Foundations of Inventory Management **Fall 2022, NUS**

Lecture 5: Economic Lot Sizing Models

Lecturer: Zhenyu Hu

5.1 The Wagner-Whitin Model

Compared to the EOQ model, here we consider a discrete-time problem with time-varying demands. This generalization is also known as the Wagner-Whitin Model (Wagner and Whitin, 1958). The planning horizon is divided into $t = 1, ..., T$ discrete periods. Demand in each period t is $d_t > 0$. Again, there is no lead time in delivering the product and each order incurs a fixed ordering cost K and a variable ordering cost c. On-hand inventory incurs holding cost h per-unit held, per period. As a convention, we assume that all ordering and demand occur at the start of the period while inventory holding cost is charged on the amount on hand at the end of the period.

Let q_t be the amount ordered in period t and x_t be the amount of the inventory *at the end* of period t. We let x_0 denote the initial inventory level which is assumed to be zero here. The ordering cost incurred in period t is then $cq_t + K \cdot 1_{\{q_t>0\}}$. We can then formulate the problem as

$$
\begin{aligned}
\min \ & \sum_{t=1}^{T} \left[cq_t + K \cdot 1_{\{q_t>0\}} + hx_t\right] \\
\text{s.t. } & x_t = x_{t-1} + q_t - d_t, t = 1, 2, ..., T, \\
& x_0 = 0, \\
& x_t, q_t \geq 0, t = 1, 2, ..., T.
\end{aligned}
\tag{5.1}
$$

One can see that it is never optimal to have left-over inventory at the end of the planning horizon, i.e., $x_T > 0$. By noting that $x_t = \sum_{\tau=1}^{t}(q_\tau - d_\tau)$, $x_T = 0$ then implies $\sum_{t=1}^{T} cq_t = \sum_{t=1}^{T} cd_t$, which is a constant independent of the schedule of orders.[1] Hence, by letting z_t be the binary variable denoting whether we order in period t or not, we can further reformulate problem (5.1) as the following integer program:

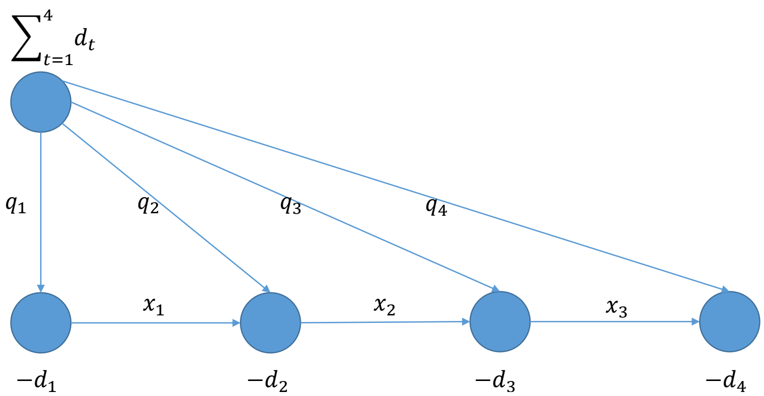
$$
\begin{aligned}
\min \ & \sum_{t=1}^{T} [Kz_t + hx_t] \\
\text{s.t. } & x_t = x_{t-1} + q_t - d_t, t = 1, 2, ..., T, \\
& x_0 = 0, \\
& q_t \leq M_t z_t, t = 1, ..., T \\
& x_t, q_t \geq 0, z_t \in \{0, 1\}, t = 1, 2, ..., T.
\end{aligned}
$$

Here, we can let $M_t = \sum_{\tau=t}^{T} d_\tau$. Again, one can observe that the "zero inventory property" holds here, i.e., $q_t \cdot x_{t-1} = 0$ for $t = 1, ..., T$. Intuitively, if $q_t \cdot x_{t-1} > 0$ for some period t, then we can reduce the left-over inventory level x_{t-1} by shifting some inventory ordered in various periods prior to t to period t, i.e., increasing q_t while keeping $q_t + x_t$ fixed. In this way, we do not change the inventory held for satisfying demands from t to T but we can reduce some holding cost incurred prior to t. Note that this argument

[1] The stationarity of the ordering cost c matters here. With non-stationary ordering cost, then one has incentive to order more in the period with cheaper ordering cost, and we have to keep the ordering cost term in our formulation.

requires the stationarity of the ordering cost c; otherwise shifting orders across periods also affect ordering cost.

A more general observation is that problem (5.1) is a special case of a minimum concave-cost network flow problem. We construct a network by creating a supply node with supply $\sum_{t=1}^{T} d_t$ and T demand nodes each with demand $d_t, t = 1, ..., T$. The supply node has an arc to each demand node t on which we send a flow of q_t. Demand node $t-1$ has an arc to demand node t and the flow on this arc is x_{t-1}. The figure below illustrates such a network for $T = 4$. The flow balance constraint in this network then exactly corresponds to the inventory balancing constraint $x_t = x_{t-1} + q_t - d_t$ in (5.1). Finally, note that the objective function in (5.1) is a concave function in q_t, x_t. For minimum concave-cost network flow problem,



there exists an optimal solution that is an extreme point of the feasible region, and recall that a flow vector in a uncapacitated network is a basic solution if and only if it forms a spanning tree of the network (see Theorem 7.4 in Bertsimas and Tsitsiklis, 1997). For the underlying network of problem (5.1), observe that in a spanning tree, we must have $q_t \cdot x_{t-1} = 0$; otherwise the network contains a cycle.

An implication of the zero-inventory property is that an order is always of size that equals to the total demands during an integer number of subsequent periods, i.e., $q_t = \sum_{\tau=t}^{t+\ell} d_\tau$ for some nonnegative integer ℓ, and deciding q_t is the same as deciding how many periods of demands to satisfy. This motivates the following dynamic programming algorithm.

Let

$$\begin{aligned} V_t = \min \ & \sum_{\tau=t}^{T} \left[K \cdot 1_{\{q_\tau > 0\}} + h x_\tau\right] \\ \text{s.t. } & x_\tau = x_{\tau-1} + q_\tau - d_\tau, \tau = t, ..., T, \\ & x_{t-1} = 0, \\ & x_\tau, q_\tau \geq 0, \tau = t, ..., T, \end{aligned}$$

be the minimum cost-to-go from period t to T given that the inventory level drops to zero at the end of period $t-1$. At period t, if we decide to make an order that satisfies demands from period t to period $t'-1$ with $t+1 \leq t' \leq T+1$, i.e., $q_t = \sum_{\tau=t}^{t'-1} d_\tau$, then we know $x_{t'-1} = 0$ and we incur a total cost of

$$K + h \sum_{\tau=t}^{t'-1} (\tau - t) d_\tau$$

during periods t to $t'-1$. It follows that V_t must satisfy the Bellman equation:

$$V_t = \min_{t+1 \le t' \le T+1} \left\{ K + h \sum_{\tau=t}^{t'-1} (\tau - t) d_\tau + V_{t'} \right\}$$

with $V_{T+1} = 0$. We can further interpret the dynamic program developed here as a shortest path problem. The set of nodes is $\mathcal{N} = \{1, 2, ..., T+1\}$ and the arc set is $\mathcal{A} = \{(i,j) | i = 1, ..., T, j = i+1, ..., T+1\}$, i.e., we have an arc (i,j) for every $i < j$. Choosing to travel along the arc (i,j) means that we are ordering in period i with an order quantity that satisfies all demands from period i to $j-1$. The cost c_{ij} on the arc (i,j) is then

$$c_{ij} = K + h \sum_{\tau=i}^{j-1} (\tau - i) d_\tau .$$

The figure below illustrates the graph for the shortest path problem when $T = 3$. The computational time

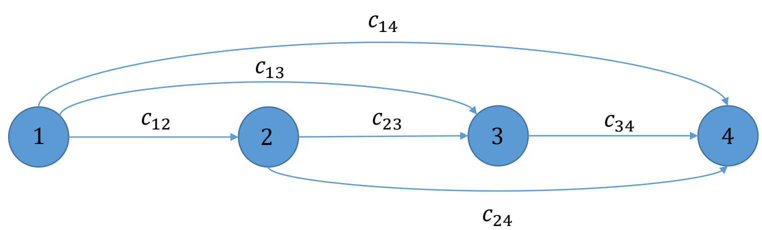


Figure 5.1: Shortest path representation of economic lot sizing

for finding the values of all c_{ij} is $O(T^2)$ and the running time of Dijkstra's algorithm in finding the shortest path is also bounded by $O(T^2)$. So we can solve the Wagner-Whitin model in $O(T^2)$ time.

5.2 More on Extreme Point Perspective

The zero-inventory property $q_t \cdot x_{t-1} = 0$ provides a way in characterizing all extreme points of the polyhedron in (5.1). Here, we describe one characterization using interval partitions to obtain a linear programming reformulation of (5.1).[2] At any extreme point, suppose $1 \le i_1 < i_2 < ... < i_m \le T$ are indices of zero components in x, i.e., $x_{i_1} = ... = x_{i_m} = 0$. Let $i_0 = 0$ and note that $i_m = T$. Then the indices $i_0, i_1, ..., i_m$ partitions the periods $\{1, ..., T\}$ into disjoint sets $\{i_0+1, ..., i_1\}, \{i_1+1, ..., i_2\}, ..., \{i_{m-1}+1, ..., i_m\}$

We can represent a partition of the set $\{1, ..., T\}$ using a set of binary variables $\{t_{ij} : i = 1, ..., T, j = i, ..., T\}$ with $t_{ij} = 1$ indicating that the set $\{i, i+1, ..., j\}$ is in the partition. The set of binary variables $\{t_{ij} : i = 1, ..., T, j = i, ..., T\}$ represent a partition if and only if

$$\sum_{i=1}^{k} \sum_{j=k}^{T} t_{ij} = 1, \ \forall k = 1, ..., T. \tag{5.2}$$

That is, each index k is covered by one and only one interval. Indeed, given a partition $\{i_0+1, ..., i_1\}, \{i_1+1, ..., i_2\}, ..., \{i_{m-1}+1, ..., i_m\}$, we can let $t_{i_\ell+1, i_{\ell+1}} = 1$ for $\ell = 0, ..., m-1$, and $t_{ij} = 0$ otherwise. Then, for

[2]The same idea described here is used in obtaining a tractable reformulation of a two-stage distributionally robust optimization problem in Mak et al. (2015) for an appointment scheduling problem.

any $\ell = 0, ..., m-1$ and any $k \in \{i_\ell + 1, ..., i_{\ell+1}\}$, by our construction we must have

$$\sum_{i=1}^{k} \sum_{j=k}^{T} t_{ij} = t_{i_\ell+1, i_{\ell+1}} = 1.$$

Conversely, given a feasible solution to (5.2), we can iteratively find the partition by first initializing i_1 to be the index of the unique binary variable $t_{1,i_1} = 1$ in the constraint (letting $k = 1$ in (5.2))

$$\sum_{j=1}^{T} t_{ij} = 1.$$

Given i_ℓ, we can find $i_{\ell+1}$ to be the index of the unique binary variable $t_{i_\ell+1, i_{\ell+1}} = 1$ in the constraint (letting $k = i_\ell + 1$ in (5.2))

$$\sum_{i=1}^{i_\ell+1} \sum_{j=i_\ell+1}^{T} t_{ij} = \sum_{j=i_\ell+1}^{T} t_{i_\ell+1,j} = 1,$$

where we used the fact that $t_{ij} = 0$ for $i = 1, ..., i_\ell$ and $j = i_\ell + 1, ..., T$ since $t_{i_{\ell-1}+1, i_\ell} = 1$.

For $t_{ij} = 1$, i.e., the set $\{i, i+1, ..., j\}$ is in the partition, the ordering quantity and inventory level during the periods $i, i+1, ..., j$ can then be uniquely determined as $q_i = \sum_{\tau=i}^{j} d_\tau, q_k = 0, k = i+1, ..., j$, and $x_k = \sum_{\tau=k+1}^{j} d_\tau, k = i, ..., j$.

Consider the following generalization of problem (5.1):

$$\begin{aligned} \min \ & \sum_{t=1}^{T} C_t(q_t, x_t) \\ \text{s.t. } & x_t = x_{t-1} + q_t - d_t, t = 1, 2, ..., T, \\ & x_0 = 0, \\ & x_t, q_t \geq 0, t = 1, 2, ..., T, \end{aligned}$$

where $C_t(q_t, x_t)$ is a general concave function in q_t, x_t. We can then arrive at the following equivalent integer programming formulation

$$\begin{aligned} \min \ & \sum_{i=1}^{T} \sum_{j=i}^{T} \sum_{k=i}^{j} C_k(q_k, x_k) t_{ij} \\ \text{s.t. } & \sum_{i=1}^{k} \sum_{j=k}^{T} t_{ij} = 1, \ \forall k = 1, ..., T, \\ & t_{ij} \in \{0, 1\}. \end{aligned} \tag{5.3}$$

Here, $\sum_{k=i}^{j} C_k(q_k, x_k) = C_i(\sum_{\tau=i}^{j} d_\tau, \sum_{\tau=i+1}^{j} d_\tau) + \sum_{k=i+1}^{j} C_k(0, \sum_{\tau=k+1}^{j} d_\tau)$ are constants that can be computed offline. In the special case when $C_k(q_k, x_k) = K \cdot 1_{\{q_k > 0\}} + hx_k$, we simply have $\sum_{k=i}^{j} C_k(q_k, x_k) = K + h \sum_{\tau=i}^{j} (\tau - i) d_\tau = c_{i,j+1}$. One can show that the matrix associated with the constraint $\sum_{i=1}^{k} \sum_{j=k}^{T} t_{ij} = 1$ has the so-called consecutive-ones property (see p.279 in Schrijver, 1998), and thus is totally unimodular. Hence, the above formulation can be solved as a linear program.

Alternatively, one can also view (5.3) as a reformulation of the shortest path problem on the network illustrated in Figure 5.1. In particular, we can let $x_{i,j+1} = t_{ij}$ denote whether we travel from i to $j+1$ and reformulate (5.3) as:

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} \\
\text{s.t.} \quad & \sum_{j=i+1}^{T+1} x_{ij} - \sum_{j=1}^{i-1} x_{ji} = \begin{cases} 1 & \text{if } i=1 \\ -1 & \text{if } i=T+1 \\ 0 & \text{otherwise} \end{cases} \\
& x_{ij} \in \{0,1\}.
\end{aligned}$$

To see the equivalence, note here that for $k = 1, ..., T$

$$\sum_{i=1}^{k}\sum_{j=k}^{T} t_{ij} = \sum_{i=1}^{k}\left(\sum_{j=i+1}^{T+1} x_{ij} - \sum_{j=1}^{i-1} x_{ji}\right).$$

That is, $\sum_{i=1}^{k}\sum_{j=k}^{T} t_{ij}$ can be interpreted as the outflow from the set of nodes $\{1, ..., k\}$ to all other nodes minus the inflow into the set of nodes $\{1, ..., k\}$ (which is zero).

5.3 Capacitated System

Consider now at each period t, there is a capacity constraint u_t on the ordering quantity q_t, and we consider the following generalization of problem (5.1)

$$\begin{aligned}
\min \quad & \sum_{t=1}^{T} \left[c_t q_t + K_t \cdot 1_{\{q_t>0\}} + h_t x_t\right] \\
\text{s.t.} \quad & x_t = x_{t-1} + q_t - d_t, t = 1, 2, ..., T, \\
& x_0 = 0, \\
& 0 \le x_t, 0 \le q_t \le u_t, t = 1, 2, ..., T.
\end{aligned} \tag{5.4}$$

It is easy to see that the zero inventory property will no longer hold since with capacity constraints one may be forced to gradually build its inventory to prepare for future demands.

Example 5.1 *Consider $d_1 = d_2 = ... = d_{T-1} = 0$ and $d_T = T$, and $u_1 = ...u_T = 1$. In this case, there is a unique feasible production plan:*

$$q_1 = q_2 = ... = q_T = 1.$$

One can see that problem (5.4) has a feasible solution if and only if $\sum_{\tau=1}^{t} u_\tau \ge \sum_{\tau=1}^{t} d_\tau$ for $\tau = 1, ..., T$, and we assume this condition holds for the rest of this section.

Problem (5.4) can still be solved via a dynamic programming algorithm. Let

$$\begin{aligned}
V_t(x) = \min \quad & \sum_{\tau=t}^{T} \left[c_\tau q_\tau + K_\tau \cdot 1_{\{q_\tau>0\}} + h_\tau x_\tau\right] \\
\text{s.t.} \quad & x_\tau = x_{\tau-1} + q_\tau - d_\tau, \tau = t, ..., T, \\
& x_{t-1} = x, \\
& 0 \le x_\tau, 0 \le q_\tau \le u_\tau, \tau = t, ..., T,
\end{aligned}$$

be the minimum cost-to-go from period t to T given that the inventory level is x at the end of period $t-1$. The Bellman equation can then be written as

$$V_t(x) = \min_{(d_t - x)^+ \le q_t \le u_t} \left\{ c_t q_t + K_t \cdot 1_{\{q_t > 0\}} + h_t(x + q_t - d_t) + V_{t+1}(x + q_t - d_t) \right\}$$

The number of possible states per period is bounded by $\sum_{t=1}^T d_t$ and the number of actions per period is bounded by u_t. Hence, the total computational time is $O\left(\sum_{t=1}^T \left(\sum_{t=1}^T d_t u_t\right)\right) = O\left(\left(\sum_{t=1}^T d_t\right)\left(\sum_{t=1}^T u_t\right)\right)$. Note, however, that like knapsack problem, the dynamic programming algorithm here is only a pseudo-polynomial time algorithm since the running time is not polynomial in the number of bits needed to describe the instance (which is $\log(\sum_{t=1}^T d_t)$).

Indeed, problem (5.4) is in general NP-hard as we show next. We use a reduction from the subset-sum problem, which is known to be NP-complete. Given a set of integer numbers $\{u_1, u_2, ..., u_T\}$ with $\mathcal{N} = \{1, 2, ..., T\}$ and a target integer A, the decision problem of subset-sum asks whether there exists a subset $\mathcal{S} \subseteq \mathcal{N}$ such that $\sum_{i \in \mathcal{S}} u_i = A$.[3]

Now we construct a specific instance of the problem (5.4) with $t = 0, 1, ..., T$, and we let

$$h_t = 0, d_t = A, K_t = 1, \ \forall t = 0, 1, 2, ..., T.$$

The capacity at period zero $u_0 = A \cdot T$ and capacities from period 1 to T are given by $\{u_1, u_2, ..., u_T\}$. The ordering cost at period zero $c_0 = 0$ and we let $c_t = 1 - 1/u_t$, $t = 1, ..., T$.

Clearly, it is optimal to let $q_0 = A \cdot T$ to satisfy demands from period 0 to $T-1$, and all orders from 1 to T are used to satisfy demand in the last period. Any feasible solution then must satisfy

$$\sum_{t=1}^T q_t = A, \ 0 \le q_t \le u_t, t = 1, ..., T.$$

Note that the total ordering cost at period t is

$$c_t q_t + K_t \cdot 1_{\{q_t > 0\}} = \begin{cases} 0 & \text{if } q_t = 0 \\ q_t + 1 - \frac{q_t}{u_t} & \text{if } 0 < q_t \le u_t. \end{cases}$$

Observe that $c_t q_t + K_t \cdot 1_{\{q_t > 0\}} \ge q_t$ with strict inequality if and only if $0 < q_t < u_t$. As a result, the total cost of problem (5.4) is lower bounded by $1 + \sum_{t=1}^T q_t = 1 + A$ and is equal to $1 + A$ if and only if $q_t \in \{0, u_t\}$ for all $t = 1, ..., T$. That is, if we can solve problem (5.4) in polynomial time, then given any $\{u_1, u_2, ..., u_T\}$ and A, we can determine whether there exists a subset $\mathcal{S} \subseteq \mathcal{N}$ such that $\sum_{i \in \mathcal{S}} u_i = A + 1$ as well, which contradicts with the fact that subset-sum is a NP-complete problem. This shows that (5.4) is NP-hard.

One is referred to Florian et al. (1980) for more discussions on the complexity issue of problem (5.4) and various computational algorithms for certain special cases.

References

Bertsimas, D. and J. N. Tsitsiklis (1997). *Introduction to linear optimization*, Volume 6. Athena Scientific Belmont, MA.

Florian, M., J. K. Lenstra, and A. Rinnooy Kan (1980). Deterministic production planning: Algorithms and complexity. *Management science 26*(7), 669–679.

[3]Observe that the subset-sum problem can be viewed as a special case of knapsack problem with $v_t = w_t = u_t$ and $W = A$.

Mak, H.-Y., Y. Rong, and J. Zhang (2015). Appointment scheduling with limited distributional information. *Management Science 61*(2), 316–334.

Schrijver, A. (1998). *Theory of linear and integer programming.* John Wiley & Sons.

Wagner, H. M. and T. M. Whitin (1958). Dynamic version of the economic lot size model. *Management science 5*(1), 89–96.