p} - \sum_{i \in Q} (a_{s_i} - \lambda_i)^+ - u_{s_{b_k}} - \sum_{i \in R} a_{s_i} = a - \delta_\ell + \rho - \sum_{i \in Q} (a_{s_i} - \lambda_i)^+.$$

Now let $I' = (q_1, q_2, \ldots, q_r)$ be some permutation of Q such that $q_1 = k$. For $a > \delta_\ell$, first letting $y_{s_{q_1}} = 0$, and then reducing the remaining $y_{s_{q_j}}$ $j = 2, 3, \ldots, r - 1$ from $a_{s_{q_j}}$ to 0, and $y_{s_{q_r}}$ from $a_{s_{q_r}}$ to $a_{s_{q_r}} - \rho$, since

$$a_{s_k} - \lambda_k + \sum_{i \in Q \setminus \{k\}} a_{s_i} - \rho = u_{s_p} - u_{s_{b_k}} - \sum_{i \in R} a_{s_i} - \rho = a - \delta_\ell$$

and $\rho \leq (a_{s_i} - \lambda_i)^+$ for all $i \in Q$, we obtain the upper bound

$$\Phi_{I'}(ag_\ell) = a - \delta_\ell - \sum_{i \in Q} (a_{s_i} - \lambda_i)^+ + \rho,$$

which equals $\Phi(ag_\ell)$.

Now suppose constraint (11) has no slack for (x, y), thus

$$u_{s_p} - (a - \delta_\ell) = \sum_{i=1}^{p} y_{s_i} = u_{s_{b_k}} + \sum_{i \in R} a_{s_i} - \varphi,$$

where $\varphi = a_{s_i} - y_{s_i}$ for some $i \in [1, b_k] \cup R$. Recall that by Proposition 5.2 we may assume there exists at most one y_{s_i} such that $a_{s_i} > y_{s_i} > (a_{s_i} - \lambda_i)^+$ and every other variable is at one of its bounds. So let $y_{s_t} = a_{s_t} - \varphi$. Then

$$\Phi(ag_\ell) = u_{s_p} - \sum_{i \in Q} (a_{s_i} - \lambda_i)^+ - u_{s_{b_k}} - \sum_{i \in R} a_{s_i} + \varphi = a - \delta_\ell - \sum_{i \in Q} (a_{s_i} - \lambda_i)^+.$$

Let $I' = (q_1, q_2, \ldots, q_r)$ be some permutation of $Q \cup \{t\}$ such that $q_1 = k$ and $q_r = t$. For $a > \delta_\ell$, first letting $y_{s_{q_1}} = 0$, and then reducing $y_{s_{q_j}}$ $j = 2, 3, \ldots, r - 1$ from $a_{s_{q_j}}$ to 0 and $y_{s_{q_r}}$ from $a_{s_{q_r}}$ to $a_{s_{q_r}} - \varphi$, since

$$\sum_{i \in Q} a_{s_i} - \lambda_k + \varphi = u_{s_p} - u_{s_{b_k}} - \sum_{i \in T} a_{s_i} + \varphi = a - \delta_\ell,$$

we obtain the upper bound

$$\Phi_{I'}(ag_\ell) = a - \delta_\ell - \sum_{i \in Q} (a_{s_i} - \lambda_i)^+ + \varphi,$$

which equals $\Phi(ag_\ell)$. □

Proposition 6 suggests a lower bound on $\Phi(ag_\ell)$ by sorting $a_{s_i} - \lambda_i$ in nonincreasing order, and λ_i in nondecreasing order separately and lining up alternating flat and sloped regions in this order.

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 $i \in [0, r]$, where $r = |\{i \in [1, p] : a_{s_i} > \lambda_i\}|$, and $\phi_\ell : \mathbb{R}_+ \mapsto \mathbb{R}_+$ as

$$\phi_\ell(a) = \begin{cases} 0 & \text{if } 0 \le a \le \delta_\ell, \\ \bar{\beta}_i & \text{if } \delta_\ell + \bar{\alpha}_i + \bar{\beta}_i \le a \le \delta_\ell + \bar{\alpha}_{i+1} + \bar{\beta}_i,\ i \in [0, r-1], \\ a - \delta_\ell - \bar{\alpha}_i & \text{if } \delta_\ell + \bar{\alpha}_i + \bar{\beta}_{i-1} \le a \le \delta_\ell + \bar{\alpha}_i + \bar{\beta}_i,\ i \in [1, r-1], \\ a - \delta_\ell - \bar{\alpha}_r & \text{if } \delta_\ell + \bar{\alpha}_r + \bar{\beta}_{r-1} \le a. \end{cases}$$

The solid line in Figure 1 shows $\phi_5(a)$ in Example 2. The bold line segments on the horizontal axis indicate the regions where the $\phi_5(a)$ equals $\Phi(ag_5)$, 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 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 description of a lower bound on $\Phi(ag_\ell)$ for a fixed $\ell \in T$; and from the preceding we have the following proposition.

Proposition 7. *Given a facet–defining inequality* (7) *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* $a_{s_i} = a_{s_k}$ *for all* $i, k \in [1, p]$ *such that* $a_{s_i} > \lambda_i$ *and* $a_{s_k} > \lambda_k$,
   (ii) *or* $\lambda_i = \lambda_k$ *for all* $i, k \in [1, p]$ *such that* $a_{s_i} > \lambda_i$ *and* $a_{s_k} > \lambda_k$.

Next we show how to obtain valid coefficients (π_i, μ_i) for all $i \in T$ in (8). The following lemma is the central result needed for proving the validity of the inequalities in this section.

Lemma 2. $\Phi(\sum_{i \in T} y_i g_i) \ge \sum_{i \in T} \phi_i(y_i)$ *for all* $T \subseteq [1, n] \setminus S$.

Proof. Let $\psi : \mathbb{R} \mapsto \mathbb{R}_+$ be defined as

$$\psi(a) = \begin{cases} 0 & \text{if } a \le 0, \\ \bar{\beta}_i & \text{if } \bar{\alpha}_i + \bar{\beta}_i \le a \le \bar{\alpha}_{i+1} + \bar{\beta}_i,\ i \in [0, r-1], \\ a - \bar{\alpha}_i & \text{if } \bar{\alpha}_i + \bar{\beta}_{i-1} \le a \le \bar{\alpha}_i + \bar{\beta}_i,\ i \in [1, r-1], \\ a - \bar{\alpha}_r & \text{if } \bar{\alpha}_r + \bar{\beta}_{r-1} \le a. \end{cases}$$

Then it follows that, for any $k \in T$,

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

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., $\psi(a_1) + \psi(a_2) \le \psi(a_1 + a_2)$ 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

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

In order to see that the last inequality holds, 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 $\psi(\sum_{i\in T} y_i - \delta_\ell) \geq \psi(\sum_{i\in T\setminus\{\ell\}}(y_i-\delta_i)) \geq \sum_{i\in T\setminus\{\ell\}} \psi(y_i-\delta_i)$. Then $\psi(\sum_{i\in T} y_i - \delta_\ell) \geq \sum_{i\in T\setminus\{\ell\}} \psi(y_i) + \psi(y_\ell-\delta_\ell) \geq \sum_{i\in T\setminus\{\ell\}} \psi(y_i-\delta_i) + \psi(y_\ell-\delta_\ell)$. □

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. [10]. 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 3. *Inequality* (8) *with* $(\pi_i, \mu_i) \in H_i$ *for* $i \in T$*, where* $S,\ T \subseteq [1, n]$ *and* $S \cap T = \emptyset$ *is valid for* F*. Moreover, such an inequality defines a facet of* $conv(F)$ *if inequality* (7) *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 (8) is a consequence of (10), which follows from Lemma 2 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 $\phi_i(a) = \Phi(ag_i)$ 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, $H_1^1 = \emptyset$, $H_1^2 = (-4, 1)$, $H_2^1 = \emptyset$, $H_2^2 = (-4, \frac{2}{3})$; 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 (8) may define facets even if $T \not\subseteq [s_p + 1, n]$. □

3. Special cases

3.1. Uncapacitated case

We obtain the uncapacitated case by letting $a_i = u_i$ for all $i \in [1, n]$. In this case, inequality (8) for $S = [1, \ell]$, $T \subseteq [\ell + 1, n]$ and $\ell \in [0, n - 1]$ reduces to

$$\sum_{i=1}^{\ell} y_i + (u_\ell - u_{\ell-1})(1 - x_\ell) + \sum_{i \in T} y_i \leq u_\ell + \sum_{i \in T}(u_i - u_{\ell-1})x_i, \tag{12}$$

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 is equivalent to the uncapacitated lot–sizing inequality (Barany et al. [2])

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

for $T \subseteq [1, \ell]$, $\ell \in [1, n]$. There is an $O(n \log n)$ separation algorithm for these inequalities and it is sufficient to add them to the LP relaxation to obtain a complete description of the lot–sizing polytope for the uncapacitated case [2].

3.2. *Constant–capacity case*

When the capacities are constant; i.e., $a_i = a$ for all $i \in [1, n]$, Proposition 3 implies that the coefficients of the binary variables in

$$\sum_{i=1}^{p} \min\{a, (a - \lambda_i)^+\}(1 - x_{s_i}) + \sum_{i=1}^{p} y_{s_i} \le u_{s_p} \tag{13}$$

are nondecreasing in $i \in [1, p]$. Since $\alpha_1 := \max_{i \in [1,p]}\{a_{s_i} - \lambda_i : a_{s_i} > \lambda_i\} = a - \lambda_p$, the lifting coefficients of inequality (8) reduce to

$$(\pi_i, \mu_i) = \begin{cases} (\lambda_p - a - \delta_i, 1) & \text{if } \delta_i < \lambda_p \\ (0, 0) & \text{otherwise} \end{cases} \quad i \in T \subseteq [1, n] \setminus S.$$

By Theorem 3 if $T \subseteq [s_p + 1, n]$, inequality (8) defines a facet of $conv(F)$ whenever (13) defines a facet of $conv(F_T)$, since $\phi_i(a) = \Phi(ag_i)$ as $a \le \alpha_1 + \beta_1$ ($= a$ in this case). The example below shows that $\phi_i(a)$ may be strictly less than $\Phi(ag_i)$ for $i \in T \cap [1, s_p]$ in the constant–capacity case.

Example 3. Let $a = 5$ and $u = (3, 7, 7, 11)$. For $S = \{s_1, s_2, s_3\} = \{1, 2, 4\}$, we have $\lambda_1 = 4, \lambda_2 = 2, \lambda_4 = 1$ and the corresponding bottleneck cover inequality

$$y_1 + y_2 + y_4 + (1 - x_1) + 3(1 - x_2) + 4(1 - x_4) \le 11 \tag{14}$$

is facet–defining for $conv(F_{\{3\}})$. The exact lifting of (14) with (x_3, y_3) gives

$$y_1 + y_2 + y_3 + y_4 + (1 - x_1) + 3(1 - x_2) - 3x_3 + 4(1 - x_4) \le 11. \tag{15}$$

Observe that $\phi_3(5) = \lambda_4 = 1 < \Phi(5g_3) = 2$ and $(\pi_3, \mu_3) = (-4, 1)$. Therefore inequality (15) cannot be obtained using ϕ_3 and H_3. □

The most general class of inequalities defined to date for constant–capacity lot sizing are the so–called $(k\ell SI)$ inequalities of Pochet and Wolsey [18]. They show that every valid inequality for F of the form

$$\sum_{i \in S} (y_i + \beta_i x_i) \le \pi_o$$

is a $(k\ell SI)$ inequality. Since the coefficient of y_i in inequality (8) is either 0 or 1, $(k\ell SI)$ inequalities subsume inequalities (8) in the constant–capacity case. Example 3 illustrates that (8) is a strict subclass of the $(k\ell SI)$ inequalities.

Separation algorithm. Now we describe a polynomial–time separation algorithm for inequalities (8) with $S \subseteq [1, n]$ and $T \subseteq [s_p + 1, n]$. Note that no polynomial–time separation algorithm is known to date for $(k\ell SI)$ inequalities unless S is fixed (Pochet and Wolsey [19]).

First we describe the separation algorithm for inequalities (13) 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 (13) with the largest left–hand–side value for (x, y). Let $G = (V, A)$ 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 $d_k = d_{t_i}$.

So G 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 with a forward pass from the sources to the sinks.

Next we assign lengths on the arcs. Observe that for the constant-capacity case, $a - \lambda_k = u_{s_p} - u_{s_{b_k}} - (p - b_k - 1)a$. 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 (13). In order to do that we define the length of an arc (i, k) as

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

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, Ahuja et al. [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 (13) under the assumption that i is the bottleneck of p. Any one of the longest of these s_p paths corresponds to a desired inequality (13).

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

$$\sum_{k \in T_i} (\bar{y}_k - \min\{a, u_{s_i} + (p - i - 1)a - u_k\}\bar{x}_k), \tag{16}$$

where $T_i = \{k \in [s_p + 1, n] : \min\{a, u_{s_i} + (p - i - 1)a - u_k\}\bar{x}_k < \bar{y}_k\}$. 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 (16) 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 (8) 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 experiments on using the inequalities introduced in Section 2 as cutting planes for solving the lot–sizing problem with a branch–and–cut algorithm. All experiments are done on a 2GHz Intel Pentium4/Linux workstation with 1GB main memory using the callable libraries of CPLEX[1] Version 8.1 Beta with one hour time limit.

For the experiments we created a data set of lot–sizing problem 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, 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 mean capacity/demand ratios $\texttt{c} \in \{2, 3, 4, 5\}$ and setup/holding cost ratios $\texttt{f} \in \{100, 200, 500, 1000\}$. Capacity c_t is drawn from integer uniform $[0.75\texttt{c}\bar{d}, 1.25\texttt{c}\bar{d}]$, setup cost s_t is drawn from integer uniform $[0.90\texttt{f}\bar{h}, 1.10\texttt{f}\bar{h}]$, 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 integer uniform [81,119] and [1,19], respectively. Here we report a summary of the solution performance measures for instances with 90 time periods. The data set is available for download at `http://ieor.berkeley.edu/~atamturk/data`.

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 (8) for each such S. Observe that for fixed 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 simply by testing $\pi_i x_i + \mu_i y_i > 0$ for $(\pi_i, \mu_i) \in H_i$ separately for each $i \in [1, n] \setminus S$.

We perform a number of experiments to evaluate the effectiveness of the inequalities described in Section 2 as cutting planes. CPLEX MIP solver also adds several classes of general cutting planes, including the flow cover inequalities [15] mentioned in Remark 1, to the formulation. In order to isolate the impact of the inequalities specific for the lot–sizing problem from CPLEX cuts, we perform two sets of experiments. The first set is without CPLEX cuts; the second is with CPLEX cuts.

The first experiment is done to find out the marginal contribution of the bottleneck cover (`bc`) inequalities (2) over the uncapacitated lot–sizing (`uls`) inequalities (12). In order to find out the effect of lifting, in the second experiment we test the lifted bottleneck cover (`lbc`) inequalities (8). Results of these experiments are summarized in Table 1. In this table we report the averages for the percentage integrality gap of the formulation before cuts are added ($\texttt{initgap} = 100 \times (\text{bestub} - \text{initlb})/\text{bestub}$), the percentage integrality gap after adding the cuts before branching ($\texttt{rootgap} = 100 \times (\text{bestub} - \text{rootlb})/\text{bestub}$), and the percentage improvement in the integrality gap at the root node ($\texttt{gapimp} = 100 \times (1 - \frac{\text{rootgap}}{\text{initgap}})$), where initlb, rootlb, bestub are the objective function values of the initial LP relaxation, LP relaxation after all cuts are added before branching, and the best feasible solution. We report also

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

Table 1. Experiments without CPLEX cuts.

c	f	exp	initgap	rootgap	gapimp	cuts	nodes	time
2	100	uls	3.60	1.52	57.85	68	700854	782
		bc		1.18	67.42	172	101858	219
		lbc		0.91	74.62	223	64693	171
	200	uls	3.10	1.68	45.28	58	1030963	1120
		bc		1.26	58.85	177	91558	182
		lbc		1.03	66.20	231	59807	149
	500	uls	2.11	1.42	32.06	42	904695	831
		bc		1.07	49.02	146	306429	504
		lbc		0.93	55.27	184	155901	290
	1000	uls	1.27	0.92	28.33	24	68675	52
		bc		0.63	54.03	91	111835	135
		lbc		0.56	59.91	122	104867	156
3	100	uls	6.11	1.77	70.89	105	449250	639
		bc		1.99	67.52	197	367582	805
		lbc		1.02	83.31	303	5677	29
	200	uls	5.60	2.24	60.05	98	1396180	1847
		bc		2.28	59.91	202	796147	1772
		lbc		1.41	75.18	340	228627	900
	500	uls	4.28	2.26	47.58	77	850426	1063
		bc		1.97	54.40	165	370474	672
		lbc		1.49	65.34	284	124742	387
	1000	uls	3.35	2.17	34.56	52	1125313	1250
		bc		1.66	49.96	151	496626	891
		lbc		1.47	55.59	206	175208	379
4	100	uls	7.54	1.51	79.80	119	91074	147
		bc		2.15	71.28	173	365794	740
		lbc		0.76	89.82	255	1804	13
	200	uls	7.92	2.47	68.54	111	162470	226
		bc		3.04	61.29	180	841806	1671
		lbc		1.53	80.49	361	18012	85
	500	uls	6.56	2.92	55.27	101	594321	767
		bc		2.97	54.30	176	1034808	1937
		lbc		1.95	70.06	307	42829	135
	1000	uls	5.03	2.67	46.04	74	720171	833
		bc		2.51	50.29	146	620410	888
		lbc		1.83	63.37	260	94628	249
5	100	uls	8.88	0.94	89.17	132	453	1
		bc		2.19	75.33	160	14826	31
		lbc		0.48	94.59	233	122	3
	200	uls	9.94	1.97	80.04	135	43955	62
		bc		3.84	61.18	171	638657	1265
		lbc		1.33	86.58	328	4459	24
	500	uls	9.08	3.28	63.98	123	379423	560
		bc		4.36	52.07	166	704089	1270
		lbc		2.42	73.44	421	51672	244
	1000	uls	7.13	3.28	54.08	97	270706	365
		bc		3.60	50.71	141	719460	884
		lbc		2.40	66.68	294	39821	136
Average		uls	5.73	2.07	56.91	89	531306	634
		bc		2.29	58.60	163	473897	867
		lbc		1.34	72.53	272	73304	209

uls: ineq. (12), bc: ineq. (2), lbc: ineq. (8).

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`). Each entry in the table corresponds to the average for five instances.

In Table 1 we observe that the initial integrality gap as well as the percentage gap improvement due to the cuts are negatively correlated with fixed cost/inventory cost ratio `f` and positively correlated with capacity/demand ratio `c`. The instances tend to become easier to solve as the capacity/demand ratio increases. The gap improvement due to the bottleneck cover (`bc`) cuts is consistently about 60% over varying capacities. However, the gap improvement due to the uncapacitated lot–sizing (`uls`) cuts increases with the capacity, as expected. While `bc` cuts are more effective for tightly capacitated instances, `uls` cuts perform better for higher capacities. These observations are clearer to see in Table 2, where we report the averages for each capacity parameter.

The positive effect of lifting is apparent in Tables 1 and 2. The lifted bottleneck cover (`lbc`) cuts improve the integrality gap by about 15% over `uls` and `bc` cuts. This leads to about 7 times reduction in the number of nodes explored and 3 times reduction in the solution time over all instances compared with `uls` cuts. Note that the difference in the impact of `uls` and `lbc` cuts is bigger for tightly capacitated instances.

In the next experiment we compare the marginal contribution of the lifted bottleneck cover (`lbc`) cuts over default CPLEX cuts, and CPLEX cuts + uncapacitated lot–sizing (`uls`) cuts. A summary of the results of this experiment is reported in Table 3. The results suggest that adding `uls` and `lbc` cuts on top of CPLEX cuts improve the performance of the algorithm considerably. Also the marginal contribution of the lifted bottleneck cover (`lbc`) cuts over CPLEX cuts + uncapacitated lot–sizing (`uls`) cuts is quite significant: the average number of nodes is reduced by a factor of 3, the average solution time is reduced by a factor of 2. The positive impact is apparent consistently for varying cost and capacity parameters.

Table 2. Experiments without CPLEX cuts (summarized).

c	exp	initgap	rootgap	gapimp	cuts	nodes	time
2	uls	2.46	1.39	39.60	47	699170	720
	bc		1.03	57.33	147	152920	260
	lbc		0.86	64.00	190	96317	192
3	uls	4.77	2.10	52.52	81	906304	1128
	bc		1.98	57.95	178	507707	1035
	lbc		1.35	69.86	283	133563	424
4	uls	6.76	2.39	62.41	101	392009	493
	bc		2.67	59.29	168	715704	1309
	lbc		1.52	75.94	296	39318	121
5	uls	8.76	2.37	71.82	122	173634	247
	bc		3.50	59.82	160	519258	863
	lbc		1.66	80.32	319	24018	102

`uls`: ineq. (12), `bc`: ineq. (2), `lbc`: ineq. (8).

Table 3. Experiments with CPLEX cuts.

c	f	exp	initgap	rootgap	gapimp	cuts	nodes	time
2	100	cpx	3.60	1.06	70.35	27	429066	583
		ulsx		0.77	78.46	90	50917	80
		lbcx		0.56	84.27	220	6067	24
	200	cpx	3.10	1.19	61.02	17	248307	276
		ulsx		0.99	67.34	73	123368	162
		lbcx		0.76	75.12	247	27952	82
	500	cpx	2.11	0.96	54.38	103	234549	436
		ulsx		0.92	56.27	132	245026	450
		lbcx		0.71	66.30	190	116169	266
	1000	cpx	1.27	0.57	58.03	220	81912	236
		ulsx		0.53	60.14	217	36836	104
		lbcx		0.40	70.31	123	33183	58
3	100	cpx	6.11	1.70	72.18	32	547439	775
		ulsx		0.96	84.38	138	17074	38
		lbcx		0.67	88.98	310	1861	13
	200	cpx	5.60	2.01	64.86	17	746093	1013
		ulsx		1.38	75.33	184	292260	596
		lbcx		1.18	79.28	348	150008	721
	500	cpx	4.28	1.52	64.76	22	162234	187
		ulsx		1.41	67.67	178	149686	281
		lbcx		1.14	73.83	284	71255	230
	1000	cpx	3.35	1.39	58.23	111	545482	615
		ulsx		1.33	60.07	213	323143	860
		lbcx		1.06	68.11	225	47061	121
4	100	cpx	7.54	1.48	80.29	37	93890	142
		ulsx		0.74	90.08	117	2293	6
		lbcx		0.36	95.21	231	566	6
	200	cpx	7.92	2.58	67.10	27	883612	1114
		ulsx		1.60	79.65	131	28603	50
		lbcx		1.10	85.94	350	4506	27
	500	cpx	6.56	2.44	62.55	113	317051	379
		ulsx		1.97	69.62	193	50073	85
		lbcx		1.58	75.69	323	45896	159
	1000	cpx	5.03	2.04	58.60	211	154954	346
		ulsx		1.87	62.30	344	145657	323
		lbcx		1.40	71.31	333	23883	74
5	100	cpx	8.88	1.11	87.23	45	2048	4
		ulsx		0.45	94.78	118	139	1
		lbcx		0.25	97.12	213	63	3
	200	cpx	9.94	2.57	74.08	38	85154	134
		ulsx		1.16	88.03	153	9731	21
		lbcx		0.85	91.24	315	1448	13
	500	cpx	9.08	3.80	58.32	20	598790	693
		ulsx		2.86	68.68	209	193174	583
		lbcx		1.86	79.51	395	42314	191
	1000	cpx	7.13	2.94	60.49	230	368214	938
		ulsx		2.51	65.44	267	253253	868
		lbcx		1.88	73.90	360	39902	176
Average		cpx	5.73	1.86	65.44	92	311002	454
		ulsx		1.34	72.99	172	117898	278
		lbcx		0.99	79.76	279	38258	135

cpx: CPLEX cuts, ulsx: CPLEX cuts + uls (12), lbcx: CPLEX cuts + lbc (8).

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 new 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 can be improved. Other construction and exchange heuristics may be developed to find more violated cuts. Identifying stronger lower bounds than ϕ for the lifting function Φ should reduce the integrality gap further.

A complete description of the lot–sizing polytope for the constant–capacity case is unknown. Investigation of the lifting function Φ for this special case deserves attention. Preliminary observations in this direction indicate that the constant–capacity lot–sizing polytope has facets that are not described by the $(k\ell SI)$ inequalities (Pochet and Wolsey [18]) (see also an example in Miller [13]); but are significantly harder to define explicitly than the ones identified in this paper.

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

Acknowledgements. We are grateful to Yves Pochet and Laurence A. Wolsey for many helpful comments on an earlier version of the paper and illuminating discussions on lot–sizing problems and mixed–integer programming during a visit to CORE, UCL. We thank Ed Rothberg for making changes in the CPLEX MIP solver that allowed us to add the cuts more efficiently and reduced the computation times significantly.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice-Hall, Englewood Cliffs NJ, 1993
2. Barany, I., Van Roy, T.J., Wolsey, L.A.: Uncapacitated lot sizing: The convex hull of solutions. Math. Program. Study **22**, 32–43 (1984)
3. Belvaux, G., Wolsey., L.A.: bc–prod: a specialized branch–and–cut system for lot–sizing problems. Manage. Sci. **46**, 724–738 (2000)
4. Belvaux, G., Wolsey., L.A.: Modelling practical lot-sizing problems as mixed integer programs. Manage. Sci. **47**, 993–1007 (2001)
5. Bitran, G.R., Yanasse., H.H.: Computational complexity of the capacitated lot size problem. Manage. Sci. **28**, 1174–1186 (1982)
6. Constantino., M.: A cutting plane approach to capacitated lot–sizing with start–up costs. Math. Program. **75**, 353–376 (1996)
7. Federgruen, A., Tzur., M.: A simple forward algorithm to solve general dynamic lot sizing models with n periods in $O(nlogn)$ or $O(n)$ time. Manage. Sci. **37**, 909–925 (1991)
8. Florian, M., Klein., M.: Deterministic production planning with concave costs and capacity constraints. Manage. Sci. **18**, 12–20 (1971)
9. Florian, M., Lenstra, J.K., Rinnooy Kan, H.G.: Deterministic production planning: Algorithms and complexity. Manage. Sci. **26**, 669–679 (1980)
10. Gu, Z., Nemhauser, G.L., Savelsbergh., M.W.P.: Sequence independent lifting in mixed integer programming. J. Combinatorial Optim. **4**, 109–129 (2000)
11. Leung, J.M.Y., Magnanti, T.L., Vachani., R.: Facets and algorithms for capacitated lot sizing. Math. Program. **45**, 331–359 (1989)
12. Loparic, M., Marchand, H., Wolsey., L.A.: Dynamic knapsack sets and capacitated lot–sizing. Math. Program. **95**, 53–69 (2003)
13. Miller, A.,: Polyhedral Approaches to Capacitated Lot-Sizing Problems. PhD thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, December 1999

14. Miller, A., Nemhauser, G.L., Savelsbergh., M.W.P.: On the capacitated lot–sizing and continuous 0–1 knapsack polyhedra. Eur. J. Oper. Res. **125**, 298–315 (2000)
15. Padberg, M.W., Van Roy, T.J., Wolsey., L.A.: Valid linear inequalities for fixed charge problems. Oper. Res. **32**, 842–861 (1984)
16. Pochet., Y.: Valid inequalities and separation for capacitated economic lot sizing. Oper. Res. Lett. **7**, 109–115 (1988)
17. Pochet, Y., Wolsey., L.A.: Solving multi–item lot–sizing problems using strong cutting planes. Manage. Sci. **37**, 53–67 (1991)
18. Pochet, Y., Wolsey., L.A.: Lot–sizing with constant batches: Formulation and valid inequalities. Math. Oper. Res. **18**, 767–785 (1993)
19. Pochet, Y., Wolsey., L.A.: Polyhedra for lot-sizing with Wagner-Whitin costs. Math. Program. **67**, 297–323 (1994)
20. Van Hoesel, C.P.M., Wagelmans., A.P.M.: An $O(T^3)$ algorithm for the economic lot-sizing problem with constant capacities. Manage. Sci. **42**, 142–150 (1996)
21. Wagelmans, A., Van Hoesel, S., Kolen., A.: Economic lot sizing: An $O(n \log n)$ algorithm that runs in linear time in the Wagner-Whitin case. Oper. Res. **40**, S145–S156 (1992)
22. Wagner, H.M., Whitin., T.M.: Dynamic version of the economic lot size model. Manage. Sci. **5**, 89–96 (1958)
23. Wolsey., L.A.: Submodularity and valid inequalities in capacitated fixed charge networks. Oper. Res. Lett. **8**, 119–124 (1989)
24. Wolsey., L.A.: Solving multi-item lot-sizing problems with an mip solver using classification and reformulation. Manage. Sci. **48**, 1587–1602 (2002)