Solving large structured Markov Decision Problems for perishable inventory management and traffic control

Haijema, R.

Publication date
2008

Appendix C

Simulation-based search for order-up-to levels

Finding an optimal order-up-to level S^* to an ordinary order-up-to S rule for a stationary problem with period 1 is easy when the cost function appears to be convex. Commonly used procedure are binary search and Fibonacci search. When the cost function is not convex but may have multiple local optima, then an optimal order-up-to level is obtained by complete enumeration over a range of order-up-to levels. If no explicit expression for the cost function in terms of S exist, simulation is used to evaluate a chosen value of S.

The optimization problem that we are studying is periodic with period 7 and an optimal order-up-to S rule consist of five parameters: $(S_1, S_2, \ldots, S_5)$; one for each of the five working days. Production volumes are set according to $\min(S_d - x, U_d)$, with x the total stock level and U_d is the production capacity, which is most studies assumed to be (virtually) infinite. Finding optimal values for $(S_1, S_2, \ldots, S_5)$ is much more difficult, since depending on the costs structure there are many local optima.

Finding nearly optimal parameter values for the 2D-rule seems to be even more complicated since it consists of ten parameters for problems with five working days a week. Despite these difficulties we present two fast incremental search algorithms based on simulation. We do not guarantee these algorithms to find the best optimal order-up-to levels, but they have shown to be successful to a number of cases that we have considered. In fact we have also observed cases where the incremental search, should be inverted, such that one starts at high order-up-to values and successively decrease the levels. Since these search algorithm are not the focus of this thesis, we did not studied the conditions under which the algorithm does execute well.

For the description of the algorithm we restrict ourselves to the PPP with production on Monday to Friday.

C.1 Ordinary order-up-to S rule

For the order-up-to S rule with one order-up-to level per working day, we present the 1D-local-search algorithm. First we introduce some notation.

(Notation) Let $\mathbf{e}_d$ denote the 5-dimensional vector with all components equal to 0 except for element d which is 1, e.g., $\mathbf{e}_2 = (0, 1, 0, 0, 0)$. Vector $\mathbf{1}$ is the 5-dimensional vector with all elements equal to 1: $\mathbf{1} = (1, 1, 1, 1, 1)$.

1D-local-search algorithm

- *Initialize.* Define $\mathbf{S} = (0, 0, 0, 0, 0)$
- *Step 1.* Replace $\mathbf{S}$ by the best improvement $\mathbf{S} + \mathbf{1}$ until no further improvement is found,
- *Step 2.* Replace $\mathbf{S}$ by the best improvement $\mathbf{S} + \mathbf{e}_d$ until no further improvement is found over the last 5 iterations,
- *Step 3.* Replace $\mathbf{S}$ by the best improvement $\mathbf{S} - \mathbf{e}_d$ until no further improvement is found over the last 5 iterations.

Starting in with an initial choice of $\mathbf{S}$ we apply an incremental search; to test for improvements the order-up-to S rule is evaluated through a few, say 10, short simulation runs. The production volumes follow in principle from the order-up-to levels, but production will not exceed the production capacity on some day. The search algorithm thus anticipates that in practice the production capacity may be restrictive.

In the 1D-local-search algorithm the result of Step 1 is a set of order-up-to levels identical for each of the five working days. Applying the order-up-to S rule with these order-up-to levels implies that the production volume on some day equals the demand on the previous day. The production level on Monday is the demand over the last three days. Due to the perishability and the periodicity some of the order-up-to levels should be higher and maybe some of them should be set lower. This is done through steps 2 and 3.

In each iteration of Step 2 the production on some day d is raised by 1, consequently the production on day $d + 1$ is lowered by 1. (When the production on Friday is raised by 1, the available stock on Monday may be 1 higher and thus the production volume on Monday is 1 lower.) In Step 3 the production on day d is lowered by 1, consequently the production on day $d + 1$ is raised by 1.

Initial S – Instead of taking the null-vector as an initial choice of **S** one could set **S** to the mean demand over the next days until the next production is released. S_1 is then the mean demand over Monday and Tuesday, S_5 is the sum of the mean demands over Friday, Saturday, Sunday and Monday. When the initial value of **S** is still very low, e.g. because Step 1 is skipped, we should be careful in setting the order in which increments are evaluated in Step 2 and 3. One should enforce fair tie-breaking when increments on different days are equally well.

Simulation – Each increment is evaluated through 10 simulation runs. When the current value of **S** is still far from optimal, the simulation runs can be very short to evaluate whether an increment of **S** is an improvement. When an increment does not result in an improvement over all 10 runs then the length of simulation run is increased to obtain a more accurate evaluation. The initial run length that we have chosen is 250 weeks, which is successively increased to 1,000, and 4,000 weeks. When 10 runs of 4,000 weeks do not result in an improvement Step 2 of the search algorithm is executed. Before an increment in Step 2 is selected one has to evaluate all 5 possible increments. Step 2 is terminated and Step 3 is started when none of the 5 increments result in an improvement.

Running time – The running time of the search procedure depends on the scale of the problem. For the down-scaled problem the search last about one minute. For the full size problem the search may take 15 minutes to half an hour, depending on the length of the simulation runs. When production and demand happens on a very large scale, the simulation-based search will take much longer than solving the MDP formulation after scaling the problem.

C.2 2D-order-up-to S rule

For the 2D-order-up-to S rule one has to optimize over ten different parameters $\left(\mathbf{S}^{\mathrm{T}}, \mathbf{S}^{\mathrm{Y}}\right)$: $\left(S_1^{\mathrm{T}}, S_2^{\mathrm{T}}, S_3^{\mathrm{T}}, S_4^{\mathrm{T}}, S_5^{\mathrm{T}}; S_1^{\mathrm{Y}}, S_2^{\mathrm{Y}}, S_3^{\mathrm{Y}}, S_4^{\mathrm{Y}}, S_5^{\mathrm{Y}}\right)$. Parameter S_1^{T} denotes the order-up-to level for the total stock, S_1^{Y} is the order-up-to level related to the 'young' products in stock. The optimization over ten parameters seems to be a complicated task, but the 1D-local-search algorithm of the previous section is helpful to find initial values. Again an incremental search, with increments and decrements of the order-up-to levels commonly results in good parameter values. Again, we do not guarantee the algorithm to converge to the best parameter set, since under different cost structures, it may end-up in a local optimum.

2D-local-search algorithm

- *Initialize.* First execute the 1D-local-search algorithm. Let $\mathbf{S}^*$ be the best policy found. Set $\mathbf{S}^{\mathrm{T}} := \mathbf{S}^*$ and $\mathbf{S}^{\mathrm{Y}} := \mathbf{S}^*$.
- *Step 1.* This overestimates the levels for 'young', thus replace $\mathbf{S}^{\mathrm{Y}}$ by the best improvement $\mathbf{S}^{\mathrm{Y}} - \mathbf{e}_d$ until no further improvement is found.
- *Step 2.* Replace $\mathbf{S}^{\mathrm{T}}$ by the best $\mathbf{S}^{\mathrm{T}} - \mathbf{e}_d$ for $d = 1, \ldots, 5$ until no further improvement is found.
- *Step 3.* Replace $\mathbf{S}^{\mathrm{Y}}$ by the best $\mathbf{S}^{\mathrm{Y}} + \mathbf{e}_d$ until no further improvement is found.

Starting with $\mathbf{S}^{\mathrm{T}}$ and $\mathbf{S}^{\mathrm{Y}}$ equal to the order-up-to levels for the ordinary order-up-to S rule, one successively decreases in Step 1 the order-up-to levels for young. Since under the 2D-rule the production volume in some states is determined by $S_d^{\mathrm{Y}} - x^{\mathrm{Y}}$, one may decrease S_d^{T}, as in Step 3. Further fine-tuning $\mathbf{S}^{\mathrm{Y}}$ may result in an increase of $\mathbf{S}^{\mathrm{Y}}$, since $\mathbf{S}^{\mathrm{T}}$ may be lowered in the previous step. Fine-tuning the 10 parameters is time-consuming since it requires relatively long simulation runs for an accurate evaluation.