Large-Scale Network Revenue Management Models

The only model we have seen thus far that can efficiently handle network revenue management problems is the virtual nesting model. This model is compatible with the "protection level viewpoint" of the single-leg revenue management models. In particular, if we use the virtual nesting model, then the shift from a single-leg revenue management model to a network revenue management model only requires bucketing the different itineraries. After this, we can use traditional single-leg network revenue management models that either use the optimal protection levels computed through a dynamic programming formulation or near-optimal protection levels computed through EMSR-type heuristics. For these reason, the virtual nesting model is widely used in the airline industry.

We now focus on the bid-price based network revenue models and propose different methods to compute the bid-prices. The advantages of bid-price based network revenue models are that they do not require bucketing the itineraries and they usually provide better policies than the virtual nesting model. However, computing good bid-prices is a research problem that is still under investigation.

1. Network revenue management setting

We begin by recalling the problem setup and some of the notation that we used earlier. We have a set of flight legs that can be used to satisfy the itinerary requests that arrive randomly over time. At each time period, a request for an itinerary arrives. If we accept the itinerary request, then we generate a revenue and consume the capacities on the relevant flight legs. A rejected itinerary simply leaves the system. The vocabulary we use here is tailored towards the airline industry, but generalizations to other industries are obvious.

The problem takes place over the time periods $\{1,\ldots,\tau\}$. All flights depart at time period $\tau + 1$. We have m flights and n itineraries. The initial capacity on flight leg i is c_i. If we accept a request for itinerary j, then we generate a revenue of f_j and consume a_{ij} units of capacity on flight leg i. If flight leg i is not in itinerary j, then we have $a_{ij} = 0$. We let A be the capacity consumption matrix $[a_{ij}]$ and use A^j to note the j-th column of A.

At most one itinerary request arrives at each time period. This assumption is reasonable because we can divide the time interval between now and the time of the departure into very small time slices so that the probability of having two or more itinerary requests in a time slice is negligible. In fact, we assume that *exactly* one itinerary request arrives at each time period. It is easy to make generalizations to the case where *at most* one itinerary request arrives at each time period. The probability that we have a request for itinerary j at time period t is p_{jt}. Since exactly one itinerary request arrives at each time period, we have $\sum_{j=1}^{n} p_{jt} = 1$.

We let x_{it} be the remaining capacity on flight leg i at time period t. In this case, the vector $x_t = \{x_{it} : i = 1,\ldots,m\}$ gives the state of the system at time period t. We use $y_t = \{y_{jt} : j = 1,\ldots,n\}$ to denote the decisions at time period t, where y_{jt} takes value 1 if we accept a request for itinerary j at time period t, otherwise it takes value 0.

Given that the capacities on the flight legs at time period t are x_t and we observe a request for itinerary j, the expected revenue obtained over the time periods $t,\ldots,\tau$ satisfies the optimality equation

$$V_t(x_t, j) = \max \quad f_j\, y_{jt} + \sum_{j'=1}^{n} p_{j',t+1} \, V_{t+1}(x_t - y_{jt} A^j, j')$$

$$a_{ij}\, y_{jt} \le x_{it} \qquad i = 1, \dots, m$$

$$y_{jt} \in \{0,1\}$$

with the boundary condition that $V_{\tau+1}(\cdot,\cdot) = 0$. The summation on the right side above only depends on $x_t - A_j\, y_{jt}$ and letting

$$\bar{V}_t(x_t) = \sum_{j=1}^{n} p_{jt} \; V_t(x_t, j) \qquad \qquad = \bar{V}_{t+1}(x_t - y_t A^j)$$

the optimality equation becomes

$$\begin{aligned} V_t(x_t, j) = \max \quad & f_j\, y_{jt} + \bar{V}_{t+1}(x_t - y_{jt}\, A^j) \\ \text{subject to} \quad & a_{ij}\, y_{jt} \le x_{it} \quad i = 1, \dots, m \\ & y_{jt} \in \{0, 1\}. \end{aligned}$$

Taking expectations of both sides, the optimality equation above can be written as

$$\bar{V}_t(x_t) = \sum_{j=1}^{n} p_{jt} \; V_t(x_t, j) = \sum_{j=1}^{n} p_{jt} \left\{ \begin{aligned} & \max \quad f_j\, y_{jt} + \bar{V}_{t+1}(x_t - y_{jt} A^j) \\ & a_{ij}\, y_{jt} \le x_{it} \qquad i = 1, \dots, m \\ & y_{jt} \in \{0,1\} \end{aligned} \right\}$$

We write the final optimality equation as

$$\max \quad \sum_{j=1}^{n} p_{jt} \left\{ f_j\, y_{jt} + \bar{V}_{t+1}(x_t - y_{jt} A^j) \right\}$$

$$\text{st}$$

$$a_{ij}\, y_{jt} \le x_{it} \qquad i = 1, \dots, m \quad j = 1, \dots, n$$

$$y_{jt} \in \{0,1\} \qquad j = 1, \dots, n$$

We note that this dynamic program only involves the state variable x_t. The state variable in this dynamic program has m dimensions. If the capacity on each flight is on the order of C, then the number of possible values of the state variable is on the order of C^m and it is very difficult to solve this dynamic program by the traditional backward recursion approach that we use for single-leg revenue management problems.

One important observation from the dynamic programming formulation of the network revenue management problem is that if the capacities on the flight legs at time period t are x_t and we observe a demand for itinerary j, then

we accept a request for itinerary j if

$$f_j + \bar{V}_{t+1}(x_t - A^j) \geq \bar{V}_{t+1}(x_t) \quad \text{and} \quad a_{ij} \leq x_{it} \quad i=1,\dots,m$$

otherwise we reject a request for itinerary j at time period t

$$f_j \geq \bar{V}_{t+1}(x_t) - \bar{V}_{t+1}(x_t - A^j)$$

Consequently, if we can approximate $\bar{V}_{t+1}(x_t) - \bar{V}_{t+1}(x_t - A^j)$ in a tractable manner, then we can find good policies for the network revenue management problem.

2. Linear programming formulation of the network revenue management problem
Given that the capacities on the flight legs at time period t are x_t, the value function $\bar{V}_t(x_t)$ *exactly* gives the expected revenue obtained over the time periods $t, \dots, \tau$. On the other hand, the linear program

$$\sum_{j=1}^{n} f_j w_j$$

$$\text{st} \quad \sum_{i=1}^{n} a_{ij} w_j \leq x_{it} \quad i=1,\dots,m$$

$$0 \leq w_j \leq \sum_{t'=t}^{\tau} p_{jt'}$$

can be used to approximate the expected revenue over the time periods $t, \dots, \tau$, where the decision variable w_j is the number of requests for itinerary j that we plan to accept to over the remaining portion of the planning horizon. In this case, letting $L_t(x_t)$ be the optimal objective value of the linear program above, we can use the decision rule

accept a request for itinerary j at time period t if

$$f_j \geq L_t(x_t) - L_t(x_t - A^j) \quad \text{and} \quad a_{ij} \leq x_{it} \quad i=1,\ldots,m.$$

to make the decisions at time period t.

Note that we compare f_j with $L_t(x_t) - L_t(x_t - A^j)$ to decide whether to accept or reject a request for itinerary j at time period t. More "correct" approach would be to compare f_j with $L_{t+1}(x_t) - L_{t+1}(x_t - A^j)$, but this difference is minor.

3. Computing bid-prices

It is easy to realize that the dynamics of the airline network revenue problem do not work the way we described above. In particular, the airline does not observe an itinerary request and decide whether to accept or reject the itinerary request. Instead, the airline decides which itineraries should stay open and which itineraries should be closed, and if a request for an open itinerary arrives, then the airline has to accept the itinerary request. (We implicitly assume that if there is not enough capacity for an itinerary, then the itinerary is closed by default.)

Therefore, if an airline is to use the approach described above to decide which itineraries should stay open and which itineraries should be closed, then it has to compute $L_t(x_t) - L_t(x_t - A^j)$ for all $j = 1, \ldots, n$. The difficulty here is that

n can be large.

As an alternative approach, we note that $\bar{V}_{t+1}(x_t) - \bar{V}_{t+1}(x_t - A^j)$ describes how much the expected revenue over the time periods $t+1, \ldots, \tau$ changes if we give away all the seats consumed by itinerary j. On the other hand, if we let $\{\mu^*_{it}(x_t) : i = 1, \ldots, m\}$ be the dual variables associated with the first set of constraints in the linear program, then $\mu^*_{it}(x_t)$ may be used to approximate how much the expected revenue over the time periods $t, \ldots, \tau$ changes if we give away one seat on flight leg i. Consequently, if we give away the seats on all the flight legs consumed by itinerary j, then the change in the expected revenue can be approximated by

$$\bar{V}_{t+1}(x_t) - \bar{V}_{t+1}(x_t - A^j) \approx \sum_{i=1}^{m} a_{ij} \, \mu_i^*(x_t)$$

In this case, we can use the decision rule

Accept a request for itinerary j at time period t if

$$f_j \geq \sum a_{ij} \mu_i^*(x_t) \quad \text{and} \quad a_{ij} \leq x_{it} \quad i = 1, \ldots, m.$$

to decide which itineraries should stay open and which itineraries should be closed.

For the rest of this section, we continue with the "false" dynamics of the airline network revenue management problem. In particular, we assume that the airline first observes an itinerary request, and then, decides whether to accept or reject the itinerary request.

4. Connections between the dynamic programming formulation and the linear programming approximation

Consider dropping the capacity availability constraints at time periods $1, \ldots, \tau - 1$ in the dynamic programming formulation of the network revenue management problem, which is given by

max cx
st
$Ax \leq b$
$Wx \leq u$

$cx + \lambda(u - Wx)$.

$$\begin{aligned}
\bar{V}_t(x_t) = \max \quad & \sum_{j=1}^{n} p_{jt} \left[f_j \, y_{jt} + \bar{V}_{t+1}(x_t - y_{jt} \, A^j) \right] \\
\text{subject to} \quad & a_{ij} \, y_{jt} \leq x_{it} \quad i = 1, \ldots, m, \; j = 1, \ldots, n \\
& y_{jt} \in \{0, 1\} \quad j = 1, \ldots, n.
\end{aligned}$$

Furthermore, consider associating the positive Lagrange multipliers $\{p_{j\tau} \lambda_i : i = 1, \ldots, m, \; j = 1, \ldots, n\}$ with the capacity availability constraints at the last time period τ to add these constraints to the objective function. In particular, we solve the optimality equation

$$\begin{aligned}
\bar{V}_t^\lambda(x_t) = \max \quad & \sum_{j=1}^{n} p_{jt} \left[f_j \, y_{jt} + \bar{V}_{t+1}^\lambda(x_t - y_{jt} A^j) \right] \\
\text{st} \quad & y_{jt} \in \{0, 1\}
\end{aligned}$$

for time periods $1, \ldots, \tau - 1$, whereas we solve the optimality equation

$$\begin{aligned}
\bar{V}_t^\lambda(x_t) = \max \quad & \sum_{j=1}^{n} p_{j\tau} \, f_{j\tau} \, y_{j\tau} + \sum_{i=1}^{m} \sum_{j=1}^{n} p_{j\tau} \, \lambda_i \, (x_{i\tau} - a_{ij} \, y_{j\tau}) \\
\text{st} \quad & y_{j\tau} \in \{0, 1\}
\end{aligned}$$

for the last time period. The Lagrange multipliers work as a penalty term. If the relaxed constraint is violated, then they make the objective function smaller. We use the superscript

λ in the value functions to emphasize that the solution to the optimality equation above depends on the Lagrange multipliers. We scale the Lagrange multipliers by $\{p_{j\tau} : j = 1, \ldots, n\}$ because this makes the arithmetic cleaner. This dynamic programming formulation is by no means equivalent to the original formulation. However, it can be a good substitute as long as it can be solved efficiently.

We begin by showing that the Lagrangian relaxation strategy described above gives an upper bound on the value functions. That is, we have $\overline{V_t}(x_t) \le \overline{V_t}^{\lambda}(x_t)$ for all time periods and for all capacity levels, as long as the Lagrange multipliers are positive. This result is easy to show by induction. To see that the result holds for the last time period τ, we have

$$\overline{V_\tau}(x_\tau) = \max_{\substack{\text{st} \\ a_{ij}\, y_{j\tau} \le x_{i\tau} \quad i=1,\ldots,m,\ j=1,\ldots,n \\ y_{j\tau} \in \{0,1\} \quad j=1,\ldots,n}} \sum_j p_{j\tau}\, f_{j\tau}\, y_{j\tau} = \sum_j p_{j\tau}\, f_{j\tau}\, y_{j\tau}^*$$

$$\le \sum_j p_{j\tau}\, f_{j\tau}\, y_{j\tau}^* + \sum_j \sum_i p_{j\tau}\, \lambda_i\, (x_{i\tau} - a_{ij}\, y_{j\tau}^*)$$

$$\le \max_{\substack{\text{st} \\ y_{j\tau} \in \{0,1\} \quad j=1,\ldots,n}} \sum p_{j\tau}\, f_{j\tau}\, y_{j\tau} + \sum_j \sum_i p_{j\tau}\, \lambda_i\, (x_{i\tau} - a_{ij}\, y_{j\tau}^*) = V_\tau^{\lambda}(x_\tau)$$

Similarly, assuming that the result holds for time period $t+1$ (that is, $\overline{V_{t+1}}(x_t) \le \overline{V_{t+1}^{\lambda}}(x_t)$ for all capacity levels), we have

$$\overline{V_t}(x_t) = \max_{\substack{\text{st} \\ a_{ij}\, y_{jt} \le x_i \quad i=1,\ldots,m \quad j=1,\ldots,n \\ y_{jt} \in \{0,1\}}} \sum p_{jt} \left[f_j \cdot y_{jt} + \overline{V_{t+1}}(x_t - y_{jt} A^j) \right]$$

$$= \sum_j p_{jt} \left[f_j\, y_{jt}^* + \overline{V_{t+1}}(x_t - y_{jt}^* A^j) \right]$$

$$\le \sum_j p_{jt} \left[f_j\, y_{jt}^* + \overline{V_{t+1}}^{\lambda}(x_t - y_{jt}^* A^j) \right]$$

$$\le \max_{y_{jt} \in \{0,1\} \quad j=1,\ldots,n} \sum_j p_{jt} \left[f_j\, y_{jt} + \overline{V_{t+1}}^{\lambda}(x_t - y_{jt} A^j) \right]$$

This shows that $\bar{V}_t(x_t) \leq \bar{V}_t^\lambda(x_t)$.

Another property of the Lagrangian relaxation strategy is that for a fixed value of Lagrange multipliers, the value function $\{V_t^\lambda(\cdot) : t = 1, \ldots, \tau\}$ can easily be computed. In particular, we have

$$V_t^\lambda(x_t) = \sum_{i=1}^m \lambda_i \, x_{it} + \sum_{j=1}^n \sum_{t'=t}^\tau p_{jt'} \left[f_j - \sum_{i=1}^m a_{ij} \, \lambda_i \right]^+ .$$

This result is also easy to show by induction. To see that the result holds for the last time period τ, we have

$$\bar{V}_\tau^\lambda(x_\tau) = \max_{y_{j\tau} \in \{0,1\},\ j=1,\dots,n} \sum_{j=1}^n p_{j\tau} \left(f_j - \sum_{i=1}^m a_{ij} \lambda_i \right) y_{j\tau} + \sum_i \Big(\sum_j p_{j\tau} \Big) \lambda_i x_{i\tau}$$

$$= \sum_{j=1}^n p_{j\tau} \left(f_j - \sum_{i=1}^m a_{ij} \lambda_i \right)^+ + \sum_i \lambda_i x_{i\tau}$$

Similarly, assuming that the result holds for time period $t+1$, we have

$$\bar{V}_t^\lambda(x_t) = \max \sum_{j=1}^n p_{jt} \left(f_j y_{jt} + \bar{V}_{t+1}^\lambda (x_t - y_{jt} A^j) \right)$$

$$= \max \sum_{j=1}^n p_{jt} \left(f_j y_{jt} + \sum_{i=1}^m \lambda_i (x_{it} - y_{jt} a_{ij}) + K_{t+1} \right)$$

$$= \max \sum_{j=1}^n p_{jt} \left(f_j - \sum_{i=1}^m \lambda_i a_{ij} \right) y_{jt} + \sum_i \sum_j p_{jt} \lambda_i x_{it} + K_{t+1}$$

$$= \sum_{j=1}^n p_{jt} \left(f_j - \sum_{i=1}^m \lambda_i a_{ij} \right)^+ + \sum \lambda_i x_{it} + K_{t+1}$$

This completes the proof.

The most obvious question is how to choose the Lagrange multipliers. Since the initial capacities on the flight legs are given by $c = \{c_i : i = 1, \ldots, m\}$, the maximum total expected revenue over the time periods $\{1, \ldots, \tau\}$ is given by $V_1(c)$. On the other hand, we know that the value function that we obtain under the Lagrangian relaxation strategy provides an upper bound on the exact value function. In particular, we have $\overline{V_1}(c) \leq \overline{V_1^\lambda}(c)$ as long as the Lagrange multipliers are positive. Therefore, we can solve the problem

$$\min_{\lambda \geq 0} \overline{V_1^\lambda}(c)$$

to find a good set of Lagrange multipliers. As complicated as it looks, the minimization problem above can be solved as a linear program.

To see this, we first note that for a fixed value of λ, we have

$$V_1^\lambda(c) = \sum_{i=1}^{m} \lambda_i \, c_i + \sum_{j=1}^{n} \sum_{t=1}^{\tau} p_{jt} \left[f_j - \sum_{i=1}^{m} a_{ij} \, \lambda_i \right]^+ .$$

The first thing to note is that for a fixed value of λ, $V_1^\lambda(c)$ is the optimal solution to the trivial linear program

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} \lambda_i c_i + \sum_{j=1}^{n} \left(\sum_{t=1}^{\tau} p_{jt} \right) \beta_j \\ & \beta_j \geq f_j - \sum_{i=1}^{m} a_{ij} \lambda_i \qquad j = 1, \ldots, n \\ & \beta_j \geq 0 \qquad j = 1, \ldots, n \end{aligned}$$

If we want to solve the problem $\min_{\lambda \geq 0} V_1^\lambda(c)$, then all we need to do is to treat the Lagrange multipliers in the linear program above as decision variables and solve the problem

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} \lambda_i c_i + \sum_{j=1}^{n} \left(\sum_{t=1}^{\tau} p_{jt} \right) \beta_j \\ \text{st} \quad & \beta_j + \sum_{i=1}^{m} a_{ij} \lambda_i \geq f_j \qquad j = 1, \ldots, n \qquad (w_j) \\ & \beta_j \geq 0 \qquad j = 1, \ldots, n \\ & \lambda_i \geq 0 \qquad i = 1, \ldots, m \end{aligned}$$

Therefore, we know how to pick a good set of values for the Lagrange multipliers. Once we have these Lagrange multipliers, we can compute a good approximation to the value function by

$$V_t^\lambda(x_t) = \sum_{i=1}^m \lambda_i \, x_{it} + \sum_{j=1}^n \sum_{t'=t}^\tau p_{jt'} \left[f_j - \sum_{i=1}^m a_{ij} \, \lambda_i \right]^+ .$$

A side reason that we do all these computations is to show the link between the dynamic programming formulation of the network revenue management problem and the linear programing approximation. We consider the dual of the last linear program that we have above. Note that this linear program computes $\min_{\lambda \geq 0} V_1^\lambda(c)$. The dual of this linear program is

$$\max \quad \sum_{j=1}^n f_j \, w_j$$

st

$$\sum_{j=1}^n a_{ij} \, w_j \leq c_i \qquad (\lambda_i) \qquad i=1,\ldots,m$$

$$w_j \leq \sum_{t=1}^T p_{jt} \qquad j=1,\ldots,n$$

$$w_j \geq 0 \qquad j=1,\ldots,n$$

Therefore, the Lagrangian relaxation strategy is equivalent to the linear programming approximation.

This observation may not sound too interesting. However, if we face a network revenue management problem for which we do not have a linear programming approximation, then we can first write the dynamic programming formulation of the network revenue management problem. After this, we can relax the capacity availability constraints to see what the linear programming approximation should look like. For example, we do not have a linear programming approximation for the network revenue management problem with cancellations. However, we know how to formulate the network revenue management problem with cancellations as a dynamic program. In this case, we can use the Lagrangian relaxation strategy to see what the linear programming approximation should look like in the presence of cancellations.