Lot Sizing Problem with Backlogging

Ping Zhan*

Abstract

Production planning of lot sizing problems was an important subject almost half a century ago, and still is studied extensively recently. In this manuscript, we formulate the problem and clarify a condition related and introduce its application.

1. Introduction

Optimization problems of minimizing costs or maximizing profits in production planning and economics have been widely studied for a long time. Affluent results not only beauty and profound in mathematics have been obtained, their practical successes in applications are great achievements.

Lot sizing models were treated almost at the beginnings of operations research and management science. With the introduction of Materials Requirement Planning (MRP), or today's Enterprise Resource Planning (ERP) systems and Advanced Planning and Scheduling (APS) systems, supported by new information technology, lot sizing problems clearly have got some new meaning and attention.

First, unlike many classical production models, they concentrate production functions, lot sizing models also integrate set-ups and inventory costs, with longer time horizon planning. These ideas are matching the advanced planning system, or Supply Chain Planning (SCP), which is the main trend in today's business management.

Consider mathematical formulations of lot sizing models, production and inventory variables are usually real numbers, while set-up (or start-up) variables are often binary, which means that the problems are Mixing Integer Programming (MIP), and it is hard to find an efficient algorithm in general. Thanks to Traveling Salesman Problem (TSP), one of most intensively studied problems in optimization, the efforts on TSP attract the research on cutting plane method which combining other approaches have greatly improved efficiency of solving MIP in last decade, and make large size MIP more attackable in a reasonable time.

Lot sizing problems are also a very interesting subject for researchers on mathematical programming. A wide range of theories and algorithms have been applied to find structures of optimal solutions and to solve them. For example, convex analysis, minimum cost of networks, madroids and submodular functions, also, dynamic programming, cutting planes.

For single item lot sizing models, complete linear description of polyhedra have been known by introducing cutting planes or variables. Recently the linear description also has been solved even when demands are violated by backlogging ([1]). Efficient algorithms also be developed to solving them, especially when lot sizing models are single item and uncapacitated, or constant capacity and discrete.

On the other side, many problems are remained, one extreme is if capacities of lot sizing model are time vary, it is NP-hard. And when there are more than two items, especially some components can be produced only others have been finished, we can only solve these production planning problems case by case.

In next section, we define notations and give the formulation of lot sizing problems. Section 3 clarifies a condition related to backlogging and Section 4 its application in formulating lot sizing model. Finally we show an example using branch-and-bound algorithm.

2. Definitions and Preliminaries

Let n be the length of the planning time horizon. For each period $t \in \{1, 2 \cdots, n\}$ the following data are given:

p_t unit production cost in t,

h_t unit holding cost in t (defined also for $t=0$), ,

q_t set-up cost in t,

b_t unit backlogging cost (defined also for $t=0$),

d_t demand in t,

C_t production capacity in t.

We suppose all data is nonnegative rational. For easy of presentation, we also denote

$d_{kl} = \sum_{t=k}^{t=l} d_t$. And variables are defined as follows:

x_t production in t,

s_t stock at the end of t (defined also for $t=0$),

y_t set-up binary variables,

r_t total accumulated backlog at the end of t (defined also for $t=0$).

Now we can formulate lot sizing model with capacity and backlogging (LSB):

$$\min \sum_{t=1}^{n} (p_t x_t + h_t s_t + q_t y_t + b_t r_t) \tag{2.1}$$

$$s_{t-1} + x_t - r_{t-1} = d_t + s_t - r_t \qquad \text{for } 1 \le t \le n \tag{2.2}$$

$$(LSB) \qquad\qquad\qquad x_t \le C_t y_t \qquad \text{for } 1 \le t \le n \tag{2.3}$$

$$x_t, s_t, r_t \in \mathbf{R}_+, y_t \in \{0, 1\} \qquad \text{for } 1 \le t \le n \tag{2.4}$$

Here we suppose $s_0 = 0$ and $r_0 = 0$ for simiplicity.

Replace flow conservation equation (2.2) to (2.1), we can rewritten objective function as

$$\min \sum_{t=1}^{n} (h'_t s_t + q_t y_t + b'_t r_t) \ (+ \sum_{t=1}^{n} p_t d_t), \tag{2.5}$$

with $h'_t = h_t + p_t - p_{t+1}$ and $b'_t = b_t + p_{t+1} - p_t$, where we take $p'_0 = 0$. We say that a problem with backlogging has Wagner-Whitin costs if $h'_t \geq 0$ for $(t = 0, 1, \cdots, n-1)$ and $b'_t \geq 0$ for $(t = 2, 3, \cdots, n)$. Wagner-Whitin condition means that it will be costly for early production when there are positive stocks and for late production with positive backlogs.

3. Backlogging Conditions

Without other ristrictions, it seems unreasonable for same time period t we have both $s_t > 0$ and $r_t > 0$. We discuss this in detail.

Suppose $s_q \geq r_q > 0$ for any time period q, let $s'_q = s_q - r_q > 0$, $s'_t = s_t$ if $t \neq q$ and $r'_q = 0$, $r'_t = r_t$ if $t \neq q$, then

$$\sum_{t=1}^{n} (p_t x_t + f_t y_t + h_t s_t + b_t r_t) - \sum_{t=1}^{n} (p_t x_t + f_t y_t + h_t s'_t + b'_t r_t) = h_q s'_q + b_q r_q \geq 0,$$

while flow conservation condition and other nonnegative condition are remained unchanged. For the case of $r_q > s_q > 0$, the discussion is the sams as the above.

Summerize above discussion we have following proposition.

Proposition 3.1: *For an optimimal solution of problem LSB, we always have a necessary condition.*

$$s_t r_t = 0 \qquad \text{for } 1 \leq t \leq n. \tag{3.1}$$

Note when the problem have Wagner-Whitin costs, we can obtain the same result.

The solution structure of necessary condition can also be explained in following Figure 3.1., i.e., there exists no two parallel flows in any one period.
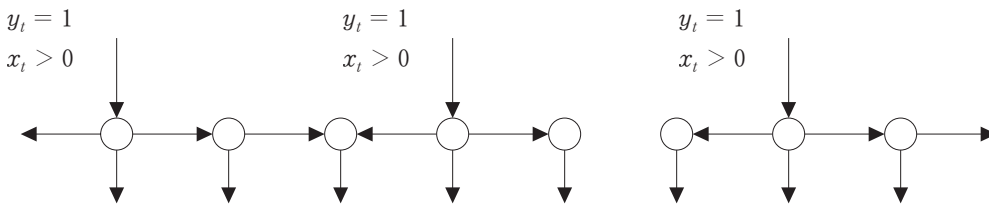

$$\begin{array}{ccc} y_t = 1 & y_t = 1 & y_t = 1 \\ x_t > 0 & x_t > 0 & x_t > 0 \end{array}$$

Figure 3.1 Structure of an optimal solution of *LSB*

4. Optimization Modeling

In this section, we see how (3.1) is applied to modeling and optimization.

For simplicity, we consider discrete lot sizing problem, i.e., $x_t = C_t y_t$ for $(t = 1, 2 \cdots, n)$. Then, the flow conservation constraints can be rewritten as

$$\sum_{u=1}^{t} C_u y_u = d_{1t} + s_t - r_t.$$

Similarly, if we eliminate the r_t variables, the objective function can be written as

$$\sum_{t=1}^{n} (h_t s_t + b_t r_t + q_t y_t) = \sum_{t=1}^{n} [(h_t + b_t) s_t + (q_t - C_t \sum_{u=t}^{n} b_u) y_t] + const,$$

where the constant term is $\sum_{t=1}^{n} b_t d_{1n}$.

Eliminate r_t in flow conservation constraints, we have

$$s_t \geq \sum_{u=1}^{n} C_u y_u - d_{1t},$$

or by (3. 1) we have

$$s_t = (\sum_{u=1}^{n} C_u y_u - d_{1t})^+,$$

Summerize the above discussions, we have

$$z(j, p) = \min \{ \sum_{i=1}^{j} [q_i' y_i + h_i' (\sum_{u=1}^{i} C_u y_u - d_{1i})^+] : \sum_{i=1}^{j} y_i = p, \ y \in \{0, 1\}^j \}.$$

Now we consider the problem with *constant* capacities, and suppose that $y \in \{0, 1\}^n$ is the characteristic vector of $S \subseteq N = \{1, 2, \cdots, n\}$, and rephrase the problem,

$$v(s) = \sum_{i \in S} q_i' + \sum_{i \in N} h_i' (C \mid S \cap N_i \mid - d_{1i})^+,$$

where $N_i = \{1, 2, \cdots, i\}$. The set function has supermodular property, and an efficient greedy algorithm can be used to solve it ([2]). While the original (BLS) problem is equivalent to

$$\min_{p} \ z(n, p) = \min_{s} \ v(S).$$

5. Branch-and-Bound Algorithm

Note when $d_t = 0$ for $t = 1, 2, \cdots, n-1$, and $x_t = C_t y_t$, then (LSB) can be simplified as

$$\min \sum_{t=1}^{n} (p_t C_t + q_t) y_t = \min \sum_{t=1}^{n} q_t' y_t$$

$$\sum_{t=1}^{n} C_t y_t \geq d_n,$$

$$y \in \{0, 1\}^n$$

which is a knapsack problem, and is NP-hard. Therefore, it is almost hopeless to found a good algorithm for a LSB problem with non-constant capacities.

Here we try to solve (LSB) with a general branch-and-bound algorithm. We suppose $x_t = C_t y_t$, i.e., the (LSB) is discrete. We reformulate the problem:

$$\min \sum_{t=1}^{n} (h_t s_t + q_t y_t + b_t r_t)$$

(DLSB) $$\sum_{u=1}^{t} C_u y_u = d_{1t} + s_t - r_t \qquad \text{for } 1 \leq t \leq n$$
$$s, r \in \mathbf{R}_+^n, \ y \in \{0, 1\}^n.$$

And its linear relaxation is:

$$\min \sum_{t=1}^{n} (h_t s_t + q_t y_t + b_t r_t)$$

$(RDLSB)$
$$\sum_{u=1}^{t} C_u y_u = d_{1t} + s_t - r_t \qquad \text{for } 1 \le t \le n,,$$

$$s, r \in \mathbf{R}_+^n, \ y \in (0, 1)^n.$$

Let Z and Z_R be the values of optmization solutions of DLSB and RDLSB respectively, and $\bar{Z}$ be the objective value of any feasible solution of DLSB. Then we have

$$Z_R \le Z \le \bar{Z}$$

Here we show an example. The time period n is 6, and other data are given in Table 4.1.

Table 4.1 Given data

C	5	7	9	12	8	6
d	3	6	2	3	7	5
q	70	20	50	30	40	10
h	1	2	3	2	1	3
b	2	3	6	4	2	4

First, we obtain $Z_R = 100.72$ with optimization solution $y^* = (0, 1, 0.44, 0.75, 0, 1)$. Here we choose the initial binary $y = (0, 1, 0, 0, 1, 0, 1)$ by round y^*, and obtain $\bar{Z} = 114$ with $s = (0, 0, 0, 5, 0, 0)$ and $r = (3, 2, 4, 0, 2, 1)$.

Since y^* is not integer, The following branch-and-bound iterations are summerized in Table 4.2 and Table 4.3.

Table 4.2 Branch-and-bound iterations by depth-first search

Steps	Z	$\bar{Z}$	y, s, r	branch list L
initialization		114		$\{\{y_3=0\}, \{y_3=1\},$ $\{y_4=0,\}, \{y_4=1\}\}$
$y_3=0$ $Z < \bar{Z}$, branch	113		$y = (0.2, 1, 0, 1, 0, 1)$ $s = (0, 0, 0, 6, 0, 0)$ $r = (2, 1, 3, 0, 1, 0)$	$\{\{y_3=1\}, \{y_4=0,\}, \{y_4=1\}$ $\{y_3=0, y_1=0\}, \{y_3=0, y_1=1\}\}$
$y_3=0, y_1=1$ $Z < \bar{Z}$, prune	139.64			$\{\{y_3=1\}, \{y_4=0,\}, \{y_4=1\}$ $\{y_3=0, y_1=0\}\}$
$y_3=0, y_1=0$ $Z < \bar{Z}$, branch	113		$y = (0, 1, 0, 1, 0.1, 1)$ $s = (0, 0, 0, 5, 0, 0)$ $r = (3, 2, 4, 0, 1, 0)$	$\{\{y_3=1\}, \{y_4=0,\}, \{y_4=1\}$ $\{y_3=0, y_1=0, y_5=0\}$ $\{y_3=0, y_1=0, y_5=1\}\}$
$y_3=0, y_1=0, y_5=1$ $Z > \bar{Z}$, prune	121.16			$\{\{y_3=1\}, \{y_4=0,\}, \{y_4=1\}$ $\{y_3=0, y_1=0, y_5=0\}\}$
$y_3=0, y_1=0, y_5=0$ $Z = \bar{Z}$, opt.	114		$y = (0, 1, 0, 10, 1)$ $s = (0, 0, 0, 5, 0, 0)$ $r = (3, 2, 4, 0, 2, 1)$	$\{\{y_3=1\}, \{y_4=0,\}, \{y_4=1\}\}$

Table 4.3   Branch-and-bound iterations by breadth-first search

Steps	Z	$\bar{Z}$	y, s, r	branch list L
initialization		114		$\{\{y_3{=}0\}, \{y_3{=}1\},$ $\{y_4{=}0,\}, \{y_4{=}1\}\}$
$y_3{=}0$ $Z < \{\text{barZ}\}$, branch	113		$y{=}(0.2, 1, 0, 1, 0, 1)$ $s{=}(0, 0, 0, 6, 0, 0)$ $r{=}(2, 1, 3, 0, 1, 0)$	$\{\{y_3{=}1\}, \{y_4{=}0, \}, \{y_4{=}1\}$ $\{y_3{=}0, y_1{=}0\}, \{y_3{=}0, y_1{=}1\}\}$
$y_3{=}1$ $Z > \{\text{barZ}\}$, prune	129.21			$\{\{y_4{=}0, \}, \{y_4{=}1\}$ $\{y_3{=}0, y_1{=}0\}, \{y_3{=}0, y_1{=}1\}\}$
$y_4{=}0$ $Z > \{\text{barZ}\}$, prune	121.89			$\{\{y_4{=}1\} \{y_3{=}0, y_1{=}0\},$ $\{y_3{=}0, y_1{=}1\}\}$
$y_4{=}1$ $Z < \{\text{barZ}\}$, branch	107.44		$y{=}(0, 1, 0.2, 1, 0, 0.8)$ $s{=}(0, 0, 0, 7, 0, 0)$ $r{=}(3, 2, 2, 0, 0, 0)$	$\{\{y_3{=}0, y_1{=}0\}, \{y_3{=}0, y_1{=}1\}$ $\{y_4{=}1, y_3{=}0\}, \{y_4{=}1, y_3{=}1\}$ $\{y_4{=}1, y_6{=}0\}, \{y_4{=}1, y_6{=}1\}$
$y_3{=}0, y_1{=}0$ $Z < \{\text{barZ}\}$, branch	113		$y{=}(0, 1, 0, 1, 0.1, 1)$ $s{=}(0, 0, 0, 5, 0, 0)$ $r{=}(3, 2, 7, 0, 1, 0)$	$\{\{y_3{=}0, y_1{=}1\}$ $\{y_4{=}1, y_3{=}0\}, \{y_4{=}1, y_3{=}1\}$ $\{y_4{=}1, y_6{=}0\}, \{y_4{=}1, y_6{=}1\}$ $\{y_3{=}0, y_1{=}0, y_5{=}0\})$ $y_3{=}0, y_1{=}0, y_5{=}1\}\}$
$y_3{=}0, y_1{=}1$ $Z > \{\text{barZ}\}$, prune	139.64			$\{\{y_4{=}1, y_3{=}0\}, \{y_4{=}1, y_3{=}1\}$ $\{y_4{=}1, y_6{=}0\}, \{y_4{=}1, y_6{=}1\}$ $\{y_3{=}0, y_1{=}0, y_5{=}0\})$ $\{y_3{=}0, y_1{=}0, y_5{=}1\}\}$
$y_4{=}1, y_3{=}0$ …	113		$y{=}(0.2, 1, 0, 1, 0, 1)$ …	$\{\{y_4{=}1, y_3{=}1\}$ …

In above example, we initialized $\bar{Z}$ as lower as possible, and therefore there are no need to update list L when a better $\bar{Z}$ obtained, it is clear to reach an optimization, we have to iterate four more steps for breadth-first search.

In each iteration, we optimize the related relaxation problem, equation (3.1) can not be used as a cutting plane here. Although we have little hope to find a tight characterization of polyhedra of (DLSB), it is still a challenge to find a good cutting plane.

References

[1] S. Küçükyavuz and Y. Pochet: Uncapacitated lot sizing with backlogging: the convex hull, *Mathematical Programming* **118** (2009) 151–175.
[2] V. Pochet, L.A. Wolsey: *Production Planning by Mixed Integer Pregramming,* Springer, 2006.