Scenario Tree Based Heuristics for Stochastic Inventory Routing Problems

Lars Magnus Hvattum, Molde University College, Norway
Arne Løkketangen, Molde University College, Norway
Gilbert Laporte, HEC Montreal, Canada

In the business practice called vendor managed inventory replenishment (VMI) the vendor decides when to make deliveries to its customers, how much to deliver, and how to combine the shipments using the available equipment, e.g., trucks. This gives rise to the *Inventory Routing Problem* (IRP), in which the goal is to coordinate inventory replenishment and transportation in order to minimize costs. That is, the vendor decides, on every day, which customers to visit, how much to deliver to each customer, and how to arrange the customers on vehicle routes. Often, however, the exact demand of each customer is not known with certainty, and it can be appropriate to consider the *Stochastic Inventory Routing Problem* (SIRP), in which the demand of each customer is assumed to be a random variable with a specific distribution.

Introduction

In this presentation the aim is to handle the variations of the *Stochastic Inventory Routing Problem* as described in Kleywegt, Nori and Savelsberg (2002,2004), and Adelman (2004). In each of these papers, a SIRP is formulated as a Markov Decision Process (MDP) over a discounted, infinite horizon. The MDP governs one vendor, a single product (assuming that different products cannot be transported together on any vehicle), a set of customers, and a set of homogeneous vehicles.

The MDP has decision epochs that correspond to days, with the assumption that the state of the system can be observed at the beginning of an epoch, that actions can be taken, and that the demands of the customers are observed after any action has been made. As for the state of the system, this is described through the inventory states of all the customers. The objective is to find a policy for the MDP that maximizes the expected total discounted value over an infinite horizon, where the value is based on rewards and costs associated with the process. These rewards/costs are calculated based on four components, which include the travel costs, the holding costs at the customer locations, penalties for stock-outs at the customers (back-logging is not considered), and possibly a revenue gathered for each delivery made. Note that in Adelman (2004) the holding costs are calculated after observing the demand, while in both of Kleywegt, Nori and Savelsberg (2002, 2004) it remains unclear how the holding costs are treated.

It should be noted that the inventory of each customer is discretized, and that the demand of each customer is assumed to be described by a general discrete distribution.

The main difference between the problems handled in Kleywegt, Nori and Savelsberg (2002, 2004) and Adelman (2004) lies in the actions that are allowed in each state, where the actions are understood to consist of a set of vehicle itineraries including delivery quanta for each customer visit. In Kleywegt, Nori and Savelsberg (2002) only direct deliveries were allowed, meaning that each vehicle route can only accommodate at most one customer visit. This constraint was slightly relaxed in Kleywegt, Nori and Savelsberg (2004) where the vehicle routes were allowed to include up to three customer visits. Finally, in Adelman (2004) there were no constraints on the number of customer visits per vehicle route (except implicitly through the vehicle capacity), but here one instead assumed that there was no limit on the number of vehicles available.

In order to cover all the possibilities from Kleywegt, Nori and Savelsberg (2002, 2004) and Adelman (2004), one has to allow restrictions on the number of visits per vehicle, restrictions on the vehicle capacity, and restrictions on the number of vehicles, although each of these could also possibly be unrestricted. The storage capacity of each customer is always considered to be limited. In each of the three aforementioned papers, split deliveries were not allowed, meaning that on any day a customer may be visited at most once.

The Scenario Tree Problem

In this section, a general framework for handling the *Stochastic Inventory Routing Problem* is proposed. While previous work considering Markov Decision Process formulations of the SIRP has mainly focused on making approximations of the exact value function (see Kleywegt, Nori and Savelsberg, 2002, 2004, and Adelman, 2004), the framework developed here is based on using scenario trees. The idea is that although the underlying MDP is formulated over an infinite horizon, it is possibly sufficient to examine a finite horizon in order to make good decisions with respect to the inventory and the routing. Hence we create a scenario tree intended to capture the stochastics of the SIRP, formulate a *Scenario Tree Problem* (STP) for a given state (i.e., a given set of inventory levels), and conjectures that the solution of this STP will correspond to decent decisions with respect to the underlying MDP.

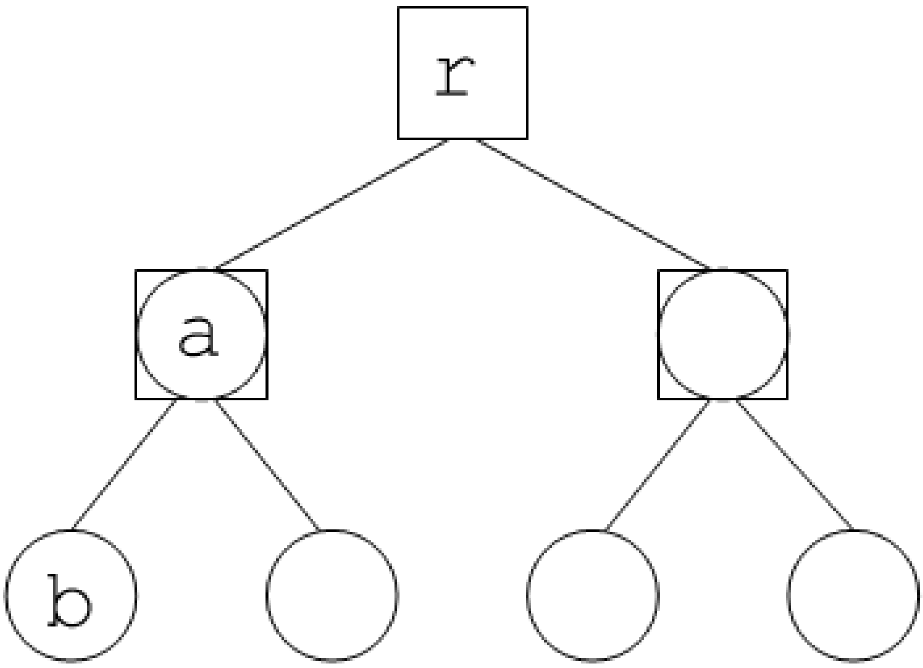

Figure 1. A sketch of a small scenario tree

For each period one has an observed state (inventory levels), then performs a set of actions (deliveries to customers), after which the actual demand is observed. In compliance with Adelman (2004) the holding costs and the penalty for lost demand is calculated after observing the actual demand, whereas the reward from delivery and the transportation costs are calculated immediately as a consequence of the set of actions performed. With respect to the scenario tree, as given in Figure 1, this means that all nodes except the leaves includes active decisions, and that all nodes except the root includes an observed demand (whereas the root node only includes the starting state). In Figure 1 the squares represent decisions and the circles outcomes. The letter *r* represents the root node, *a* is similarly an internal node, while *b* is a leaf node.

A formalized model of the problem, using mathematical notation, henceforth referred to as the *Scenario Tree Problem* (STP), will be presented at the conference, while an informal description is given in the following. The objective consists of four parts which represents

a) the reward for delivery
b) the holding costs
c) the transportation costs
d) the penalties for lost demand

Note that a) and c) can be calculated directly from the decisions made in each node, whereas b) and d) can only be calculated after observing the actual demand (as the cost added here really belongs to the parent node, the discount factor is also that of the parent node).

The constraints ensure that the inventory level of each customer never exceeds the maximum limit, and that the transfer of inventory from one period to the next is handled while taking into account the demand and deliveries. The initial inventory levels also needs to be set.

They also ensure that the vehicle capacity is not violated and that a vehicle must visit a customer in order to deliver any goods.

The movement of the vehicles is controlled through the remaining constraints. Each vehicle must respectively leave and arrive at the depot (possibly without visiting any customers, in which case the vehicle may be considered unused). Any vehicle that arrive at a customer also leaves and vice versa, and subtours must be eliminated. As customers are not required to be visited, equations, we need to ensure that no customer is visited too many times in a period (usually limited to one). The number of customers on a tour is limited, but not the total travel length.

GRASP for the Scenario Tree Problem

Solving the original Markov Decision Problems, which involves finding the optimal policy for each possible state, is unpractical for all but the smallest instances. Since the exact evaluation of a policy is equally problematic (requiring the specification of the policy for each state, the number of states being very large), one is reduced to evaluating the different heuristic methods through simulations. The idea here is to use a *Scenario Tree Problem* for the SIRP to find actions for each state encountered during the simulation.

The main heuristic idea investigated in this paper is based on GRASP (Greedy Randomized Adaptive Search Procedure, see, e.g., Resende and Ribeiro, 2000). In GRASP, each iteration involves the construction of a solution from scratch, making greedy choices that are somewhat randomized in order to get different results from each iteration. The evaluations of the possible choices that can be made during the construction are updated as the new solution evolves, and hence it is said to be adaptive. GRASP has already been used for a variety of combinatorial problems, including the *Inventory Routing Problem* in Campbell and Savelsbergh (2004). However, that paper did not include the use of scenario trees, and the use of GRASP was only a very minor part of the work.

The Decisions during a GRASP Construction

The typical decision that is made in each step of the construction in standard GRASP, is to assign a value permanently to a variable. In the case of the *Scenario Tree Problem*, this type of decision might seem too inflexible since several groups of variables (i.e., routing/inventory/delivery) are involved simultaneously. Therefore, an alternative is found where the explicit decisions are concerned with the delivery quanta only, and where the remaining decisions are treated implicitly. In essence, each step of the construction consists of making a decision of the form:

Deliver y more units to customer i, using vehicle k in node v.

In other words, the routing decisions are assumed to be deduced from the delivery quanta, e.g., by solving a *Traveling Salesman Problem* (TSP) for each vehicle in each node, including all customers having a delivery by that vehicle. This type of decision allows more flexibility than if routing decisions were to be included explicitly.

The Evaluation of the Decisions

Having established the type of decision to make at each step of the construction, a word on the evaluation of those decisions is warranted. The starting point of each construction is a solution with zero delivery, which can be evaluated by calculating the negative effects of any penalties that are due to stock-outs in addition to any holding costs based on the initial inventory. Considering any decision of the type described above, one can calculate the exact impact on the objective function value of the current solution. The increase in travel cost can be readily found by solving the current and the resulting TSP for the vehicle, while the increase in rewards gathered follows immediately. Calculating the change in holding costs and penalties is slightly more involved, as a recursive inspection may possibly need to involve the entire subtree which is rooted in the current node. The feasibility check of the increased delivery can be performed simultaneously with the evaluation.

The Restricted Candidate List

After evaluating each of the possible increases in delivery quantum, one has to choose one of the increases to be added in the current solution. In GRASP this is done by randomly selecting one of the decisions from a Restricted Candidate List (RCL), which consists of a subset of all the decisions based on either their rank or their evaluation. That is, there is a parameter γ such that the decision at any step of the construction is either randomly selected from the γ% best decisions (rank-based), or from the set of decisions that are within γ% of the best decision (value-based). In this case, though, a decision is not allowed if it is either infeasible, or if its evaluation is less than zero (which would indicate that the overall objective function value would decrease if the decision was to be implemented at the current step). Note that the requirement of the evaluations being non-negative, together with the fact that only feasible insertions are made, leads to an obvious stopping criterion for each construction.

One common variation to either of the two schemes for selecting the Restricted Candidate List, is to allow for the parameter γ to be updated dynamically. This is commonly referred to as reactive GRASP (see, e.g., Prais and Ribeiro, 2000), and is usually handled by first selecting a set of possible values for γ. A value is then selected at random from this set at each iteration of the GRASP. While the probabilities for selecting any particular value for γ are initially equal, they are updated throughout the search so as to favor values of γ that lead to good solutions.

Reducing the Number of Possible Decisions

Taking into account the large number of possible insertions, especially at the early stages of the construction, one is required to consider ways of reducing the number of candidate decisions. This is due to the fact that reducing the number of candidate decisions may both help speed up the construction process, as well as preventing certain types of low-quality or counter-intuitive decisions from being made.

One possible way of reducing the number of decisions at any step of the construction is to utilize the scenario tree structure of the problem. Rather than considering insertions in any node, it is possible to limit the insertions to be made in a top-down fashion, where all decisions regarding the root node is made first, before continuing the construction node by node, recursively in the tree.

The number of possible decisions is also reduced by limiting the available choices of the vehicle to use for any combination of quantity, customer and node. This is accomplished by only considering, for given quantity, customer and node, the one vehicle that results in the highest evaluation.

It is also possible to consider limitations on the size of the increase of delivery. Henceforth, four different alternatives have been identified. While it is always an option to evaluate all feasible quantities of increased delivery, one could also let the increase be limited to the one with highest evaluation, for a given customer and node. The two other alternatives considered both consists of always inserting one unit of additional delivery at each step, but possibly to evaluate this insertion based on the best amount to deliver rather than only one unit. The reasoning behind the evaluation being based on the best amount rather than 1, even if the latter is the inserted quantum, is that it better represents the eventual gain by starting to increase the delivery to the given customer in the given node.

The Principle of Marginal Conditional Validity

The principle of *Marginal Conditional Validity* (MCV) was introduced by Glover (2000). It concerns the fact that in a constructive framework, early decisions are myopic (and thus more likely bad), while later decisions will tend to make the earlier decisions seem better.

A procedure based on these ideas is proposed in the GRASP for the STP. Its goal is to find features of the current solution that looks like they should have been different, and then to encourage (or force) changes in future constructions. The hope is that this will help create better, and more diverse, solutions. To this extent, three different types of features are identified by the MCV-procedure. They can be summarized as follows.

- There is no delivery made to a customer that will have a stock-out (in at least one of the possible realizations of its demand), and there is free capacity on the vehicles.
- There is delivery made to a customer that will not have a stock-out, and there is no free capacity on the vehicles.
- There is delivery made to a customer for which the delivery is not immediately profitable (i.e., it is only profitable since it saves penalty that occurs later than the current stage) and the customer has a non-zero inventory.

While the first of these indicates that one should try to make a delivery to the customer, the other two indicate that one should test whether it is better not to make a delivery. In the second case, it is perhaps better to service some other customer, but in the third case it might actually be better simply to delay service till a later period (this could happen if the holding costs are relatively high compared to the expected stock-out penalty). Note that for all three cases, it might be that the current decision is actually the best choice, so that any reaction based on observing the features should not be too severe.

Hence, two different reactions are implemented: to allow the opposite decision (delivery/no delivery) to be enforced, with a given probability, in the next construction, and/or to add a bonus/penalty for delivery that is used when evaluating the insertions during the GRASP-iterations. In the latter case it is important that a mechanism to forget the added bonus/penalty is included, so that it does not get too influential. Also, one might restrict the penalty/bonus so that it never turns an evaluation from positive to negative or vice versa (this might have different consequences depending on whether the GRASP is performed top-down or on all nodes simultaneously).

Computational Results and Analyses

Comprehensive results and analysis will be presented at the conference.

References

Adelman D. (2004). "A Price-Directed Approach to Stochastic Inventory/Routing", *Operations Research*, Vol 52, pp 499-514

Campbell A.M. and M.W.P. Savelsbergh. (2004). "A Decomposition Approach for the Inventory-Routing Problem", *Transportation Science*, Vol 38, pp 408 – 502

Glover F. (2000). “Multi-Start and Strategic Oscillation Methods - Principles to Exploit Adaptive Memory", In: **OR Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research**, Eds: M. Laguna and J.L. González-Velarde, pp 1 - 23

Kleywegt A.J., V.S. Nori and M.W.P. Savelsbergh. (2002). “The Stochastic Inventory Routing Problem with Direct Deliveries", *Transportation Science*, Vol 36, pp 94 - 118

Kleywegt A.J., V.S. Nori and M.W.P. Savelsbergh. (2004). "Dynamic Programming Approximations for a Stochastic Inventory Routing Problem", *Transportation Science*, Vol 38. pp 42 - 70

Prais M. and C.C. Ribeiro. (2000). "Reactive GRASP: An application to a matrix decomposition problem in TDMA traffic assignment", *INFORMS Journal of Computing*,Vol 12, pp 164 - 176

Resende M.G.C. and C.C. Ribeiro. (2003). "Greedy Randomized Adaptive Search Procedures", In: **Handbook of Metaheuristics**, Eds: F. Glover and G.A. Kochenberger Kluwer Academic Publishers, pp 219- 249