Special issue: Theoretical computer science and discrete mathematics

Research Paper

A dynamic programming algorithm for lot-sizing problem with outsourcing

Ping ZHAN[1]

[1]*Department of Communication and Business, Edogawa University*

ABSTRACT

Lot-sizing problem has been extensively researched in many aspects. In this manuscript, we give a dynamic programming algorithm scheme for lot-sizing problems with outsourcing.

KEYWORDS

Mixed integer problem, Lot-sizing, Dynamic programming, Minimum cost network flow

1 Introduction and definitions

For lot-sizing mixing integer problems, tight description of their polyhedra has been achieved for many variations by cutting plans, extended formulations, totally unimodular matrices and other properties crossing wide fields, see [4]. Dynamic programming (DP) algorithms are still powerful for general lot-sizing problems because their structures ([2], [3]).

Now we give notations and formulate the general lot-sizing problem as follows. 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,
g'_t unit outsourcing cost in t,
h'_t unit holding cost in t (defined also for $t = 0$),
q_t set-up cost in t,
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,
z_t outsourcing in t,
s_t stock at the end of t (defined also for $t = 0$),
y_t set-up binary variables,

Now we can formulate lot-sizing model ($LS - O$):

$$\min \sum_{t=1}^{n} (p'_t x_t + g'_t z_t + h'_t s_t + q_t y_t) + h'_0 s_0 \qquad (1.1)$$

$$s_{t-1} + x_t + z_t = d_t + s_t \qquad \text{for } 1 \le t \le n \qquad (1.2)$$

$$x_t \le C_t y_t \qquad \text{for } 1 \le t \le n \qquad (1.3)$$

$$x_t, z_t, s_t \in \mathbf{R}_+, \ y_t \in \{0, 1\} \qquad \text{for } 1 \le t \le n. \qquad (1.4)$$

Throughout the manuscript we suppose $s_0 = 0$ for simplicity.

2 Structure of optimal solutions

First, note that if the capacities production are period varying, the problem is NP-hard, without special mention, we suppose the capacities are constant, i.e., equations (1.3) are rewritten as

$$x_t \le C y_t \qquad \text{for } 1 \le t \le n. \qquad (2.1)$$

By the definition of problem, it is reasonable to assume that

$$g'_t \ge p'_t \qquad \text{for } 1 \le t \le n. \qquad (2.2)$$

Hence, there is an optimal solution that we have $z_t > 0$ only if $x_t = C$ or $x_t = 0$ by the assumption of nonnegative of set-up production cost and (2.2).

Suppose that set-up variables $y \in \{0, 1\}^n$ are known, the variables in the flow conservation of constraints (1.2) together with set-up conditions (1.3) (or (2.1)) can be represented as flows in a network. And then lot-sizing problem ($LS - O$) is a minimum cost network flow problem. In Fig. 1., such a network with $n = 5$ is shown as an instance. By $(x_i(y_i), z_i)$, we means that there are two flows on the arc, instead of this, we can draw two arcs, one with flow $x_i(y_i)$, the other with flow z_i.

Received October 28, 2011; Revised January 10, 2012; Accepted January 11, 2012.

[1)] *zhan@edgawa-u.ac.jp*

DOI: 10.2201/NiiPi.2012.9.6

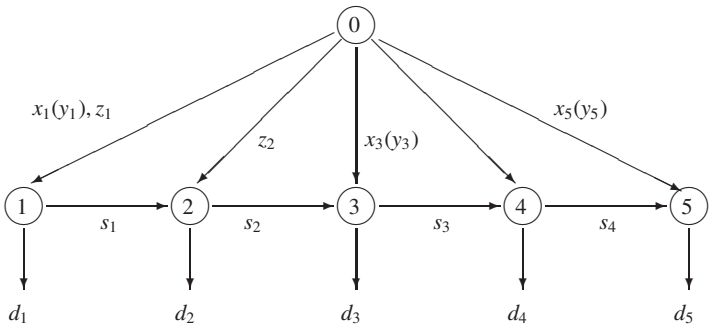

Fig. 1 An example of the fixed charge netowrk flow variables

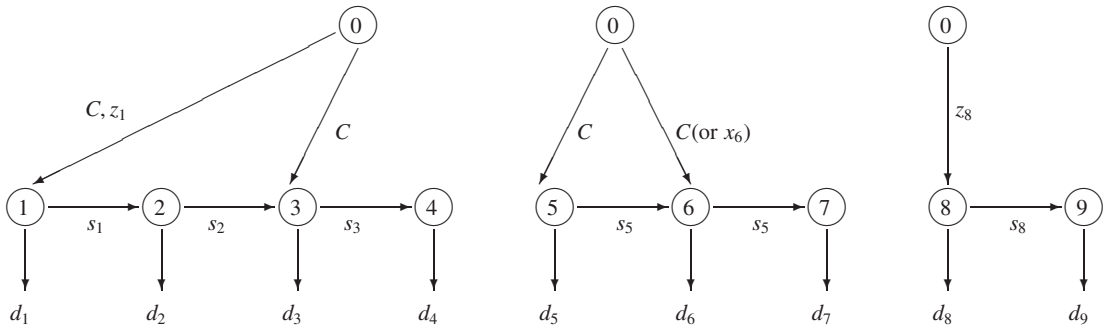

Fig. 2 An example of structure of extreme optimal solution

For a minimum cost network flow problem, a well-known and fundamental property of minimum cost network flow problem tells:

Observation 2.1: *For a basic feasible solution of a minimum cost network flow problem, the arcs corresponding to variables with flows strictly between their lower and upper bounds form an acyclic graph.*

Before giving the structure of optimal solutions, we need a concept defined as following.

Definition 2.1: Planning time period n are partitioned into intervals $[t_1, t_2 - 1], [t_2, t_3 - 1], \cdots, [t_{r-1}, t_r]$, where no stock entering or leaving each interval, are called *regeneration intervals*.

Now we have the property about the structure of optimal solutions to $LS - O$, which is critical for the dynamic programming algorithm described in next section.

Proposition 2.1 *There exists an extreme optimal solution to $LS - O$, with each regeneration interval, there exists at most one production with partial capacity or outsourcing.*

Note Proposition 2.1 is also applied to the case when production capacities are periods varying. An instance with $n = 9$ is shown in Fig. 2.

3 Dynamic programming algorithm

Using the flow balance equalities (1.2), we can omit stock valuables from object function.

Observation 2.1: *The objective function (1.1) of LS-O can be written as*

$$\sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} g_t z_t + \sum_{t=1}^{n} q_t y_t + K, \tag{3.1}$$

where $p_t = p'_t + \sum_{j=t}^{n} h'_j$ *and* $g_t = g'_t + \sum_{j=t}^{n} h'_j$ *for* $t = 1, \cdots, n$, $K = -\sum_{t=1}^{n} h'_t d_{1t}$.

Proof. By (1.2), we obtain $s_t = \sum_{u=1}^{t} x_u + \sum_{u=1}^{t} z_u - d_{1t}$, substitute it to objective function

$$\begin{aligned}
&\sum_{t=1}^{n} p'_t x_t + \sum_{t=1}^{n} g'_t z_t + \sum_{t=1}^{n} h'_t s_t \\
&= \sum_{t=1}^{n} p'_t x_t + \sum_{t=1}^{n} g'_t z_t + \sum_{t=1}^{n} h'_t \sum_{u=1}^{t} x_u \\
&\quad + \sum_{t=1}^{n} h'_t \sum_{u=1}^{t} z_u - \sum_{t=1}^{n} h'_t d_{1t} \\
&= \sum_{t=1}^{n} (p'_t + h'_t + \cdots + h'_n) x_t
\end{aligned}$$

$$+\sum_{t=1}^{n}(g'_t + h'_t + \cdots + h'_n)z_t - \sum_{t=1}^{n} h'_t d_{1t}. \quad \square$$

Note by same arguments, we can also omits production variables or outsourcing variables from objective function.

Now we go on to DP for the problem of finding an optimal solution on a given regeneration interval $[k, l]$. Let $\rho_{kl} = d_{kl} - \lfloor \frac{d_{kl}}{C} \rfloor C$ with $0 \le \rho_{kl} < C$. Define also $d^m_{kt} = \min\{i \mid i * C + \rho_{kl} > d_{kt}\}$ for $i \in \mathbf{Z}$, $k \le t < l$, and $d^m_{kl} = \lfloor \frac{d_{kl}}{C} \rfloor$. Let $d^{m\tau}_{kt} = \max\{0, d^m_{kt} - \tau\}$ for $k \le t \le l$ and $0 \le \tau \le \lfloor \frac{d_{kl}}{C} \rfloor$.

Let $G_k(t, \tau, \theta, \chi^v(t))$ be the value of a minimum cost solution for periods k up to t during which production occurs τ times at full capacity, $\theta \in \{0, 1\}$ times at level ρ_{kl}, and outsourcing occurs $\chi^v(t) \in \{0, 1\}$ times at level $(d_{kt} - \tau)C + \rho_{kl}$, where v is the period with $\chi^v(v) = 1$.

Dynamic programming recursion for regeneration intervals of $LS - O$

First set $G_k(k-1, \tau, \theta, \chi^v(t)) = 0$ for all $(\tau, \theta, \chi^v(t))$ with $\tau \le 0$, $\theta \le 0$ and $\chi^v(t) \le 0$, and $G_k(k-1, \tau, \theta, \chi^v(t)) = \infty$ otherwise. Also set $G_k(t, -1, 0, 1) = \infty$ for all $k \le t \le l$. A forward recursion to compute $G_k(t, \tau, \theta, \chi^v(t))$ is:

$$G_k(t,\tau,0,0) = \begin{cases} \infty & \text{if } \tau C \le d_{kt} \text{ or } \tau > t-k+1 \\ \min\left\{ \begin{array}{l} G_k(t-1,\tau,0,0), \\ G_k(t-1,\tau-1,0,0) + q_t + p_t C \end{array} \right\} & \text{otherwise} \end{cases} \tag{3.2}$$

$$\text{for } t = k, \cdots, l, \tau = 0, \cdots, \left\lfloor \frac{d_{kl}}{C} \right\rfloor$$

$$G_k(t,\tau,1,0) = \begin{cases} \infty & \text{if } \tau C + \rho_{kl} \le d_{kt} \text{ or } \tau > t-k \\ \min\left\{ \begin{array}{l} G_k(t-1,\tau,1,0), \\ G_k(t-1,\tau-1,1,0) + q_t + p_t C, \\ G_k(t-1,\tau,0,0) + q_t + p_t \rho_{kl} \end{array} \right\} & \text{otherwise} \end{cases} \tag{3.3}$$

$$\text{for } t = k, \cdots, l, \tau = 0, \cdots, \left\lfloor \frac{d_{kl}}{C} \right\rfloor$$

$$G_k(t,\tau,0,1)(*) = \begin{cases} \infty & \text{if } \tau > t-k+1 \\ \min\left\{ \begin{array}{ll} G_k(t-1,\tau,0,1) & \text{if } z_v + \tau C > d_{kt}, \\ G_k(t-1,\tau,0,1) + g_v(d^{m\tau}_{kt} - \lfloor \frac{z_v}{C} \rfloor)C & \text{if } z_v + \tau C \le d_{kt}, \\ G_k(t-1,\tau-1,0,1) + q_t + p_t C & \text{if} z_v + \tau C > d_{kt}, \\ G_k(t-1,\tau,0,0) + g_t(d^{m\tau}_{kt} C + \rho_{kl}), & \\ G_k(t-1,\tau-1,0,0) + q_t + p_t C + g_t(d^{m\tau}_{kt} C + \rho_{kl}) & \end{array} \right\} & \text{otherwise} \end{cases} \tag{3.4}$$

$$\text{for } t = k, \cdots, l, \tau = 0, \cdots, \left\lfloor \frac{d_{kl}}{C} \right\rfloor,$$

$(*)$ put $v = t$ if $G_k(t,\tau,0,1) = G_k(t-1,\tau,0,0) + g_t(d^{m\tau}_{kt} C + \rho_{kl})$

or $G_k(t,\tau,0,1) = G_k(t-1,\tau-1,0,0) + q_t + p_t C + g_t(d^{m\tau}_{kt} C + \rho_{kl})$.

Note, in equations about $G_k(t,\tau,0,0)$ and $G_k(t,\tau,1,0)$, if $t = l$ and $\tau = \lfloor \frac{d_{kl}}{C} \rfloor$ the condition of ∞ will be $\lfloor \frac{d_{kl}}{C} \rfloor C < d_{kl}$ if $\rho_{kl} = 0$ or $\lfloor \frac{d_{kl}}{C} \rfloor C + \rho_{kl} < d_{kl}$. Also in equation about $G_k(t,\tau,0,1)$, the condition of $z_v + \tau C > d_{kt}$ will be $z_v + \tau C = d_{kl}$ for $t = l$.

The optimal value of regeneration interval $[k, l]$ then can be obtained by

$$\begin{aligned} &G_k(l, \tau^*, \theta^*, \chi^v(l)^*) \\ &\quad = \min\left\{ G_k\left(l, \left\lfloor \frac{d_{kl}}{C} \right\rfloor, 1, 0\right), G_k(l,\tau,0,1) \mid 0 \right. \\ &\quad \left. \le \tau \le \left\lfloor \frac{d_{kl}}{C} \right\rfloor \right\}. \end{aligned} \tag{3.5}$$

Note that if costs on every intervals $[k, l]$ with $1 \le k \le l \le n$ have been known, the original $LS - O$ can been solved by shortest path problem.

The dynamic program for each interval is $O(n^2)$, and as there are $O(n^2)$ intervals, we have:

Theorem 3.1 *There is an algorithm which solves LS-O whose running time is $O(n^4)$.*

4 Numerical implementation

In this section we give numerical tests about the algorithm suggested in Section 3. We program in C and the results are summerized in Table 1. Every (real) running time is the average of ten examples.

Table 1 Resuls of numerical implementation.

Length of regeneration intervals	200	400	600	800	1000
Running time (Sec.)	0.0719	0.0937	0.1137	0.1389	0.1702

CPU: Core Duo, 1.20GHz; RAM: 2.00GB; 32bit Windows Vista.

Table 2 An example of regeneration interval of lot-sizing problem with outsourcing.

Demands	4	6	2	2	2	6	6	2	5	2	5	2	4	2	3
Production costs	3	2	3	3	2	2	2	1	1	2	3	2	3	3	3
Set-up costs	17	16	17	18	16	19	16	18	15	17	15	17	15	17	18
Outsourcing costs	6	6	3	4	5	5	5	6	6	4	6	5	3	6	3
Production capacity:10; Optimal value: 177 (Working backwards for optimal solution)															
Production	10	10	0	0	0	0	0	0	10	0	0	0	0	0	0
Outsourcing	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0

Note the algorithm is very efficient in practice. One reason is that there are only plus and multiple operations. Given all parameters integers, no floats are needed in programming.

Finally, we give a small example generated in our prgramming.

Lot-sizing problem with constant capacities can be solved in $O(n^3)$ by some greedy scheme based on $q_t + p_t * C$ for all $k \leq t \leq l$ and feasibilities ([1]). With outsourcing, whether it can also be solved in $O(n^3)$ will be our next works.

References

[1] C.P.M. van Hoesel and A.P.M. Wagelmans, "An $O(T^3)$ Algorithm for the economic lot-sizing problem with constant capacities," *Manage. Sci.*, vol.42 the resul, pp.142–150, 1996.

[2] H.C. Hwang, W. Jaruphongsa, S. Cetinkaya, and C.Y. Lee, "Capacited dynamic lot-sizing problem with delivery/production time windows," *Oper. Res. Lett.*, vol.38, pp.408–413, 2010.

[3] L.A. Wolsey, "Lot-sizing with production and delivery time windows," *Math. Program.* Series A, vol.107, pp.471–489, 2006.

[4] V. Pochet and L.A. Wolsey, *Production Planning by Mixed Integer Pregramming*, Springer, 2006.

Ping ZHAN

Ping Zhan is an associate professor of Department of Communication and Business, Edogawa University. She received Ph.D. from Tsukuba University. Her main research interests include combinatorial optimization, mixed integer programming, algorithm design and implementation, and supply chain management. Research keywords contain submodular function and bisubmodualr function, majorization ordering, enumeration; minimum norm point. Recently she is interested in lot-sizing problems, they are typical mixed integer problems and have been studied extensively for a long time.