

A Computational Approach for the Inventory Routing Problem

Anton Kleywegt

Vijay Nori

Martin Savelsbergh

School of Industrial and Systems Engineering

Georgia Institute of Technology

Atlanta, GA 30332-0205

April 24, 1998

Abstract

Vendor managed inventory replenishment is a business practice in which vendors monitor their customers' inventories, and decide when and how much inventory should be replenished. The inventory routing problem addresses the coordination of inventory management and transportation. It needs to be solved to design a strategy that realizes the potential savings in inventory and transportation costs brought about by vendor managed inventory replenishment. The inventory routing problem is hard, especially if a large number of customers is involved. We formulate the inventory routing problem as a Markov decision process, and we propose approximation methods to find good solutions with reasonable computational effort. Computational results are presented for the direct delivery inventory routing problem.

1 Introduction

The inventory routing problem (IRP) is one of the core problems that has to be addressed when implementing the emerging business practice called vendor managed inventory replenishment (VMI). VMI is an innovative approach to inventory management that shifts responsibility for replenishment decisions from the buyer to the supplier. In a VMI partnership, the supplier makes the inventory replenishment decisions for the consuming organization.

Suppliers are attracted to VMI for several reasons. First, VMI may reduce production and inventory costs. Infrequent large orders from consumers force manufacturers to maintain surplus production capacity or excessive finished goods inventory to ensure responsive customer service. VMI helps to dampen peaks and valleys caused by orders, allowing smaller buffers of capacity and inventory. Second, VMI may reduce transportation costs. By coordinating the resupply process for different customers, as opposed to responding to orders when they come in, it may be possible to increase low-cost full truckload shipments and decrease or eliminate high-cost less-than-truckload shipments. Furthermore, it may be possible to create more efficient routes — for example, a single truck making multiple stops to replenish inventories for several nearby customers. Finally, VMI may increase service levels. From a buyer’s perspective, service is usually measured by product availability. With VMI, coordination of inventory levels and deliveries across multiple customers helps improve service. A non-critical delivery to one customer can be postponed for a day or two to accommodate a critical delivery to another customer. Similarly, a smaller-than-usual replenishment to one customer may enable a larger-than-usual shipment to another customer in dire need. Customers benefit too, in that they are assured that their most critical needs will get the closest attention. Without VMI, the supplier has a difficult task prioritizing customer shipments effectively.

Another important factor responsible for the increasing popularity of VMI is the rapidly decreasing cost of technology that allows the monitoring of customers’ inventories. VMI requires accurate and timely information about the inventory status of customers.

If VMI is a win-win situation for both suppliers and buyers, and relatively cheap monitoring technology is available, then it is natural to ask why VMI is not applied on a larger scale. The main reason is that it is a complex task to develop an effective replenishment strategy that realizes the potential savings in production, inventory and transportation costs through the coordination of these activities.

The inventory routing problem (IRP) addresses the coordination of inventory replenishment and transportation, and is the topic of this paper. Specifically, we study the problem of determining optimal policies for the distribution of a single product from a single supplier to multiple customers. For this purpose, the supplier operates a fleet of homogeneous vehicles. The demands at the customers are assumed to have a known probability distribution. The objective is to maximize the expected discounted value, incorporating sales revenues, production costs, transportation costs, inventory holding costs, and shortage penalties, over an infinite horizon.

In Section 2 the IRP is defined, and the related research is briefly reviewed. Section 3 presents a special

case of the IRP, namely the Direct Delivery IRP, and proposes an approximation method for hard instances of this problem. Computational results are presented in Section 4, in which the solution values of the proposed method are compared with the optimal values for small problems, as well as with the values of a heuristic proposed in the literature for small and medium sized problems. Future research in this area is briefly discussed in Section 5.

2 Problem Definition

2.1 Problem Description

A product is distributed from a plant to n customers, using a fleet of m homogeneous vehicles, each with known capacity C_V . Each customer i has a known storage capacity C_i , and a known demand probability distribution F_i . The demands on different days are independent and identically distributed. The supplier can measure the inventory X_{it} of each customer i at any time t . In practice these measurements are made with sensors and transmitted with telecommunications equipment, and a cost is incurred each time a measurement is made and transmitted. Because these measurement costs are rapidly decreasing, they are ignored in this paper. In practice, and in our model, inventory levels are measured and decisions are made at regular time intervals, usually once per day. Decisions are made regarding which customers' inventories to replenish, how much to deliver at each customer, how to combine customers into vehicle tours, and which vehicle tours to assign to each of the m vehicles. We call such a decision an itinerary. A vehicle can perform more than one tour per day, as long as all tours assigned to a vehicle together do not take more than a day to complete. Thus all vehicles are available at the beginning of each day, when the tasks for that day are assigned.

Travel times t_{ij} and travel costs c_{ij} on the arcs (i, j) of the distribution network are known, and include the time spent and the cost incurred at customers' sites. If quantity d_i is delivered at customer i , a reward of $r_i(d_i)$ is earned. Because demand is uncertain, there is often a positive probability that a customer runs out of stock, and thus shortages cannot always be prevented. Shortages are discouraged with a penalty $p_i(s_i)$ per day that the unsatisfied demand at customer i is s_i . If the inventory at customer i is x_i at the beginning of the day, an inventory holding cost of $h_i(x_i)$ is incurred. The objective is to choose a distribution policy that maximizes the expected discounted value (rewards minus costs) over an infinite time horizon.

2.2 Review of Related Research

The long-term dynamic and stochastic control problem presented above is extremely difficult to solve. Therefore, almost all approaches that have been proposed and investigated up to now solve only a short-term problem. In most of the early work, the short-term problems covered just a single day (Federgruen and Zipkin (1984), Golden, Assad, and Dahl (1984), Chien, Balakrishnan, and Wong (1989)). Later, the short-term problems were expanded to several days (Fisher et al. (1982), Bell et al. (1983), Dror, Ball and Golden (1985),

Dror and Ball (1987), Trudeau and Dror (1992), Bard et al. (1996), Jaillet et al. (1997)). The approaches differ mainly in how they decide which customers to include in the short-term problems and how they take the long-term effects of short-term decisions into account. Another stream of research analyzed the asymptotic behavior of certain simple policies (Anily and Federgruen (1990, 1991), Gallego and Simchi-Levi (1990), Bramel and Simchi-Levi (1995), Chan, Federgruen, and Simchi-Levi (1998)).

Minkov (1993) formulated the IRP as a Markov decision process (MDP), which is the modeling approach followed by us as well. Unlike our model, his model focused on the case with an unlimited number of vehicles. To overcome the computational difficulties caused by large state spaces, he proposed a decomposition heuristic. The heuristic solves a linear program to allocate joint transportation costs to individual customers, and then solves individual customer subproblems. The value functions of the subproblems are added to approximate the value function of the combined problem.

2.3 Problem Formulation

We formulate the IRP as a discrete time Markov decision process with the following components:

1. The state x is the current inventory at each customer. Thus the state space \mathcal{X} is $[0, C_1] \times [0, C_2] \times \dots \times [0, C_n]$. Let $X_t \in \mathcal{X}$ denote the state at time t .
2. The action space $\mathcal{A}(x)$ for each state x is the set of all itineraries that satisfy the tour duration constraints, such that the vehicles' capacities are not exceeded, and the customers' storage capacities are not exceeded after deliveries. Let $A_t \in \mathcal{A}(X_t)$ denote the itinerary chosen at time t .
3. The known demand distribution gives a known Markov transition function Q , according to which transitions occur, i.e., for any state $x \in \mathcal{X}$, and any itinerary $a \in \mathcal{A}(x)$,

$$P[X_{t+1} \in B \mid X_t = x, A_t = a] = \int_B Q[dy \mid x, a]$$

4. A reward of $r_i(d)$ is earned for delivery of quantity d at customer i . In addition, transportation costs, which depend on the vehicle tours chosen, inventory holding costs $h(x)$, and penalties $p_i(s)$ when customers run out of inventory, are taken into account. Let $g(x, a)$ denote the expected net reward if the process is in state x at the beginning of the day, and itinerary $a \in \mathcal{A}(x)$ is implemented.
5. The objective is to maximize the expected total discounted value over an infinite horizon. Let $\alpha \in [0, 1)$ denote the discount factor. Let $V^*(x)$ denote the optimal expected value given that the initial state is x , i.e.,

$$V^*(x) \equiv \sup_{\{A_t\}_{t=0}^{\infty}} E \left[\sum_{t=0}^{\infty} \alpha^t g(X_t, A_t) \mid X_0 = x \right] \quad (1)$$

The actions A_t are restricted such that $A_t \in \mathcal{A}(X_t)$ for each t , and A_t has to depend only on the history $(X_0, A_0, X_1, \dots, X_t)$ of the process up to time t , i.e., when we decide on an itinerary at time t , we do not know what is going to happen in the future.

Under certain conditions that are not very restrictive, the optimal expected value in (1) is attained by a policy in the class Π of stationary deterministic policies. It follows that for any $x \in \mathcal{X}$,

$$\begin{aligned} V^*(x) &= \sup_{\pi \in \Pi} E^\pi \left[\sum_{t=0}^{\infty} \alpha^t g(X_t, \pi(X_t)) \mid X_0 = x \right] \\ &= \sup_{a \in \mathcal{A}(x)} \left\{ g(x, a) + \alpha \int_{\mathcal{X}} V^*(y) Q[dy \mid x, a] \right\} \end{aligned} \quad (2)$$

To determine an optimal policy, the optimality equation (2) has to be solved. This requires a number of major computational tasks to be performed, including estimating the optimal value function V^* , estimating the integral in (2), and solving the maximization problem on the right hand side of (2) to determine the optimal itinerary for each state. The conventional algorithms for solving MDPs are practical only if the state space \mathcal{X} is small, and the optimization problem on the right hand side can be solved efficiently. These requirements are not satisfied by practical inventory routing problems, as the state space \mathcal{X} is usually extremely large, even if customers' inventories are discretized, and the vehicle routing problem, which is NP-hard, is a special case of the optimization problem on the right hand side of (2). Our approach is therefore to develop approximation methods based on the MDP formulation above. Such an approximation method is discussed in the next section, as applied to the specific case of the Direct Delivery IRP.

3 The Direct Delivery IRP

If the storage capacities and demands of the customers are sufficiently large relative to the vehicle capacity, then it is often optimal to deliver full vehicle loads or nearly full vehicle loads to customers. Gallego and Simchi-Levi (1990) analyzed a single-depot/multi-customer distribution system with constant demand rates, and without shortages. Transportation cost proportional to the total distance travelled, a linear inventory holding cost, and ordering cost at the customers are taken into account. They assumed availability of an unlimited number of vehicles. They studied conditions under which direct delivery is an efficient policy. A lower bound on the long-run average cost over all inventory-routing policies was derived, by adding the average inventory holding and ordering costs, using a traditional economic lot sizing model, and the long run transportation costs, obtained from the model of Haimovich and Rinnooy Kan (1985). An upper bound was derived on the average cost of a particular direct delivery policy as a function of economic lot sizes (ELS) of customers. It was concluded that the effectiveness (the ratio of the infimum of long-run average cost over all policies to the long-run average cost of the direct delivery policy) is at least 94% when ELS exceeds 71% of truck capacity.

Barnes-Schuster and Bassok (1997) studied a single-depot/multi-customer distribution system with random demands over an infinite horizon. Linear inventory holding costs and transportation costs between the depot and the retailers were incorporated. The fleet size was assumed to be unlimited. The objective was to study the cost effectiveness for the depot of using a particular direct delivery policy. The policy delivers as many full truck loads at a customer as the remaining capacity at the customer can accommodate. A

lower bound was developed for the expected long run average cost per period as a sum of the expected inventory holding cost, using an infinite horizon newsvendor problem, and the expected transportation cost, extending the bound developed by Haimovich and Rinnooy Kan (1985) for one retailer and a single period. The policy of direct delivery with full truck loads was simulated and compared with the lower bound. The results indicate that the policy performs well for situations where truck sizes are close to the means of the customer demand distributions.

In this section we consider the problem where only one customer is visited on each vehicle tour. This important special case of the IRP is called the direct delivery IRP (DDIRP). Although the hard routing and delivery quantity decisions of the IRP are much easier if only one customer is visited on each vehicle tour, the DDIRP is still a hard problem to solve if there are more than four customers and a limited number of vehicles, due to the size of the state space. Because direct deliveries are important in practice, and to study approximation methods without being hampered by hard routing problems, the DDIRP was investigated first.

To solve the DDIRP, a method was developed to construct a good approximation \hat{V} of the optimal value function V^* . Because transportation costs are separable over customers, the expected net reward per stage $g(x, a)$ is separable over customers,

$$g(x, a) = \sum_{i=1}^n g_i(x_i, a_i)$$

The only consideration that prevents the decomposition of the DDIRP into individual customer subproblems, is the limited number of vehicles that have to be assigned to customers each day. Under an optimal policy π^* , the assignment of vehicles depends on the inventory levels at the customers. For a countable state space \mathcal{X} , let $\nu^*(x)$ denote the stationary probability of state x under optimal policy π^* , assuming the existence of such stationary probabilities, unique for policy π^* . Then the conditional probability $p_i(k_i|y_i)$ that k_i vehicles are assigned to customer i under optimal policy π^* , given that the inventory level at customer i is y_i , is given by

$$p_i(k_i|y_i) = \frac{\sum_{\{x \in \mathcal{X}: x_i = y_i, \pi_i^*(x) = k_i\}} \nu^*(x)}{\sum_{\{x \in \mathcal{X}: x_i = y_i\}} \nu^*(x)}$$

Let $f_i(u_i)$ denote the probability that the demand of customer i is u_i . Then, given the current inventory level x_i and delivery quantity d_i at customer i , the probability $q_i(y_i, k_i|x_i, d_i)$ that under optimal policy π^* , at the beginning of the next day the inventory level at customer i is y_i , and k_i vehicles are assigned to customer i , is given by

$$q_i(y_i, k_i|x_i, d_i) = \begin{cases} f_i(x_i + d_i - y_i)p_i(k_i|y_i) & \text{if } y_i > 0 \\ \sum_{u_i = x_i + d_i}^{\infty} f_i(u_i)p_i(k_i|y_i) & \text{if } y_i = 0 \end{cases}$$

if no backlog is allowed. These probabilities define a MDP for customer i , with state (x_i, k_i) denoting that the inventory level at customer i is x_i and k_i vehicles *can* be dispatched to customer i . The admissible

actions a_i when the state is (x_i, k_i) , is the dispatching of up to k_i vehicles to customer i , and the delivery of an amount of product constrained by the vehicle capacity C_V and the customer storage capacity C_i . The expected net reward per stage, given state (x_i, k_i) and action a_i , is $g_i(x_i, a_i)$.

To construct an approximate value function \hat{V} , these individual customer MDPs are formulated and solved. To formulate these MDPs, the unknown probabilities $p_i(k_i|y_i)$ have to be estimated. These can be estimated by simulating the DDIRP under a chosen policy, and by exploiting some of the known structure of these probabilities, such as the fact that $p_i(0|y_i)$ is nondecreasing in y_i . Then the optimal values $V_i(x_i, k_i)$ of the individual customer MDPs are computed. These computations do not require much computational effort, as the state spaces of the individual customer MDPs are much smaller than that of the DDIRP. The approximate value function \hat{V} is given by the optimal value of the following nonlinear knapsack problem.

$$\begin{aligned} \hat{V}(x) &\equiv \max_{k_i} \sum_{i=1}^n V_i(x_i, k_i) \\ \text{s.t.} \quad &\sum_{i=1}^n k_i \leq m \end{aligned} \quad (3)$$

The idea is that the m vehicles are assigned to the n customers to maximize the values given by the resulting individual customer MDPs. The nonlinear knapsack problem is easily solved using dynamic programming. Although the resulting vehicle assignment may constitute a good policy, this knapsack problem is primarily solved to obtain the value function \hat{V} , and the policy $\hat{\pi}$ is given by a maximizer in the optimality equation, using \hat{V} to approximate the values of future states, as follows.

$$\hat{\pi}(x) \in \arg \max_{a \in \mathcal{A}(x)} \left\{ g(x, a) + \alpha \sum_{y \in \mathcal{X}} P[y | x, a] \hat{V}(y) \right\} \quad (4)$$

This method can also be interpreted as a multistage lookahead method, whereby the knapsack problem is solved to determine the tentative decision at the second stage, and the optimal value functions of the individual customer MDPs give the objective function for the knapsack problem to take into account the expected net reward from the second stage onwards.

The method described above can be improved using simulation embedded in an iterative algorithm. In this approach, the approximate value function \hat{V} is parameterized by a vector β of parameters, and is given by

$$\hat{V}(x, \beta) \equiv \beta_0 + \sum_{i=1}^n \beta_i V_i(x_i, k_i^*(x)) \quad (5)$$

where $k_1^*(x), \dots, k_n^*(x)$ is the optimal solution of the nonlinear knapsack problem (3). The probabilities $p_i(k_i|y_i)$ and the parameters β are updated by the simulation to improve the quality of the approximation. One such method is the following algorithm, that is similar to an algorithm called approximate policy iteration by Bertsekas and Tsitsiklis (1996).

Algorithm

1. Start with an initial policy π_0 . Set $t = 0$.

2. Simulate the DDIRP under policy π_t to estimate the probabilities $p_i(k_i|y_i)$.
3. Formulate and solve the individual customer MDPs.
4. Policy π_1 is given by (4), where \hat{V} is given by (3).
5. For a number of iterations, do steps 6 through 9.
6. Set $t = t + 1$.
7. Simulate the DDIRP under policy π_t to update the estimates of the probabilities $p_i(k_i|y_i)$ and the parameters β . Stochastic approximation and/or Kalman filter methods can be used, as explained in the following paragraphs.
8. With the updated estimates of the probabilities $p_i(k_i|y_i)$, formulate and solve the updated individual customer MDPs.
9. Policy π_{t+1} is given by (4), where \hat{V} is given by (5).

Van Roy et al. (1997) used a similar approach, called neuro-dynamic programming, to develop an approximation method for a retailer inventory management problem, that was introduced by Nahmias and Smith (1994). However, the type of value function approximation proposed above is quite different from that of Van Roy et al.

The estimates $p_i(k_i|y_i)_s$ of the probabilities $p_i(k_i|y_i)$ at each transition s of the simulation, are updated as follows. Let the state of customer i at transition s of the simulation be denoted by x_{is} , and let the number of vehicles dispatched to customer i be denoted by k_{is} . Let $N(s, y_i)$ denote the number of times that customer i has been in state y_i by transition s of the simulation. Then

$$p_i(k_i|y_i)_{s+1} = \begin{cases} \frac{(N_0 + N(s, y_i))p_i(k_i|y_i)_s + 1}{N_0 + N(s, y_i) + 1} & \text{if } x_{is} = y_i \text{ and } k_{is} = k_i \\ \frac{(N_0 + N(s, y_i))p_i(k_i|y_i)_s}{N_0 + N(s, y_i) + 1} & \text{if } x_{is} = y_i \text{ and } k_{is} \neq k_i \\ p_i(k_i|y_i)_s & \text{if } x_{is} \neq y_i \end{cases}$$

where N_0 represents a weight, equivalent to N_0 observations, assigned to the initial estimate $p_i(k_i|y_i)_0$. It follows that $p_i(k_i|y_i)_s \rightarrow p_i(k_i|y_i)$ as $s \rightarrow \infty$ with probability 1 for all states y_i that are visited infinitely often, i.e., for all states y_i such that $N(s, y_i) \rightarrow \infty$ as $s \rightarrow \infty$. Convergence with probability 1 can also be established for more general update methods.

The parameter estimates β_s can be updated as follows using stochastic approximation methods. At each transition s of the simulation, when the state is x_s , (note that notation is abused somewhat, since x_i denotes the state of customer i),

$$\beta_{s+1} = \beta_s + \gamma_s d_s z_s$$

where γ_s is the step size at iteration s ,

$$d_s = g(x_s, \pi_t(x_s)) + \alpha \hat{V}(x_{s+1}, \beta_s) - \hat{V}(x_s, \beta_s)$$

is the temporal difference, or

$$d_s = g(x_s, \pi_t(x_s)) + \alpha \sum_{y \in \mathcal{X}} P[y | x_s, \pi_t(x_s)] \hat{V}(y, \beta_s) - \hat{V}(x_s, \beta_s)$$

is the expected temporal difference,

$$z_s = \alpha \lambda z_{s-1} + \nabla_{\beta} \hat{V}(x_s, \beta_s)$$

is the eligibility vector, $\lambda \in [0, 1]$ is a memory parameter, and $\nabla_{\beta} \hat{V}(x_s, \beta_s)$ has components $\partial \hat{V}(x_s, \beta_s) / \partial \beta_{s0} = 1$, $\partial \hat{V}(x_s, \beta_s) / \partial \beta_{si} = V_i(x_{si}, k_i^*(x_s))$ for $i = 1, \dots, n$, and $k_1^*(x_s), \dots, k_n^*(x_s)$ is the optimal solution of non-linear knapsack problem (3). Bertsekas and Tsitsiklis (1996) have shown that, under mild conditions, with probability 1, as $s \rightarrow \infty$, the parameters β_s converge to the parameters that give the best approximation of the value function V^{π_t} of the current policy π_t , using the approximation based on the current set of basis functions. In this case the basis functions are the optimal value functions $V_i(x_i, k_i^*(x))$ of the individual customer MDPs. The parameters give the best approximation of the value function V^{π_t} in the following sense. Let $\nu^{\pi_t}(x)$ denote the stationary probability of state x under policy π_t , again assuming the existence of such unique stationary probabilities for policy π_t . Then, if $\lambda = 1$, the parameters β_s converge to the optimal solution of

$$\min_{\beta} \sum_{x \in \mathcal{X}} \nu^{\pi_t}(x) \left[V^{\pi_t}(x) - \hat{V}(x, \beta) \right]^2 \quad (6)$$

The parameters β can also be estimated as follows. The value function V^{π_t} of policy π_t satisfies

$$V^{\pi_t}(x) = g(x, \pi_t(x)) + \alpha \sum_{y \in \mathcal{X}} P[y | x, \pi_t(x)] V^{\pi_t}(y)$$

Then it seems natural to choose the parameters β to be the optimal solution of

$$\min_{\beta} \sum_{x \in \mathcal{X}} \nu^{\pi_t}(x) \left[\hat{V}(x, \beta) - \left(g(x, \pi_t(x)) + \alpha \sum_{y \in \mathcal{X}} P[y | x, \pi_t(x)] \hat{V}(y, \beta) \right) \right]^2 \quad (7)$$

This approach is called a Bellman error method. The optimal solutions of (6) and (7) are not the same in general. However, from our computational experience, they are usually quite similar for the DDIRP. The corresponding parameter estimates β_s can be computed as follows. Let the state at transition s of the simulation be denoted by x_s , and let $\tilde{V}(x_s) = (1 - \alpha, V_1(x_{1s}, k_1^*(x_s)) - \alpha \sum_{y \in \mathcal{X}} P[y | x_s, \pi_t(x_s)] V_1(y_1, k_1^*(y)), \dots, V_n(x_{ns}, k_n^*(x_s)) - \alpha \sum_{y \in \mathcal{X}} P[y | x_s, \pi_t(x_s)] V_n(y_n, k_n^*(y)))$ denote the column vector of basis function values corresponding to (7). Let $M_0 = 0$ be an $(n+1) \times (n+1)$ matrix, and let $M_{s+1} = M_s + \tilde{V}(x_s) \tilde{V}(x_s)^T$. Let $Y_0 = 0$ be an $(n+1) \times 1$ matrix, and let $Y_{s+1} = Y_s + \tilde{V}(x_s) g(x_s, \pi_t(x_s))$. Then β_s is the solution of the system of linear equations $M_s \beta_s = Y_s$. The solution is unique if M_s is nonsingular, for which it is necessary that $s \geq n+1$. Of course, it is not necessary to compute β_s at every transition s , but only when stopping criteria are to be checked. If the Markov chain under policy π_t is unichain, then the parameters β_s converge to the optimal solution of (7) as $s \rightarrow \infty$ with probability 1.

4 Computational Results

We have conducted various computational experiments to establish the viability of the proposed method. Our goals were to verify the correctness of our implementation, to establish the quality of the resulting

policies, and to identify the most time consuming components of the approach.

At this stage, the focus has been on establishing the quality of the policies produced by our algorithm. In a first set of experiments, the value functions of the policies produced by our algorithm were compared with the optimal value functions as well as the value functions of the policies produced by the method proposed by Chien, Balakrishnan, and Wong (1989) (CBW). This was done for several small instances of the DDIRP with varying characteristics, in terms of costs, customer storage capacities and demand probability distributions. The instances were small enough for the optimal value functions to be computed, and for exact policy evaluation to be performed. In a second set of experiments, we took several larger instances, again with varying characteristics, and compared the values of the policies produced by our algorithm to the values of the policies produced by the CBW method. They developed an integer programming based single-day model, in which long-term effects are taken into account by passing information from one day to the next. We adapted the CBW method to take the rewards and costs of the DDIRP into account. An integer program is formulated that maximizes the daily profit, which consists of revenue per unit delivered, transportation costs, inventory holding costs, and shortage costs. The integer program determines an assignment of vehicles to customers for each day. Once a solution is found for one day, the results are used to modify the rewards and costs for the next day. Unsatisfied demand at a customer in one day causes an increased revenue per unit delivered for that customer the next day. We used simulation to evaluate the performance of the policies for the larger instances. Starting at several initial states, both the policy produced by our algorithm and the policy of CBW were simulated. The integer program that forms the basis of the approach of CBW is given in Appendix A, and the problem instances used in both sets of experiments are given in Appendix B.

The preliminary results are promising. Table 1 shows the maximum percentage differences over all states between the value functions V^{π_1} , V^{π_2} , V^{π_3} , and V^{CBW} , and the optimal value function V^* for problem instances given in Appendix B. The initial policy π_1 is given by the optimal solution to the nonlinear knapsack problem (3), the second policy π_2 is obtained after one iteration of our algorithm, and the third policy π_3 is obtained after two iterations of our algorithm. V^{CBW} is the value function of the method of Chien, Balakrishnan, and Wong. The value functions were computed using the Gauss-Seidel policy evaluation algorithm. The results show that for these instances, the policies produced by our method are very close to optimal and are consistently better than the policy of CBW. Figure 1 shows V^* , V^{π_1} , V^{π_2} , V^{π_3} , and V^{CBW} for instance 3. It is not only the maximum differences between V^{π_i} and V^* which are smaller than the maximum differences between V^{CBW} and V^* , but V^{π_1} , V^{π_2} , and V^{π_3} are consistently better than V^{CBW} for all states.

For larger instances, the Gauss-Seidel policy evaluation algorithm cannot be used to compute the value functions. The most important reason for this is that the number of states becomes too large, and the available computer memory is not sufficient to store the values of all the states. For larger instances, policies π_3 and π_{CBW} were evaluated by randomly choosing a number of initial states, and simulating the system under the different policies starting from the chosen initial states. Each replication produced a sample

Instance	Max percentage difference			
	π_1	π_2	π_3	π_{CBW}
1	0.18%	0.002%	0.002%	0.54%
2	1.93%	0.22%	0.15%	5.22%
3	1.76%	0.009%	0.009%	10.02%
4	0.84%	0.15%	0.10%	16.36%

Table 1: Maximum percentage difference, $\max_{x \in \mathcal{X}} \{1 - V^\pi(x)/V^*(x)\}$, between optimal value functions V^* and value functions V^{π_1} , V^{π_2} , V^{π_3} , and V^{CBW} , for problem instances given in Appendix B, obtained using Gauss-Seidel policy evaluation.

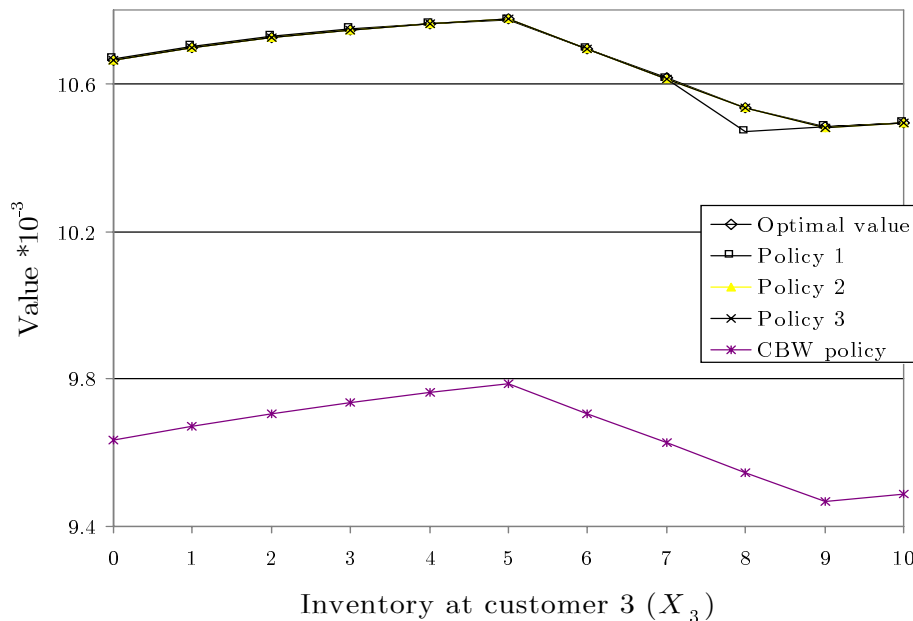


Figure 1: Optimal value function V^* , value functions V^{π_1} , V^{π_2} , V^{π_3} , and V^{CBW} , as a function of the inventory at customer 3, for instance 3 given in Appendix B.

path over a relatively long but finite time horizon of 800 time periods. The length of the time horizon was chosen to bound the discounted truncation error to less than 0.01. Six sample paths were generated for each combination of policy and initial state, for each instance. The results are shown in Table 2. Also shown are the sample averages μ and sample standard deviations σ over the six sample paths, as well as intervals $(\mu - 2\sigma, \mu + 2\sigma)$. As can be seen from the results, policy π_3 consistently performed better than policy π_{CBW} . The relative amounts by which policy π_3 performed better than policy π_{CBW} varied widely from instance to instance. We are investigating how these performance differences are related to the characteristics of the instances.

Instance 5											
Initial State	Policy	Replicates for Values ($*10^{-3}$)						μ	σ	$\mu - 2\sigma$	$\mu + 2\sigma$
1,4,3,2,0	π_3	81.47	81.37	81.31	82.33	81.72	80.63	81.47	0.55	80.37	82.58
	π_{CBW}	78.80	79.13	79.82	77.80	79.40	78.42	78.89	0.72	77.45	80.34
3,6,0,5,7	π_3	81.09	80.39	79.97	80.48	80.67	81.15	80.63	0.45	79.73	81.52
	π_{CBW}	78.61	78.54	78.95	77.09	78.65	77.88	78.29	0.68	76.92	79.66
5,1,1,4,4	π_3	81.77	81.51	81.42	80.43	81.30	81.25	81.28	0.46	80.37	82.19
	π_{CBW}	78.84	78.48	79.36	77.18	79.11	78.35	78.55	0.77	77.01	80.10
7,7,2,1,5	π_3	79.82	81.07	80.39	80.86	80.43	80.91	80.58	0.46	79.66	81.50
	π_{CBW}	78.55	78.12	78.60	76.66	78.39	77.89	78.03	0.72	76.58	79.48

Instance 6											
Initial State	Policy	Replicates for Values ($*10^{-3}$)						μ	σ	$\mu - 2\sigma$	$\mu + 2\sigma$
6,2,3,7,3	π_3	74.15	75.67	74.87	76.14	77.10	74.84	75.46	1.06	73.34	77.59
	π_{CBW}	69.56	69.20	68.82	70.02	69.92	68.55	69.34	0.59	68.16	70.53
0,10,0,10,4	π_3	75.27	75.70	75.15	76.43	77.28	75.19	75.84	0.86	74.12	77.55
	π_{CBW}	70.73	69.79	69.52	70.53	71.36	68.58	70.09	0.99	68.10	72.07
2,5,0,10,9	π_3	74.86	74.17	74.16	75.33	76.97	74.62	75.02	1.05	72.91	77.13
	π_{CBW}	70.18	69.09	69.30	70.25	70.05	68.03	69.48	0.86	67.76	71.21
10,6,8,6,0	π_3	74.42	73.62	73.54	74.86	75.94	74.16	74.42	0.89	72.64	76.21
	π_{CBW}	69.74	69.68	68.82	69.79	69.45	67.57	69.18	0.86	67.45	70.90

Instance 7											
Initial State	Policy	Replicates for Values ($*10^{-3}$)						μ	σ	$\mu - 2\sigma$	$\mu + 2\sigma$
2,3,0,5,6,4	π_3	52.18	52.64	53.16	52.81	54.81	54.04	53.27	0.98	51.32	55.23
	π_{CBW}	41.51	38.56	40.80	39.20	41.60	41.17	40.48	1.28	37.91	43.04
1,6,6,6,6,3	π_3	51.70	52.06	52.58	52.12	54.75	53.28	52.75	1.12	50.51	54.99
	π_{CBW}	41.32	37.99	39.65	38.48	40.19	39.79	39.57	1.20	37.17	41.97
3,0,1,5,5,3	π_3	52.03	52.90	53.32	52.93	55.44	56.01	53.77	1.58	50.61	56.94
	π_{CBW}	41.94	39.24	40.67	39.72	43.15	43.84	41.43	1.86	37.71	45.15
6,4,2,4,0,2	π_3	52.07	53.06	53.14	52.78	55.39	52.83	53.21	1.13	50.95	55.48
	π_{CBW}	40.52	37.74	37.72	39.09	42.97	40.49	39.76	2.01	35.74	43.77

Instance 8											
Initial State	Policy	Replicates for Values ($*10^{-3}$)						μ	σ	$\mu - 2\sigma$	$\mu + 2\sigma$
3,4,0,7,8,6	π_3	65.27	64.83	65.65	64.97	64.61	64.02	64.89	0.56	63.77	66.01
	π_{CBW}	36.52	35.77	34.33	40.11	37.24	35.34	36.55	2.01	32.54	40.57
1,8,8,8,8,4	π_3	65.07	64.33	65.44	64.76	64.49	65.14	64.87	0.42	64.03	65.71
	π_{CBW}	36.14	35.59	32.41	39.72	35.90	37.98	36.29	2.46	31.36	41.22
4,0,1,6,6,4	π_3	65.72	65.25	65.81	65.43	65.17	65.36	65.46	0.26	64.94	65.97
	π_{CBW}	37.35	36.05	33.10	40.94	39.24	35.21	36.98	2.83	31.33	42.64
7,5,2,5,0,3	π_3	65.67	64.93	66.04	65.36	64.87	64.76	65.27	0.51	64.26	66.28
	π_{CBW}	38.25	36.12	34.48	40.41	38.02	40.22	37.92	2.31	33.30	42.54

Table 2: Values V^{π_3} and V^{CBW} for problem instances given in Appendix B, obtained using simulation.

5 Future Work

The methods and results described in this paper represent only the initial stages of our research on the IRP. Much work remains to be done before some of the large IRPs encountered in practice can be solved. Even for the DDIRP several computational issues need to be addressed before larger instances can be solved. As the number of customers increases, the size of the action space $\mathcal{A}(x)$ increases rapidly. Thus methods should be developed to determine the best action, or at least a good action, in $\mathcal{A}(x)$, without enumerating all actions in $\mathcal{A}(x)$. The effort required to compute the expected value in (4) also increases as the number of customers increases, and thus there is a need for fast approximation methods to estimate the expected value. The convergence of the parameter estimates β_s is slow, especially when using stochastic approximation. Improvements in convergence rates will also improve the usefulness of the proposed approach.

An important extension of our work involves routing vehicles to more than one customer on a delivery tour. This version of the IRP is likely to be much harder than the DDIRP, since the optimization problem on the right hand side of (2) is much harder for the IRP than for the DDIRP. In the case of the IRP, this optimization problem involves solving both a vehicle routing problem, which is NP-hard, as well as determining the optimal quantities to be delivered to each customer on a delivery tour, which involves solving a nonlinear optimization problem with a nonunimodal objective function, as shown in Campbell et al. (1997).

Other issues that have to be addressed before IRPs can be solved in practice, include the estimation of the problem parameters. These include the rewards and costs, as well as the demand distributions. Estimating these parameters from noisy data lead to hard statistical and optimization problems. It is surprising how little work has been done in this area, since it is clear that the estimation of problem parameters from data is an essential activity for the formulation and solution of practical optimization problems.

References

- ANILY, S. AND FEDERGRUEN, A. 1990. One Warehouse Multiple Retailer Systems with Vehicle Routing Costs. *Management Science*, **36**, 92–114.
- ANILY, S. AND FEDERGRUEN, A. 1991. Rejoinder to “Comments on One Warehouse Multiple Retailer Systems with Vehicle Routing Costs”. *Management Science*, **37**, 1497–1499.
- BARD, J., HUANG, L., JAILLET, P. AND DROR, M. 1996. A Decomposition Approach to the Inventory Routing Problem with Satellite Facilities. not published.
- BARNES-SCHUSTER, D. AND BASSOK, Y. 1997. Direct Shipping and the Dynamic Single-depot/Multi-retailer Inventory System. *European Journal of Operational Research*, **101**, 509–518.

- BELL, W., DALBERTO, L., FISHER, M., GREENFIELD, A., JAIKUMAR, R., KEDIA, P., MACK, R. AND PRUTZMAN, P. 1983. Improving the Distribution of Industrial Gases with an On-Line Computerized Routing and Scheduling Optimizer. *Interfaces*, **13**, 4–23.
- BERTSEKAS, D. P. AND TSITSIKLIS, J. N. 1996. *Neuro-Dynamic Programming*. Athena Scientific, New York, NY.
- BRAMEL, J. AND SIMCHI-LEVI, D. 1995. A Location Based Heuristic for General Routing Problems. *Operations Research*, **43**, 649–660.
- CAMPBELL, A. M., CLARKE, L. W., KLEYWEGT, A. J. AND SAVELSBERGH, M. W. P. 1997. The Inventory Routing Problem, *Technical Report TLI-97-07*, School of Industrial and Systems Engineering, Georgia Institute of Technology.
- CHAN, L. M. A., FEDERGRUEN, A. AND SIMCHI-LEVI, D. 1998. Probabilistic Analysis and Practical Algorithms for Inventory-Routing Models. *Operations Research*, **46**, 96–106.
- CHIEN, T., BALAKRISHNAN, A. AND WONG, R. 1989. An Integrated Inventory Allocation and Vehicle Routing Problem. *Transportation Science*, **23**, 67–76.
- DROR, M. AND BALL, M. 1987. Inventory/Routing: Reduction from an Annual to a Short Period Problem. *Naval Research Logistics Quarterly*, **34**, 891–905.
- DROR, M., BALL, M. AND GOLDEN, B. 1985. A Computational Comparison of Algorithms for the Inventory Routing Problem. *Annals of Operations Research*, **4**, 3–23.
- FEDERGRUEN, A. AND ZIPKIN, P. 1984. A Combined Vehicle Routing and Inventory Allocation Problem. *Operations Research*, **32**, 1019–1036.
- FISHER, M., GREENFIELD, A., JAIKUMAR, R. AND KEDIA, P. 1982. Real-Time Scheduling of a Bulk Delivery Fleet: Practical Application of Lagrangean Relaxation, *Technical report*, The Wharton School, University of Pennsylvania, Department of Decision Sciences.
- GALLEGO, G. AND SIMCHI-LEVI, D. 1990. On the Effectiveness of Direct Shipping Strategy for the One-Warehouse Multi-Retailer R-Systems. *Management Science*, **36**, 240–243.
- GOLDEN, B., ASSAD, A. AND DAHL, R. 1984. Analysis of a Large Scale Vehicle Routing Problem with an Inventory Component. *Large Scale Systems*, **7**, 181–190.
- HAIMOVICH, M. AND RINNOOY KAN, A. H. G. 1985. Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, **10**, 527–542.
- JAILLET, P., HUANG, L., BARD, J. AND DROR, M. 1997. A Rolling Horizon Framework for the Inventory Routing Problem. not published.

- MINKOFF, A. S. 1993. A Markov Decision Model and Decomposition Heuristic for Dynamic Vehicle Dispatching. *Operations Research*, **41**, 77–90.
- NAHMIA, S. AND SMITH, S. A. 1994. Optimizing Inventory Levels in a Two-echelon Retailer System with Partial Lost Sales. *Management Science*, **40**, 582–596.
- TRUDEAU, P. AND DROR, M. 1992. Stochastic Inventory Routing: Route Design with Stockouts and Route Failures. *Transportation Science*, **26**, 171–184.
- VAN ROY, B., BERTSEKAS, D. P., LEE, Y. AND TSITSIKLIS, J. N. 1997. A Neuro-dynamic Programming Approach to Retailer Inventory Management, *Technical report*, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA.

Appendix A

In this appendix, we present a slightly modified version of the method proposed by Chien, Balakrishnan, and Wong (1989) (CBW), adapted for the DDIRP. At the start of each day, an integer program is solved to determine the vehicle assignments for that day. The parameters and variables of the integer program are given in the following table.

n :	Number of customers
r_i :	Revenue earned per unit delivered to customer i
c_i :	Round-trip travel cost between depot and customer i
h_i :	Holding cost per unit stored in inventory at customer i
p_i :	Penalty per unit short at customer i
C_i :	Storage capacity of customer i
X_i :	Initial inventory at customer i
D_i :	Estimate of the demand of customer i
m :	Number of vehicles
C_V :	Vehicle capacity (for all vehicles)
d_{ij} :	Quantity delivered at customer i by vehicle j
y_{ij} :	1 if vehicle j is assigned to customer i , 0 otherwise
δ_i :	Lower bound on the final inventory at customer i
η_i :	Upper bound on the shortage at customer i

The integer program is given below.

$$\text{Maximize } \sum_{i,j} r_i d_{ij} - \sum_{i,j} c_i y_{ij} - \frac{1}{2} \sum_i h_i \left(X_i + \sum_j d_{ij} + \delta_i \right) - \alpha \sum_i p_i \eta_i \quad (8)$$

$$\text{subject to } \sum_j d_{ij} \leq C_i - X_i, \quad \forall i \quad (9)$$

$$d_{ij} - C_V y_{ij} \leq 0, \quad \forall i, j \quad (10)$$

$$\sum_i y_{ij} \leq 1, \quad \forall j \quad (11)$$

$$X_i + \sum_j d_{ij} - D_i \leq \delta_i \quad \forall i \quad (12)$$

$$- \left(X_i + \sum_j d_{ij} - D_i \right) \leq \eta_i \quad \forall i \quad (13)$$

$$\delta_i, \eta_i, d_{ij} \geq 0 \quad \forall i, j$$

$$y_{ij} \quad \text{binary} \quad \forall i, j$$

Constraints (9) ensure that the total amount of product delivered to a customer does not exceed the customer's remaining capacity. Constraints (10) ensure that the amount of product delivered to a customer

by a single vehicle is no more than the vehicle capacity. Constraints (11) ensure that a vehicle is assigned to at most one customer. Constraints (12) and (13) determine, for each customer, the final inventory or shortage at the end of the day. The inventory at the end of the day is computed as $\max\{0, X_i + \sum_j d_{ij} - D_i\}$, where D_i is taken to be the maximum demand as suggested by CBW. Likewise, shortage is computed as $\max\{0, -X_i - \sum_j d_{ij} + D_i\}$. Note that by the choice of D_i , the holding costs are underestimated and the shortage costs are overestimated. This may result in a conservative low-risk policy.

The objective function consists of four parts: the revenue earned, the transportation cost, the inventory holding cost and the shortage cost. As proposed by CBW, the revenue earned per unit is given by $r_i + p_i$ or r_i depending on whether or not there was a shortage in the previous period. Their model has been modified slightly by incorporating a linear inventory holding cost given by half the sum of inventory after delivery and inventory at the end of the day, times the per unit holding cost. We have also assumed that shortages occur at the end of the day and are discounted at a rate α to the beginning of the day. Finally, it is assumed that the depot has an unlimited supply of the product.

Appendix B

Instances for which the optimal value functions were computed.

Instance 1											
i	C_i	f_i						c_i	r_i	p_i	h_i
		0	1	2	3	4	5				
1	5	0	0.2	0.2	0.2	0.2	0.2	50	80	50	15
2	5	0	0	0.5	0	0.4	0.1	20	100	50	15
3	5	0	0.4	0.1	0.2	0.1	0.2	20	80	50	15
$n = 3, m = 2, C_V = 5, \alpha = 0.98$											

Instance 2																
i	C_i	f_i											c_i	r_i	p_i	h_i
		0	1	2	3	4	5	6	7	8	9	10				
1	10	0	0	0	0.5	0	0	0	0	0	0	0.5	120	80	40	10
2	10	0	0	0	0	0.5	0	0	0	0	0	0.5	80	80	50	5
3	10	0	0	0	0	0	0.5	0	0	0	0	0.5	120	100	50	5
$n = 3, m = 2, C_V = 7, \alpha = 0.98$																

Instance 3																	
i	C_i	f_i											c_i	r_i	p_i	h_i	
		0	1	2	3	4	5	6	7	8	9	10					
1	10	0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	120	80	50	5
2	10	0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	120	80	50	5
3	10	0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	120	80	50	5
$n = 3, m = 2, C_V = 5, \alpha = 0.98$																	

Instance 4										
i	C_i	f_i					c_i	r_i	p_i	h_i
		0	1	2	3	4				
1	4	0	0	0.5	0	0.5	100	80	50	5
2	4	0	0.25	0.5	0	0.25	120	80	50	10
3	4	0	0	0.5	0.25	0.25	120	80	50	5
4	4	0	0.5	0	0	0.5	100	80	60	4
$n = 4, m = 3, C_V = 3, \alpha = 0.98$										

Instances evaluated with simulation.

Instance 5														
i	C_i	f_i									c_i	r_i	p_i	h_i
		0	1	2	3	4	5	6	7	8				
1	8	0	0.4	0	0.1	0.1	0	0	0.2	0.2	40	100	50	2
2	8	0	0.1	0	0	0.4	0.4	0	0	0.1	30	120	50	3
3	8	0	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	45	80	70	5
4	8	0	0	0	0.3	0.3	0.3	0.1	0	0	50	80	40	4
5	8	0	0	0.1	0.2	0	0.2	0	0.2	0.3	40	80	30	3
$n = 5, m = 3, C_V = 8, \alpha = 0.98$														

Instance 6																
i	C_i	f_i											c_i	r_i	p_i	h_i
		0	1	2	3	4	5	6	7	8	9	10				
1	10	0	0	0.2	0.3	0.4	0	0	0	0	0	0.1	100	180	40	15
2	10	0	0	0	0	0.4	0.4	0	0	0	0	0.2	100	80	50	5
3	10	0	0.4	0	0	0	0	0	0	0.5	0	0.1	120	40	30	15
4	10	0	0	0	0.3	0	0	0.3	0	0.3	0	0.1	140	80	25	5
5	10	0	0	0.1	0.2	0	0.2	0	0.2	0	0.2	0.1	120	120	50	15
$n = 5, m = 3, C_V = 8, \alpha = 0.98$																

Instance 7													
i	C_i	f_i							c_i	r_i	p_i	h_i	
		0	1	2	3	4	5	6					
1	6	0	0	0.5	0	0.5	0	0	60	80	40	5	
2	6	0	0.1	0.1	0.3	0.3	0.1	0.1	50	80	50	8	
3	6	0	0	0.5	0	0	0	0.5	40	80	30	5	
4	6	0	0.1	0.1	0.3	0.3	0.1	0.1	40	80	20	5	
5	6	0	0.5	0	0	0	0.5	0	50	80	50	15	
6	6	0	0.1	0.1	0.3	0.3	0.1	0.1	60	80	70	5	
$n = 6, m = 4, C_V = 5, \alpha = 0.98$													

Instance 8														
i	C_i	f_i									c_i	r_i	p_i	h_i
		0	1	2	3	4	5	6	7	8				
1	8	0	0	0	0	0.5	0	0	0	0.5	100	80	40	5
2	8	0	0	0	0.5	0	0	0	0	0.5	100	80	50	5
3	8	0	0	0	0	0	0.5	0	0	0.5	120	80	30	15
4	8	0	0	0.5	0	0	0	0	0	0.5	120	80	20	5
5	8	0	0	0	0	0	0	0.5	0	0.5	120	80	50	15
6	8	0	0.5	0	0	0	0	0	0	0.5	120	80	70	5
$n = 6, m = 5, C_V = 6, \alpha = 0.98$														