Inventory Routing Problems

Martin Savelsbergh
Goals

• Introduce Inventory Routing Problems
• Introduce Solution Approaches for Inventory Routing Problems
• Introduce Inventory Routing Game
Inventory Routing

Inventory Management

↓

Vehicle Routing
Conventional Inventory Management

- Customer
  - monitors inventory levels
  - places orders
- Vendor
  - manufactures/purchases product
  - assembles order
  - loads vehicles
  - routes vehicles
  - makes deliveries

You call – We haul
Problems with Conventional Inventory Management

- Large variation in demands on production and transportation facilities
- Workload balancing
- Utilization of resources
- Unnecessary transportation costs
- Urgent vs nonurgent orders
- Setting priorities
Vendor Managed Inventory

- **Customer**
  - trusts the vendor to manage the inventory
- **Vendor**
  - monitors customers’ inventory
    - customers call/fax/e-mail
    - remote telemetry units
    - set levels to trigger call-in
  - controls inventory replenishment & decides
    - when to deliver
    - how much to deliver
    - how to deliver

You rely – We supply
Vendor Managed Inventory

- VMI transfers inventory management (and possibly ownership) from the customer to the supplier
- VMI synchronizes the supply chain through the process of collaborative order fulfillment
Advantages of VMI

- **Customer**
  - less resources for inventory management
  - assurance that product will be available when required

- **Vendor**
  - more freedom in when & how to manufacture product and make deliveries
  - better coordination of inventory levels at different customers
  - better coordination of deliveries to decrease transportation cost
Inventory Routing

- Decide when to deliver to a customer
- Decide how much to deliver to a customer
- Decide on the delivery routes
Inventory Routing

Inventory Management

Vehicle Routing

Long-Term Problem
Inventory Costs

+ Transportation Costs
Single Customer (Single Link)

Determine shipping policies that optimize the trade-off between:

- Inventory cost
- Transportation cost

transp. cost

inv. cost
Problem Description

Fleet of vehicles:
- transportation capacity $r = 1$
- transportation cost $c$ (A→B→A)
Problem Description

- Volume produced in A per unit time: \( v \)
- Volume consumed in B per unit time: \( v \)
- Inventory cost per unit time: \( h \)
Goal

Determine shipping policies that minimize inventory cost + transportation cost
The Continuous Variant

✓ Single frequency $f$
✓ Continuous time between shipments $t = 1/f$
✓ Single vehicle

\[
\begin{align*}
\min & \quad hv t + \frac{c}{t} \\
vt & \leq r \\
t & \geq 0
\end{align*}
\]

Optimal solution

\[t^* = \min \left( \sqrt{\frac{c}{hv}}, \frac{r}{v} \right)\]
Minimum Intershipment Times

A practical constraint: Minimum intershipment times, e.g., 1 day

- ZIO (Zero Inventory Ordering)
- FBPS (Frequency Based Periodic Shipping)
Zero Inventory Policy

- Minimum inter-shipment time
- Single frequency $f$
- Continuous time between shipments $t = \frac{1}{f}$

A shipment is performed when the inventory level of the products is zero

$$t^* = \min \left\{ \max \left\{ 1, \sqrt{\frac{c\left\lceil \frac{v}{h} \right\rceil}{h}}, \frac{\left\lceil \frac{v}{h} \right\rceil}{v} \right\} \right\}$$
Frequency Based Periodic Shipping Policies

- Minimum intershipment time
- One or more frequencies
- Integer time between consecutive shipments

Single frequency

Double frequency
Best Single Frequency Policy

$1/k^*: \text{best single frequency}$

**Lemma:** $1 \leq k^* \leq \bar{k} = \max \left\{ \left( \left\lfloor \nu \right\rfloor - \nu \right) \frac{c}{h}, 1 \right\}$

$z^{\text{Best SF}} = \min_{1 \leq k \leq \bar{k}} \left( hk + \frac{c}{k} \left\lceil \nu k \right\rceil \right)$

0 1 2 ...

\[ \text{The Logistics Institute} \]
\[ \text{Georgia Institute of Technology} \]
\[ www.tli.gatech.edu \]
Best Double Frequency Policy

\(\frac{1}{k_1^*}, \frac{1}{k_2^*}=: \text{best frequencies}\)

**Lemma:** \(1 \leq k_1^* \leq \bar{k}_1 = ...\)

\(k_1^* < k_2^* \leq \bar{k}_2 = ...\)

\[ z^{\text{Best DF}} = \min_{1 \leq k_1 \leq \bar{k}_1} \left( \min_{k_1 < k_2 \leq \bar{k}_2} \left( z^{SF}(k_1), \min_{k_1 < k_2 \leq \bar{k}_2} z^{DF}(k_1, k_2) \right) \right) \]
Multiple Customers

Determine shipping policies that optimize the trade-off between:

- Inventory cost
- Transportation cost

Inventory Routing Problem
Deterministic order-up-to level policy

• Each customer defines a minimum and a maximum level of the inventory

• The plant determines the set of delivery time instants

• Every time a customer is visited, the shipping quantity is such that the maximum level of the inventory is reached at the customer
Order-Up-To

Inventory at retailer $s$

Maximum level

Starting level

Minimum level

$U_s$

$L_s$

Time

THE LOGISTICS INSTITUTE
Georgia Institute of Technology
www.tli.gatech.edu
Problem Formulation

Decisions:

• For each customer \( s \):
  the set of delivery time instants
For each delivery time instant \( t \):
  the route followed by the vehicle

Objective function:

\[
\text{Min} \quad \text{Inv. Plant} + \text{Inv. Customer} + \text{Routing}
\]

Key Assumption: Deliveries are instantaneous
Transportation Costs
Inventory Routing Problem

• Single Plant
  – single facility
  – single product
  – set of n customers
  – set of m homogenous vehicles of capacity Q
Inventory Routing Problem

- Each customer:
  - storage capacity
  - initial inventory
  - product usage rate OR
  - probability distribution of product usage
Inventory Routing Problem

• Objective
  – Minimize distribution costs without causing any stock-outs over a finite horizon OR
  – Maximize the expected total discounted value (rewards minus costs) over an infinite horizon
Inventory Routing Problem

- Extensions
  - Operating modes
  - Delivery time windows
  - Delivery times (fixed plus variable part)
Inventory Routing

Even simple situations are non-trivial

There are 14 possible customer combinations

A-B-C-D    A-B
A-B-C    A-C
A-B-D    A-D
A-C-D    B-C
A    B-D
B    C-D
C    D

There are an infinite number of possible delivery volumes

Daily Use 1000 3000 2000 1500
Max. Delv. 5000 3000 2000 4000

Truck capacity is 5000
The “natural” solution

**Daily schedule**
trip 1: deliver 1000 to A & 3000 to B
trip 2: deliver 2000 to C & 1500 to D

**420 miles per day**

A better solution

**Day 1 schedule**
trip 1: deliver 3000 to B & 2000 to C

**Day 2 schedule**
trip 1: deliver 2000 to A & 3000 to B
trip 2: deliver 2000 to C & 3000 to D

**380 miles per day**
Complexity

- Single customer problem?
- Two customer problem?
Deterministic $d$-day policy

\[ v_T(d) = \max(0, \left\lceil \frac{T u - I}{\min(C, Q)} \right\rceil) c \]
Single Customer Problem

Stochastic $d$-day policy

- Probability that stockout occurs on day $j$
- Stockout cost
- Probability that no stockout occurs
- Delivery cost

Cost of filling up every $d$ days over $T$ day period:

$$v_T(d) = \sum_{j=1}^{d-1} p_j (v_{T-j}(d) + S) + (1-p)(v_{T-d}(d) + c)$$

$$d > T : v_T(d) = \sum_{j=1}^{T} p_j (v_{T-j}(d) + S)$$

$$d \leq T : v_T(d) = \sum_{j=1}^{d-1} p_j (v_{T-j}(d) + S) + (1-p)(v_{T-d}(d) + c)$$

$$v_T(d) = \alpha(d) + \beta(d)T + f(T, d)$$

$\alpha(d)$ constant, $f(T,d)$ goes to zero exponentially fast as $T \to \infty$

$$\beta(d) = \frac{pS + (1-p)c}{\sum_{j=1}^{d} j p_j}$$
An optimal constant replenishment period strategy over a large $T$-day planning period will correspond to choosing $d^*$ to minimize $\beta(d)$

$$\beta(d) = \frac{pS + (1-p)c}{\sum_{j=1}^{d} j p_j}$$

- Expected cost of a delivery
- Expected number of days between deliveries
Best d-day policy

Demand: uniformly in [1,20]
Deliver cost: 40
Stockout cost: 50
Optimal policy
Two customer problem

Deterministic d day policy:

Always individually:

\[ u_T = \max(0, \left[ \frac{T \cdot u_1}{\min(C_1, Q)} \right])c_1 + \max(0, \left[ \frac{T \cdot u_2}{\min(C_2, Q)} \right])c_2 \]

Always together:

\[ u_T = \left[ \frac{T}{\min\left(\frac{C_1}{u_1}, \frac{C_2}{u_2}, \frac{Q}{u_1+u_2}\right)} \right]c_{12} \]

Sometimes individually, sometimes together?

What if one cannot take a full truckload?

What if the customers are close together?

How much to deliver to each of them on a combined route?
Two Customer Problem

- Stochastic policy:
  - Storage capacity: 20
  - \( P[\text{demand} = 0] = 0.4, P[\text{demand} = 10] = 0.6 \)
  - Shortage penalty: 1000 customer 1; 1005 customer 2
  - Vehicle capacity: 10
  - Individual routes: 120; Combined route: 180

- Infinite horizon Markov Decision Process
  - Minimize expected total discounted cost
Two Customer Problem

Inventories $X(1) = 7, X(2) = 7$
Bounds

- Customer usage during period: $u_i$
- Customer storage capacity: $C_i$
- Vehicle capacity: $Q$

Minimize total mileage $D^*$

subject to

- Total volume delivered to customer $i$: $u_i$
- Maximum volume delivered per trip: $Q$
- Maximum quantity delivered to customer $i$: $\min(C_i, Q)$
Bounds

- Simple bounds on total mileage

  - Assume \( C_i \geq Q \)

  \[
  LB_1: \sum_{i \in I} \frac{u_i}{Q} 2t_{0i}
  \]

  - Assume direct delivery

  \[
  UB_1: \sum_{i \in I} \frac{u_i}{\min(C_i, Q)} 2t_{0i}
  \]

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Set of customers</td>
</tr>
<tr>
<td>( u_i )</td>
<td>Usage of customer ( i ) (period)</td>
</tr>
<tr>
<td>Q</td>
<td>Vehicle tank capacity</td>
</tr>
<tr>
<td>( t_{ij} )</td>
<td>Travel distance from ( i ) to ( j )</td>
</tr>
</tbody>
</table>
Two Customer Analysis

Plant

Customer

Patterns:
- (Q,0)
- (0,C_2)
- (Q-C_2,C_2)

Case 1: If \( \frac{w_2}{C_2} (Q - C_2) \geq u_1 \), then
\[
D^* = \frac{w_2}{C_2} 2t_{02}.
\]

Case 2: If \( \frac{w_2}{C_2} (Q - C_2) < u_1 \), then
\[
D^* = \frac{w_2}{C_2} 2t_{02} + \frac{u_1 - \frac{w_2}{C_2} (Q - C_2)}{Q} 2t_{01}.
\]
Two Customer Analysis

Assume: \[
\frac{Q}{Q - C_1} (t_{12} + t_{02} - t_{01}) < 2t_{02}
\]

Extra mileage

Case 1: If \( \frac{u_1}{C_1} (Q - C_1) \geq u_2 \), then
\[
D^* = \frac{u_2}{Q - C_1} (t_{01} + t_{12} + t_{02}) + \frac{u_1 - u_2}{Q - C_1} \frac{C_1}{2t_{01}}
\]

Case 2: \( \frac{u_1}{C_1} (Q - C_1) < u_2 \), then
\[
D^* = \frac{u_1}{C_1} (t_{01} + t_{12} + t_{02}) + \frac{u_2 - \frac{u_1}{C_1} (Q - C_1)}{Q} 2t_{02}.
\]

Patterns:
- \((C_1, 0)\)
- \((0, Q)\)
- \((C_1, Q - C_1)\)
Improved Bounds

**Delivery Patterns**

\[ P_j = (d_{j1}, d_{j2}, ..., d_{jn}) \] is feasible if
\[ \sum_{i \in I} d_{ji} \leq Q \quad \text{and} \quad 0 \leq d_{ji} \leq C_i \quad \forall i \in I. \]

\[ \delta(P_j) = \{i \in I : d_{ji} > 0\} : \text{Customers visited in } P_j \]

\[ c(P_j) : \text{The cost of delivery pattern } P_j \text{ - optimal TSP value} \]

\[ \mathcal{P} : \text{Set of all feasible delivery patterns} \]

**Pattern Selection LP**

\[ \begin{align*}
D^* &= \min \sum_{P_j \in \mathcal{P}} c(P_j) \times_j \\
\text{s.t.} \quad &\sum_{P_j \in \mathcal{P}} d_{ji} \times_j \geq u, \quad \forall i \in I \\
&\times_j \geq 0
\end{align*} \]

\[ x_j : \text{How many times should pattern } P_j \text{ be used} \]
Obstacles

- **Obstacle I**: Infinite number of feasible delivery patterns

- **Obstacle II**: The calculation of the cost of each delivery pattern involves the solution of a traveling salesman problem
Obstacle I

- **Base Pattern**
  
  A feasible delivery pattern $P$ is a base pattern if at most one customer, say $k$, in $\delta(P)$ receives a delivery quantity less than $\min(C_k, Q)$, and, in that case, the delivery quantity is $Q - \sum_{i \in \delta(P) \setminus k} C_i$.

- **Theorem**
  
  The base patterns are sufficient to find an optimal solution to the Pattern Selection LP.

  Number of columns of Pattern Selection LP is finite.
Focus on upper and lower bounds on $D^*$ instead of $D^*$ itself.

**Lower bound ($LB_k$):**
- If $C_i < Q/k$, then assume $C_i = Q/k$
- $LB_1 \leq LB_2 \leq \ldots \leq D^*$
- If $\min_{i \in 1}(C_i) \geq Q/k$, then $LB_k = D^*$
- $|\delta(P)| \leq k$ for any base pattern $P$

**Upper bound ($UB_k$):**
- At most $k$ stops in a tour
- $UB_1 \geq UB_2 \geq \ldots \geq D^*$
- If $\left\lceil Q/\min_{i \in 1}(C_i) \right\rceil \leq k$, then $UB_k = D^*$
- $|\delta(P)| \leq k$ for any base pattern $P$
Dominance

- Do we need base pattern $P$?

\[
z = \min_{\{j: \delta(P_j) \subsetneq \delta(P)\}} \sum_{j: \delta(P_j) \subsetneq \delta(P)} c(P_j)\lambda_j
\]

\[
\text{s.t. } \sum_{j: \delta(P_j) \subsetneq \delta(P)} d_{ji} \lambda_j \geq d_i, \quad \forall i \in \delta(P)
\]

\[
\lambda_j \geq 0
\]

If $z \leq c(P)$,

then the base patterns with $\lambda_j > 0$ collectively dominate $P$.

Base pattern $P$ can be eliminated from the Pattern Selection LP.
Consider base pattern $P$ with $d_4 < C_4$

### Condition I

\[ d_{44} = \min(Q, C_4) \]

### Condition II

\[ d_{ij4} = \min(Q - C_i - C_j, C_4) \]

### Condition III

\[ d_{ij4} = \min(Q - C_i - C_j, C_4) \]

\[
c(P) \geq c(P_{123}) + \frac{d_4}{\min(C_4, Q)} c(P_4)
\]

<table>
<thead>
<tr>
<th>Instances</th>
<th>$n$</th>
<th>before</th>
<th>after</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>136</td>
<td>5,015,046</td>
<td>3,029,980</td>
</tr>
<tr>
<td>2</td>
<td>157</td>
<td>7,665,722</td>
<td>4,336,466</td>
</tr>
<tr>
<td>3</td>
<td>169</td>
<td>9,086,385</td>
<td>5,420,907</td>
</tr>
<tr>
<td>4</td>
<td>147</td>
<td>15,180,701</td>
<td>8,838,137</td>
</tr>
<tr>
<td>5</td>
<td>157</td>
<td>14,471,228</td>
<td>8,975,615</td>
</tr>
<tr>
<td>6</td>
<td>194</td>
<td>22,575,528</td>
<td>16,640,122</td>
</tr>
</tbody>
</table>
Implementation: Sifting Approach

- Specialized solver for LPs with a large ratio of number of columns to number of rows

A Partial LP
 Subset of full set of columns

Check reduced costs for the remaining columns

Optimal solution

<table>
<thead>
<tr>
<th>Instance</th>
<th>n</th>
<th># of patterns</th>
<th># of iterations</th>
<th>default(sec)</th>
<th>sifting(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>136</td>
<td>3,029,980</td>
<td>5</td>
<td>85.66</td>
<td>77.09</td>
</tr>
<tr>
<td>2</td>
<td>157</td>
<td>4,336,466</td>
<td>6</td>
<td>118.37</td>
<td>121.34</td>
</tr>
<tr>
<td>3</td>
<td>169</td>
<td>5,420,907</td>
<td>6</td>
<td>155.29</td>
<td>152.56</td>
</tr>
<tr>
<td>4</td>
<td>147</td>
<td>8,838,137</td>
<td>5</td>
<td>360.20</td>
<td>240.83</td>
</tr>
<tr>
<td>5</td>
<td>157</td>
<td>8,975,615</td>
<td>6</td>
<td>397.33</td>
<td>254.81</td>
</tr>
<tr>
<td>6</td>
<td>194</td>
<td>16,640,122</td>
<td>6</td>
<td>675.98</td>
<td>533.56</td>
</tr>
</tbody>
</table>
36 plants, ~2000 customers
Inventory Routing

Schedule 1: 420 miles per day
(1000,3000,0,0),(0,0,2000,1500)

Schedule 2: 380 miles per day
(0,3000,2000,0)
(2000,3000,0,0), (0,0,2000,3000)

Pattern Selection LP with T=1day
Optimal Objective Value : 380
0.5 : (0,3000,2000,0)
0.5 : (2000,3000,0,0)
0.5 : (0,0,2000,3000)

Pattern Selection LP found schedule 2 and it shows no better schedule exists!

<table>
<thead>
<tr>
<th>Customer</th>
<th>$C_i$</th>
<th>$u_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5000</td>
<td>1000</td>
</tr>
<tr>
<td>2</td>
<td>3000</td>
<td>3000</td>
</tr>
<tr>
<td>3</td>
<td>2000</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>4000</td>
<td>1500</td>
</tr>
</tbody>
</table>

$Q = 5000$
Solution Approaches

- Deterministic
  - Based on average product usage
- Stochastic
  - Based on probability distribution of product usage
Two Phase Approach

– Phase I: Determine which customers should receive a delivery on each day of the planning period and how much
– Phase II: Create the precise delivery routes for each day

Rolling horizon approach
Deterministic Solution Approach

Two Phase Approach

– Phase I: Integer program
– Phase II: Insertion heuristic
Integer Program

Lower bound on the total volume that has to be delivered to customer $i$ by the end of day $t$:

$$L_i^t = \max(0, tu_i - I_i^0)$$

Upper bound on the total volume that can be delivered to customer $i$ by the end of day $t$:

$$U_i^t = tu_i + C_i - I_i^0$$

Delivery constraint:

$$L_i^t \leq \sum_{1 \leq s \leq t} d_i^s \leq U_i^t \quad \forall i \in N, \ t = 1, ..., T$$
Integer Program

Resource constraints:

\[
\sum_{i : i \in r} d_{i,r}^t \leq Q x_r^t \quad \forall r \in R, \; t = 1, \ldots, T
\]

Vehicle capacity

\[
\sum_{r : r \in R} T_r x_r^t \leq |M| \quad t = 1, \ldots, T
\]

Number of vehicles
Integer Program

\[
\min \sum_t \sum_r c_r x_r^t
\]

\[
L_i^t \leq \sum_{1 \leq s \leq t} d_{i}^s \leq U_i^t \quad \forall i \in N, \ t = 1, \ldots, T,
\]

Storage capacity

Vehicle capacity

\[
\sum_{i : i \in r} d_{i}^t \leq Q x_r^t \quad \forall r \in R, \ t = 1, \ldots, T,
\]

Number of vehicles

\[
\sum_{r : r \in R} T_r x_r^t \leq |M| \quad t = 1, \ldots, T.
\]
Integer Program

Improve the efficiency by

– Route elimination
– Aggregation
Insertion Heuristic

• Input for next $k$ days:
  – List of customers
  – List of *recommended* delivery amounts

• Output for next $k$ days for each vehicle:
  – Start time
  – Sequence of deliveries
  – Arrival time at each customer
  – *Actual* delivery amount at each customer
Key Issue

• How to handle variable delivery quantities?
  – We may be able to increase delivery amounts
  – We may be able to decrease delivery amounts
  – We may be able to postpone deliveries to another day
Insertion Heuristic

Minimum delivery volume:

\[ q_{ri}^{min} = d_i \]

Amount suggested by the integer program

Earliest time a delivery can be made:

\[
t_{ri}^{early} = \max \left\{ t_{rp(i)}^{early} + t_{tp(i),i}, \frac{(q_{ri}^{min} - C_i + I_i)}{u_i} \right\}
\]
Insertion Heuristic

Latest time a delivery can be made:

\[ t_{ri}^{late} = \min \left\{ t_{rs(i)}^{late} - t_{i,s(i)} , \frac{I_i}{u_i} \right\} \]

Maximum delivery volume:

\[ q_{ri}^{max} = \min \left\{ Q - \sum_{j \neq i \in r} q_{rj}^{min} , \frac{C_i}{C_i - I_i + u_i t_{ri}^{late} - l_i u_i} \right\} \]
Insertion Heuristic

For each route:
- Earliest time a route can start
- Latest time a route can start
- Earliest time a route can end
- Latest time a route can end
- Sum of minimum deliveries
- Sum of maximum deliveries
Insertion Heuristic

Feasibility check:

– Compute minimum delivery volume. Will the minimum delivery volume fit given the other deliveries?

– Compute earliest and latest delivery can take place. Is late greater than early?

– Compute maximum delivery volume. Is minimum less than maximum?
Delivery Volume Optimization

• Observe:
  – The amount that can be delivered at a customer depends on the time at which the delivery starts
  – The time it takes to make the delivery depends on the size of the delivery
  – There is a limit on the elapsed time of a route

• Result:
  – It is nontrivial to determine, given a route, i.e., a sequence of customer visits, what the maximum amount of product is that can be delivered on this route!!
Delivery Volume Optimization

Tank capacity or Truck capacity

Usage rate

Pump rate

Earliest delivery time

Latest delivery time

THE LOGISTICS INSTITUTE
Georgia Institute of Technology
www.tli.gatech.edu
There is a polynomial time algorithm that solves this problem. The algorithm constructs a series of piecewise linear graphs (one for each customer on the route) representing the maximum amount of product that can be delivered on the remainder of the route as a function of the start time of the delivery at the customer.
Delivering a little less at Customer 1 allows a much larger delivery at Customer 2.
• The insertion heuristic is embedded in a Greedy Randomized Adaptive Search Heuristic (GRASP)
Stochastic IRP

1. Determine inventory levels
2. Assign customers to vehicles
3. Load to capacity & drive to customer
4. Deliver at customer & drive back
Markov Decision Process Model

- **State, \( x \)**
  - inventory levels at different customers

- **Action, \( a \)**
  - Which customers to replenish
  - How much to deliver at each customer
  - How to combine customers into vehicle routes

- **Objective**

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

- Algorithm: Policy Iteration
- For each problem
  - Customer capacity: 10 units
  - Customer demand: 1, ..., 10 w.p. 0.1 each
  - Vehicle capacity: 5 units

<table>
<thead>
<tr>
<th>Instance</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 customers, 1 vehicle</td>
<td>3 sec</td>
</tr>
<tr>
<td>3 customers, 2 vehicles</td>
<td>15 min</td>
</tr>
<tr>
<td>4 customers, 3 vehicles</td>
<td>24 hr</td>
</tr>
</tbody>
</table>
MDP Model: Issues

- Optimality Equation

\[ V^*(x) = \max_{a \in A(x)} E[g(x, a) + \alpha V^*(X_{t+1}) | X_t = x, A_t = a] \]

- Computing optimal value function
- Computing expected value
- Computing optimal action
Approximation Methods

• Idea

Approximate $V^*$ with $\hat{V}$

• Motivation

$$\text{if } \|V^* - \hat{V}\| \leq \varepsilon \text{ and }$$

$$\hat{\pi}(x) \in \arg \max_{a \in A(x)} E \left[ g(x,a) + \alpha \hat{V}(X_1) \mid X_0 = x \right]$$

then

$$\|V^* - V^{\hat{\pi}}\| \leq \frac{2\alpha\varepsilon}{1 - \alpha}$$

• Parameterized approximation function

$$\hat{V}(x, \beta) = r_0 + r_1\phi_1(x) + \cdots + r_n\phi_n(x)$$
Examples for Basis Functions

• Polynomial function
  – inventory level at customers
  – second order effects
Approximating the Value Function

- Although IRP is not separable, the major costs (including transportation) are associated with small groups of customers (vehicle routes)
- We do not know in advance which groups will be in each vehicle route
- We can identify subsets of customers that can possibly be in the same vehicle route
MDP for subset of customers

- State, \((x_i, k_i)\)
  - inventory at customers \(\times\) vehicles which could be allocated
- Action, \(a_i\)
  - deliveries to customers in the subset
- Transition probability

\[
q_i(y_i, k_i | x_i, d_i) = f_i(x_i + d_i - y_i) p_i(k_i | y_i)
\]

MDPs for small subsets of customers can be solved optimally in advance
Approximating the Value Function

\[ \bar{V}(x, r) = r_0 + \sum_{i=1}^{n} r_i V_i^*(x_i, k_i^*) \]

- In advance, optimally solve problems for subsets of customers
- On each day, partition the customers and vehicles into subsets by solving a cardinality constrained partitioning problem
- 1-customer subsets: nonlinear knapsack problem
- 2-customer subsets: maximum weight perfect matching problem
Non-linear Knapsack Problem

\[
\begin{align*}
\text{Max} & \quad \sum_{i=1}^{n} V_{i}^{*}(x_{i}, k_{i}) \\
\text{subject to} & \quad \sum_{i=1}^{n} k_{i} \leq K 
\end{align*}
\]
Computing Parameters
Method I

- Objective Function

\[ \min_{\pi} \sum_{x \in X} \nu^{\pi t}(x) \left[ V^{\pi t}(x) - \bar{V}(x, r) \right]^2 \]

- Looks like weighted least squares regression problem

Cannot be computed for large problems
Computing Parameters
Stochastic Approximation Algorithm

- Simulate system under policy $\pi$
- Sample path $x^0, x^1, \ldots, x^{t}, \ldots$
- Update coefficients $r^{t+1} = r^t + \gamma^t \delta^t z^t$
- Step size $\sum_{t=0}^{\infty} \gamma^t = \infty$, $\sum_{t=0}^{\infty} (\gamma^t)^2 < \infty$
- Temporal difference $\delta^t = g(x^t, \pi(x^t)) + \alpha \hat{V}(x^{t+1}, r^t) - \hat{V}(x^t, r^t)$
- Eligibility vector $z^{t+1} = \alpha \lambda z^t + \nabla_r \hat{V}(x^t, r^t)$

Convergence typically very slow
Computing Parameters

Method II

- Value function for policy $\pi$

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

Cannot be computed for large problems

$$\min_{\pi} \sum_{x \in \mathcal{X}} \nu^\pi(x) \left[ \bar{V}(x, r) - \left( g(x, \pi(x)) + \alpha \sum_{y \in \mathcal{X}} P[y \mid x, \pi(x)] \bar{V}(y, r) \right) \right]^2$$

- Looks like weighted least squares regression problem
Computing Parameters
Kalman Filter Algorithm

• Simulate system under policy $\pi$
• Sample path $x^0, x^1, ..., x^t, ...$
• Update matrices $M^t$ (similar to $X'X$) & $Y^t$ (similar to $X'Y$)
• $r^t$ is the solution of $M^t r^t = Y^t$

Convergence significantly faster
Computing Parameters
Stoch App vs. Kalman F + Stoch App

Time (hours)

Parameters

Customer 1
Customer 2
Customer 1
Customer 2
Stochastic App
Kalman F
Estimating Expected Value

- Multi-dimensional Integral
  - $d = \#\text{dimensions} = \#\text{customers}$
  - very hard to compute
- Deterministic Methods
  - $\text{MSE} = O(n^2 - 2c/d)$
- Randomized Methods
  - $\text{MSE} = O(1/n)$
- Deterministic methods are better when $2 - 2c/d < -1$

Randomized methods are better for large $d$
Choosing the Best Action

- Based on sample averages of actions
- Question
  - How large should the sample be so that we are reasonably sure of choosing the best action?
  - Sample size to ensure chosen alternative has value within tolerance of best value with specified probability
- Variance reduction methods
  - Common random numbers
  - Orthogonal arrays
Variance Reduction Methods

Number of Observations for Choosing Best Action

Simulation Steps

Number of Observations

OA
Random
Approximate Policy Iteration

1. Initialization. Simulate initial policy $\pi_0$ and obtain parameters $r^{\pi_0}$

2. Use parameters $r^{\pi_{t-1}}$ for policy $\pi_t$ and obtain actions using

$$\pi_t(x) \in \arg \max_{a \in A(x)} \left\{ g(x, a) + \alpha \sum_{y \in X} P[y \mid x, a] \bar{V}(y, r^{\pi_{t-1}}) \right\}$$

3. Simulate policy $\pi_t$ and obtain parameters $r^{\pi_t}$

4. $t \leftarrow t + 1$; go to Step 2
Performance Comparison
Small Instances

<table>
<thead>
<tr>
<th>Instance</th>
<th>Max percentage difference</th>
<th>π₁</th>
<th>π₂</th>
<th>π₃</th>
<th>π_{CBW}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>0.18%</td>
<td>0.002%</td>
<td>0.002%</td>
<td>0.54%</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1.93%</td>
<td>0.22%</td>
<td>0.15%</td>
<td>5.22%</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1.75%</td>
<td>0.009%</td>
<td>0.009%</td>
<td>10.02%</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.84%</td>
<td>0.15%</td>
<td>0.10%</td>
<td>16.36%</td>
</tr>
</tbody>
</table>

Inventory at customer 3 (X₃)

Value *10^{-3}

Inventory at customer 3 (X₃)
## Performance Comparison
### Large Instances

<table>
<thead>
<tr>
<th>P</th>
<th>State</th>
<th>$\pi_1$</th>
<th>$\pi_{CBW}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\mu$</td>
<td>$\mu - 2\sigma$</td>
</tr>
<tr>
<td>1</td>
<td>$x_1$</td>
<td>400.6</td>
<td>386.7</td>
</tr>
<tr>
<td></td>
<td>$x_2$</td>
<td>401.7</td>
<td>388.0</td>
</tr>
<tr>
<td></td>
<td>$x_3$</td>
<td>404.1</td>
<td>392.2</td>
</tr>
<tr>
<td></td>
<td>$x_4$</td>
<td>403.9</td>
<td>390.2</td>
</tr>
<tr>
<td>2</td>
<td>$x_1$</td>
<td>892.0</td>
<td>887.5</td>
</tr>
<tr>
<td></td>
<td>$x_2$</td>
<td>886.0</td>
<td>871.5</td>
</tr>
<tr>
<td></td>
<td>$x_3$</td>
<td>896.4</td>
<td>892.0</td>
</tr>
<tr>
<td></td>
<td>$x_4$</td>
<td>891.2</td>
<td>881.9</td>
</tr>
</tbody>
</table>
Price-Direct Replenishment

- A control policy based on a simple economic mechanism for dispatching.
- The dispatcher receives a transfer price $d_i V_i$ from management for replenishing $d_i$ units of product at customer $i$.
- The dispatcher is responsible for paying the distribution costs $c_i$, when replenishing a set of customers $I$. 
Price-Direct Replenishment

- Net value for dispatcher
  \[ \sum_{i \in I} V_i d_i - c_I \]
- Incremental value for dispatcher
  \[ d_i V_i - (c_{I \cup \{i\}} c_I) \]
Price-Directed Replenishment

- Management’s problem: Set $V_i$ so that the dispatcher is motivated to minimize the long-run time average replenishment costs
Price-Direct Replenishment

• Management problem (single customer):

Primal

\[ \min cZ \]
\[ dZ = u \]
\[ 0 \leq d \leq \min\{C, Q\} \]
\[ 0 \leq Z \]

Dual

\[ \max uV \]
\[ dV \leq c \quad \forall 0 \leq d \leq \min\{C, Q\} \]
Price-Direct Replenishment

• Management problem (single customer):

If $V$ is interpreted as the transfer price received by the dispatcher for replenishing one unit, then this dual program maximizes the rate at which transfer revenue accumulates, subject to the constraint that the total transfer payment cannot exceed the cost on any replenishment.

Dual

\[
\begin{align*}
\max & \quad uV \\
\text{s.t.} & \quad dV \leq c, \quad \forall 0 \leq d \leq \min\{C, Q\}
\end{align*}
\]
Direct Replenishment

• Price directed operating policy maximizing the net value of a replenishment

\[ \max_{0 \leq d \leq \min\{C, Q\}} \{ V^* d - c \} \]
References

• L. Bertazzi, G. Palletta, M.G. Speranza (2005). Minimizing the total cost in an integrated vendor-managed inventory system. JH 11, 393-419.
References

Inventory Routing Game

- http://kronos.isye.gatech.edu:8081/IRGame
- Login: player1, …, player20
- Password: player1, …, player20
- Play Instance 3

- Winner gets prize on Friday…
Questions?