A Linear Time Algorithm for a Capacitated Multi-Item Scheduling Problem

Stan van Hoesel+

Kamalini Ramdas*

Alexander H.G. Rinnooy Kan+

\+ Econometric Institute
Erasmus Universiteit Rotterdam

* Decision Sciences Doctoral Program
The Wharton School
University of Pennsylvania

Abstract

We present a linear time algorithm for a multi-item, multi-period, capacitated production scheduling problem with holding costs.

Deterministic / Multi-item / Scheduling

We gratefully acknowledge the useful comments from Alan Hoffman. He provided the alternative proof that our algorithm delivers an optimal solution, by using the monge property.

Kindly address reprint order form to

Kamalini Ramdas
Department of Decision Sciences
The Wharton School
University of Pennsylvania
Philadelphia PA 19104

A LINEAR TIME ALGORITHM FOR A CAPACITATED MULTI-ITEM SCHEDULING PROBLEM

Stan van Hoesel*
Kamalini Ramdas**
Alexander H. G. Rinnooy Kan*

91-01-09

* Econometric Institute
Erasmus Universiteit
Rotterdam, The Netherlands

** Department of Decision Sciences
The Wharton School
University of Pennsylvania
Philadelphia, PA 19104-6366

1. Introduction

We consider a multi-item, multi-period, capacitated production scheduling problem with holding costs. The problem is to determine how much to produce of each item, in each period of the planning horizon, in order to minimize the total holding cost over all periods. Demand for each item is deterministic and known for all periods of the planning horizon. It is assumed that backlogging of demand is not allowed, and further, that production capacity over the planning horizon is enough to meet demand for all items. Set-up costs are assumed to be equal to zero. This problem may arise as a subproblem in a hierarchical problem formulation in which other costs are introduced at a higher level, eg. Aull and Ramdas [1] (see Section 5). It can be easily formulated as a transportation problem, and can therefore be solved using any standard transportation code, eg. Orlin [7]. In this paper, we present a faster algorithm that takes advantage of the special structure of the problem, thus leading to a reduction in complexity.

The remainder of this paper is organized as follows. In the next section, we first formulate the problem as a production scheduling problem, and then as an equivalent transportation problem. In Section 3, we provide a characterization of optimal solutions to this problem. In Section 4, we present an algorithm which computes the optimal solution. The complexity of our algorithm is shown to be O(NT), where N is the number of distinct items, and T is the number of periods in the planning horizon. In Section 5 we discuss extensions, an alternative proof that our algorithm provides an optimal solution, and applications.

2. Formulation

If set-up costs are included, the problem described in Section 1 is a multi-item capacitated lot sizing problem. A review of the literature on lot-sizing problems can be found in Bahl, Ritzman, and Gupta [2]. The multi-item capacitated lot-sizing problem is NP hard (for a complexity analysis, see Florian, Lenstra, and Rinnooy Kan [5]). Several heuristics have been devised for this problem; a comparison of heuristics can be found in Maes and Van Wassenhove [6]. In our approach to this problem, the simplifying factor is that set-up costs are ignored. The following is a formulation of our problem as a multi-item capacitated lot sizing problem without set-up costs.

Problem 1.1

Minimize

$$\sum_{j=1}^{N}\sum_{t=1}^{T} h_j I_{jt}$$

subject to

$$\sum_{j=1}^{N} x_{jt} \le C_t \qquad t=1,\ldots T$$

$$I_{j,t-1} + x_{j,t} - I_{j,t} = d_{j,t} \qquad j = 1,..N,\ t = 1,..T,$$

$$x_{j,t} \geq 0,\ I_{j,t} \geq 0, \qquad j = 1,..N,\ t = 1,..T.$$

where

$x_{j,t}$ = number of units of item j produced in period t,

$I_{j,t}$ = ending inventory of item j in period t,

$d_{j,t}$ = demand for item j in period t,

h_j = holding cost per unit per period for item j, and

C_t = total production capacity in period t.

By assumption, we have,

$$\sum_{\tau=1}^{t} C_\tau \geq \sum_{\tau=1}^{t}\sum_{j=1}^{N} d_{j,\tau} \qquad t = 1,...T.$$

Problem 1.2 below is a plant location reformulation of Problem 1.1. This reformulation can also be found in Barany, Van Roy, and Wolsey [3], and in Wagelmans, Van Hoesel, and Kolen [8].

Problem 1.2

Minimize

$$\sum_{j=1}^{N}\sum_{t=1}^{T}\sum_{s=t}^{T} (s-t) h_j y_{j,t,s}$$

subject to

$$\sum_{j=1}^{N}\sum_{s=t}^{T} y_{j,t,s} \leq C_t \qquad t = 1,..T$$

$$\sum_{t=1}^{s} y_{j,t,s} = d_{j,s} \qquad j = 1,..N,\ s = 1,..T$$

$$y_{j,t,s} \geq 0, \qquad j = 1,..N,\ t,\ s = 1,..T.$$

where

y_{jts} = number of units of item j produced in period t and used to meet demand in period s, and other symbols have the same meanings as in the previous formulation.

Note that the above formulations are related by the following equation:

$$x_{j,t} = \sum_{s=t}^{T} y_{j,t,s} \qquad j = 1,..N,\ t = 1,..T.$$

Define

$$d_D = \sum_{t=1}^{T} C_t - \sum_{j=1}^{N}\sum_{t=1}^{T} d_{j,t}.$$

If $d_D > 0$, we have excess capacity. For convenience in our analysis, we eliminate this excess capacity by introducing an artificial item, say N+1, with zero holding cost, i.e., $h_{N+1} = 0$; further, we set $d_{N+1,T} = d_D$, and $d_{N+1,t} = 0$ for all other t. In this case, Problem 1.2 can be reformulated as shown below.

Problem 1.3

Minimize

$$\sum_{j=1}^{N}\sum_{t=1}^{T}\sum_{s=t}^{T}(s-t)h_j y_{j,t,s}$$

subject to

$$\sum_{j=1}^{N+1}\sum_{s=t}^{T} y_{j,t,s} = C_t \qquad t = 1,..T$$

$$\sum_{t=1}^{s} y_{j,t,s} = d_{j,s} \qquad j = 1,..N+1,\ s = 1,..T$$

$$y_{j,t,s} \ge 0 \qquad j = 1,..N+1,\ t,\ s = 1,..T.$$

where all symbols have the same meanings as in Problem 1.2. It is evident that Problem 1.3 is a transportation problem; its network structure is depicted in Figure 1. Henceforth we will deal only with Problem 1.3. Further, we assume without loss of generality that inventory costs are different for all items. This can be accomplished by identifying items with equal inventory costs.

3. Characterization of an Optimal Solution to Problem 1.3

We present a property which is both necessary and sufficient for optimality in Problem 1.3.

Property 1

For any distinct items i and j, if there is production of item i in period t_1 to meet demand in period $t_2 \geq t_1$, and there is production of item j in period t_1' to meet demand in period $t_2' \geq t_1'$ such that $t_1' < t_1 \leq t_2'$, then $h_j < h_i$ (see Figure 2).

Theorem 1

Property 1 is a necessary as well as a sufficient condition for optimality in Problem 1.3.

Proof

We first prove that Property 1 is a necessary condition for optimality. To do this, we start with a solution that does not satisfy Property 1, and show by way of an exchange argument that a better solution exists.

Let S be a feasible solution to Problem 1.3, such that S does not satisfy Property 1. That is, there exist distinct items i and j, such that in solution S, item i is produced in period t_1 to meet demand in period $t_2 \geq t_1$ and item j is produced in period t_1' to meet demand in period $t_2' \geq t_1'$, such that $t_1' < t_1 \leq t_2'$, and $h_i < h_j$.

The cost of producing one unit of each item in this case is

$A = h_i(t_2 - t_1) + h_j(t_2' - t_1')$.

However, we can exchange production since $t_1, t_1' \leq t_2, t_2'$. Suppose that we produce one unit of item i in period t_1' to meet demand in period t_2 and we produce one unit of item j in period t_1 to meet demand in period t_2'.

In this case, the cost of producing one unit of each item is

$B = h_i(t_2 - t_1') + h_j(t_2' - t_1)$.

Since all other costs remain unchanged, the difference in costs caused by this exchange is A-B. Now A-B is equal to $(h_j - h_i)(t_1 - t_1')$, which is positive, so the modified solution is cheaper than S. Hence, S cannot be optimal.

We prove that Property 1 is a sufficient condition for optimality by assuming the contrary, and arriving at a contradiction. Assume that there exists a non-optimal solution to Problem 1.3, that satisfies Property 1. We will refer to this solution as S. S may be depicted as a set of flows in a directed network, as shown in Figure 1. Let S^* denote an optimal solution to Problem 1.3. Further, let $S-S^*$ denote the set of flows obtained by subtracting the flow along each arc of the network as dictated by S^* from the corresponding flow as dictated by S. As the flows along the demand and supply arcs are the same in S and S^*, these arcs will have zero flow in $S-S^*$. Hence, the flows $S-S^*$ must consist of a set of circuits, where a circuit is a set of flows of identical absolute magnitude, such that each flow is on a distinct arc, the arcs form a closed loop, and each node in the circuit has exactly one incoming arc, and one outgoing arc.

Let C be an arbitrarily chosen circuit in S. Let t_1 be the last time period covered by circuit C. C must contain four arcs as depicted in Figure 3 below. The arcs are numbered 1, 2, 3, and 4. Note that arc 1 corresponds to item i, whose holding cost is h_i, while arc 4 corresponds to item j, whose holding cost is h_j.

a) From arc 1 it follows that in Solution S, at least one unit of demand for item i in period t_1 is satisfied from production in a period prior to t_1.

b) From arc 2 it follows that in Solution S^*, at least one unit of item i is produced in period t_1, for consumption in or after period t_1.

c) From arc 3 it follows that in Solution S, at least one unit of item j is produced in period t_1, for consumption in or after period t_1.

d) From arc 4 it follows that in Solution S^*, at least one unit of demand for item j in period t_1 is satisfied from production in a period prior to t_1.

From a) and c) and the fact that S satisfies Property 1, it follows that $h_i < h_j$.

From b) and d) and the fact that S^* satisfies Property 1, it follows that $h_i > h_j$.

We have thus arrived at a contradiction. Hence, Property 1 is a sufficient condition for optimality.

4. Optimal Algorithm for Problem 1.3

We present a greedy algorithm for Problem 1.3. The solution provided by this algorithm is shown to satisfy Property 1 of Section 3. The complexity of the algorithm is shown to be O(NT), where N is the number of distinct items, and T is the number of periods in the planning horizon.

Algorithm

The following conventions are used. Items are arranged in order of increasing holding costs. We refer to the ordered items as $i_1, \ldots i_N$. It is assumed that total capacity is exactly equal to total demand (if this is not the case, we introduce an additional item, say i_0, as discussed in Section 2 above). Further, we assume that if $C_{t'}$ and $d_{it''}$ ($t' \leq t''$) are decreased by x, then x units of capacity in period t' are used to satisfy demand of item i in period t''. The algorithm presented below treats the most expensive items first. For each item under consideration, the production periods are determined in backward fashion, in order to guarantee the complexity of O(NT).

```
For i:= N downto 1
do
    t', t'' := T
    while t' > 0
    do
        x := min(C_t' , d_it''); C_t' := C_t' - x; d_it'' := d_it'' - x;
        if d_it'' = 0 then
            t'' := t'' - 1
        endif
        if C_t' = 0 then
            t' := t' - 1
        endif
        if t' > t'' then
```

```
            t' := t''
        endif
    enddo
enddo
```

The above algorithm ensures that in any period t', the capacity $C_{t'}$ is used first to fulfil the costliest demand as yet not met in periods greater than or equal to t'. So, if capacity from some period t'' less than t' is used to fulfil demand in a period greater than or equal to t', it must be for an item at most as costly as the items whose demand is satisfied from production in t'. Hence, this algorithm satisfies Property 1. Further, the complexity of this algorithm is clearly O(NT).

Concluding Remarks

In this section, we discuss an extension of our algorithm, and an alternative proof that it provides an optimal solution to Problem 1.3. This is followed by a discussion of possible applications.

In Problem 1.3 of Section 2, we assumed that production of each unit of item j, j=1,..N+1, requires a single unit of capacity. Now consider the more general problem in which production of a unit of item j, j=1,..N+1, requires a_j units of capacity. Assuming that production of a unit of any item can be split accross several periods, this problem can be shown to be equivalent to our Problem

1.3 through the following transformation of demands and holding costs :

$d_{jt} := a_j d_{jt}$, and $h_j := h_j/a_j$, for j=1,..N+1.

The Monge structure of the cost matrix associated with Problem 1.3 of Section 2 provides an alternative method to prove that our algorithm provides an optimal solution. For a transportation problem with allowable variables A and excluded variables E, a monge sequence is a one-to-one numbering $f:A \rightarrow \{1,..,|A|\}$. A cost matrix $C = (c_{ij})$ is said to be consonant with f if

$r < s,t$ and

$f^{-1}(r) = (i,j),\ f^{-1}(s) = (i,j'),\ f^{-1}(t) = (i',j)$

$\Rightarrow$

(1) (i',j') is in A and

(2) $c_{ij} + c_{i'j'} \leq c_{ij'} + c_{i'j}$.

For any monge sequence f, the f-greedy algorithm successively maximizes, $X_{f^{-1}(1)}$, $X_{f^{-1}(2)}$, etc, subject to demand and supply constraints.

It is easy to show that if a cost matrix C is consonant with a monge sequence f, then for all row and column sums, the f-greedy algorithm

a) produces a feasible solution if there is one, and

b) produces an optimal solution if there is a feasiblle solution.

The cost matrix associated with Problem 1.3 is consonant with the sequence followed by our algorithm; this completes the proof.

Aull and Ramdas [1] use two alternative approaches to model a multi-item, multi-period, deterministic production scheduling problem characterised by extremely high, sequence dependent set-up costs, and relatively low inventory holding costs. Our algorithm may be used to enhance both their approaches. In their article, the authors group end items into families such that changing from production of one item to another within the same family requires negligible set-up, whereas a major set-up cost is incurred in changing from production of one family to another. Items within the same family may have varying inventory holding costs. The authors assume that the number of machines available is fixed, and is enough to meet demand over the planning horizon without backlogging. Machine productivity is allowed to vary across, but not within product families. The number of machines assigned to each family, in each period of the planning horizon, must be integral, while the number of machines assigned to each item may take on continuous values. In their first approach, Aull and Ramdas formulate a mixed integer model which contains variables for aggregate as well as end-item production. 'Bridging' constraints are used to relate end item production to aggregate family production. This model minimizes the sum of set-up cost across families, and holding cost over end items. The authors also present a two level hierarchical formulation which considers set-up costs at the family level, and holding costs at the item level. Our algorithm can be used to solve their lower level problem. Also, our algorithm may be used within a decomposition method such as Benders' partitioning [4], to solve their mixed integer formulation.

REFERENCES

[1] Aull, R. and Ramdas, K., "A Hierarchical Production Planning Model with Feedback Mechanisms", Journal of Operations Management, forthcoming.

[2] Bahl, H.C., Ritzman, L.P., and Gupta, N.D., "Determining Lot Sizes and Resource Requirements: A Review", Operations Research, Vol. 35, No. 3, May - June 1987, pp. 329 - 345.

[3] Barany, I., Van Roy, T.J., and Wolsey, L.A., "Strong Formulations for Multi-Item Capacitated Lot-Sizing", Management Science, Vol.30, No. 10, October 1984, pp. 1255 - 1261.

[4] Benders, J.F., "Partitioning Procedures for Solving Mixed Variables Programming Problems", Numerische Mathematic 4, 1962, pp. 238 - 252.

[5] Florian, M., Lenstra, J.K., and Rinnooy Kan, A.H.G., "Deterministic Production Planning: Algorithms and Complexity", Management Science, Vol. 26, 1980, pp. 12 - 20.

[6] Maes, J. and Van Wassenhove, L.N., "Multi Item Single Level Capacitated Dynamic Lotsizing Heuristics: A Computational Comparizon (Part I: Static Case)", IIE Transactions, June 1986, pp. 114 - 123.

[7] Orlin, J.B., "A Faster Strongly Polynomial Minimum Cost Flow Algorithm", Sloan W.P. No. 3060-89-MS, August 1989, Sloan School of Management, M.I.T .

[8] Wagelmans, A., van Hoesel, S., and Kolen, "An O(n logn) algorithm for economic lot-sizing," Operations Research, forthcoming.

Figure 1

Network Structure of Problem 1.3

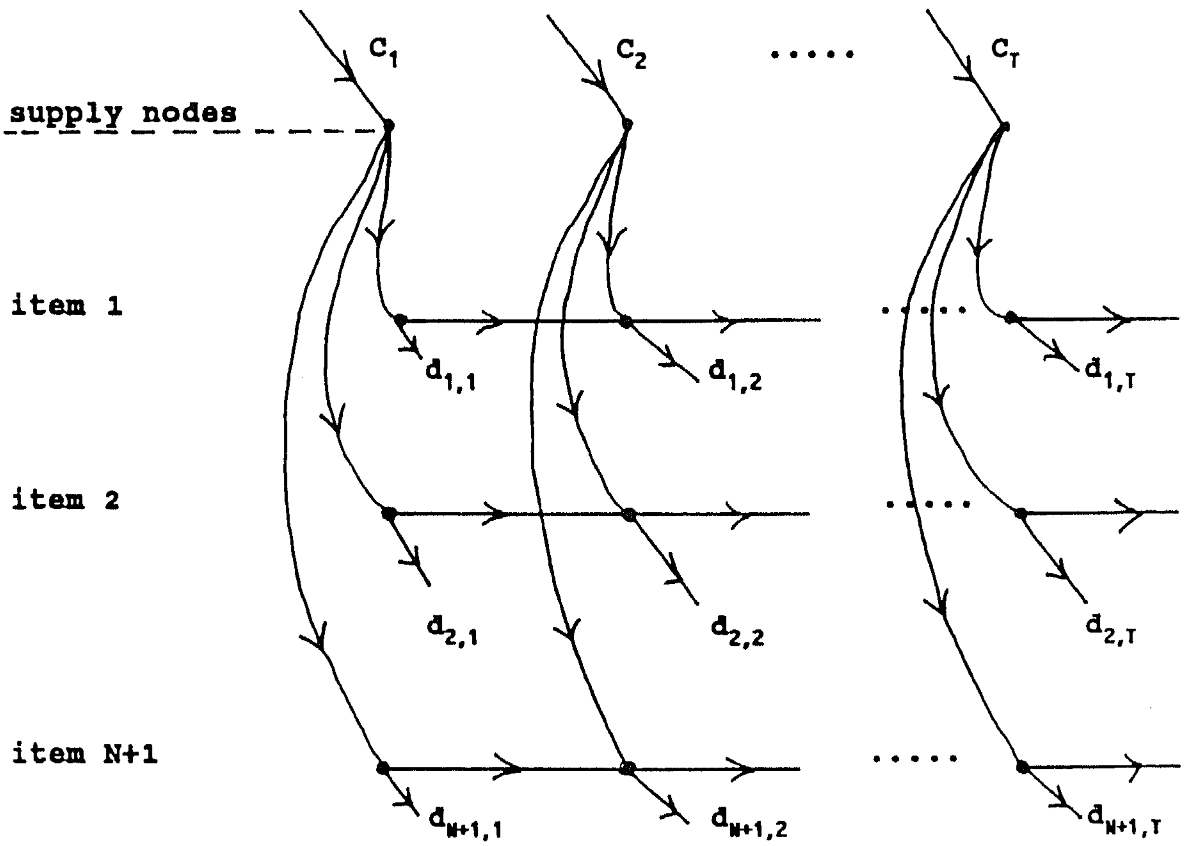

Figure 2

Representation of Property 1

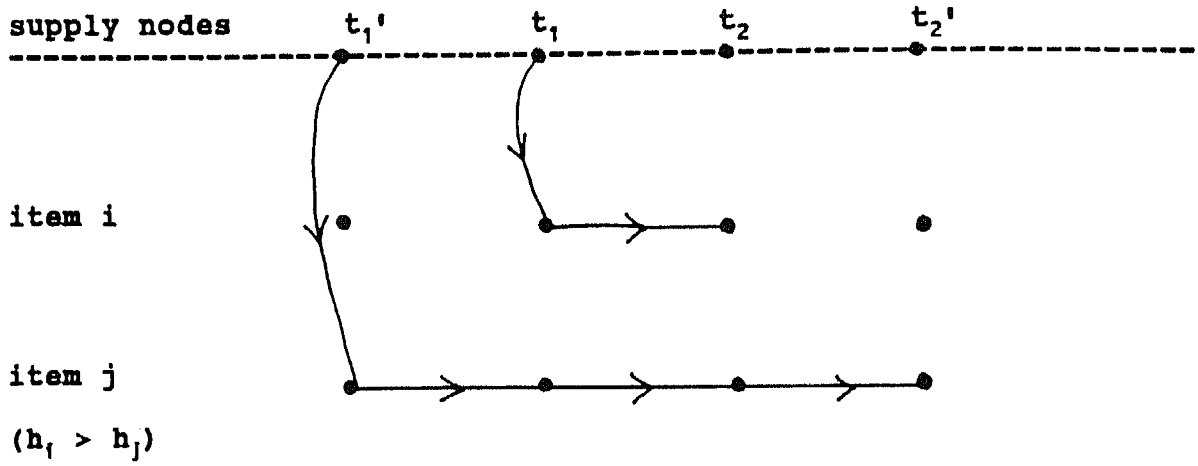

Figure 3

A Depiction of the Circuit C