VENDOR MANAGED INVENTORY

Martin Savelsbergh
School of Industrial and Systems Engineering
Georgia Institute of Technology

Joint work with
Ann Campbell, Anton Kleywegt, and Vijay Nori

Distribution Systems: Location and Vehicle Routing
Montreal
December, 2002
Outline

- Vendor Managed Inventory Replenishment
- Vendor Managed Inventory at Praxair
- Inventory Routing Problem
- Performance Measures
- Solution Approaches
  - Deterministic Inventory Routing Problem
  - Stochastic Inventory Routing Problem
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
Conventional Inventory Management -- Day 1
Conventional Inventory Management -- Day 2
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 -- Day 1

MICHIGAN

Detroit

LAKE ERIE

Cleveland

OHIO
Vendor Managed Inventory -- Day 2
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
VMI Essentials

- TRUST
  - Accurate information provided on a timely basis
  - Inventory levels that meet demands
  - Confidential information kept confidential

- TECHNOLOGY
  - Automated electronic messaging systems to exchange sales and demand data, shipping schedules, and invoicing
Applications of VMI

- Industrial Gases
  - air products distribution
  - carbon black distribution
- Petrochemical industry
  - gas stations
- Automotive Industry
  - parts distribution
- Beverage Industry
  - Vending machines
Praxair’s Business

- Plants worldwide
  - 44 countries
  - USA 70 plants
  - South America 20 plants

- Product classes
  - packaged products
  - bulk products
  - lease manufacturing equipment

- Distribution
  - 1/3 of total cost attributed to distribution
Praxair’s Business

Bulk products

- Distribution
  - 750 tanker trucks
  - 100 rail cars
  - 1,100 drivers
  - drive 80 million miles per year

- Customers
  - 45,000 deliveries/month to 10,000 customers

- Variation
  - 4 deliveries/customer/day to
  - 1 delivery/customer/2 months

- Routing varies from day to day
VMI Implementation at Praxair

- Convince management and employees of new methods of doing business
- Convince customers to trust vendor to do inventory management
- Pressure on vendor to perform - Trust easily shaken
- Praxair currently manages 80% of bulk customers’ inventories
- Demonstrate benefits
VMI Implementation at Praxair

- Praxair receives inventory level data via
  - telephone calls: 1,000 per day
  - fax: 500 per day
  - remote telemetry units: 5,000 per day
- Forecast customer demands based on
  - historical data
  - customer production schedules
  - customer exceptional use events
- Logistics planners use decision support tools to plan
  - whom to deliver to
  - when to deliver
  - how to combine deliveries into routes
  - how to combine routes into driver schedules
Benefits of VMI at Praxair

- Before VMI, 96% of stockouts due to customers calling when tank was already empty or nearly empty.
- VMI reduced customer stockouts.
What’s needed to make VMI work

- Information management is crucial to the success of VMI
  - inventory level data
  - historical usage data
  - planned usage schedules
  - planned and unplanned exceptional usage
- Forecast future demand
- Decision making: need to decide on a regular (daily) basis
  - whom to deliver to
  - when to deliver
  - How much to deliver
  - how to combine deliveries into routes
  - how to combine routes into driver schedules
Inventory Routing Problem

- Single Vendor
  - single facility
  - single product
  - set of n customers
  - set of m homogenous vehicles of capacity Q
Inventory Routing Problem

Cost Elements

- Revenue
- Transportation cost
- Shortage cost
- Inventory holding cost
Inventory Routing Problem

- Each customer:
  - storage capacity
  - initial inventory
  - average product usage 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)
Performance Measures

- $/mile
- $/volume
- volume/mile
- stock-outs/deliveries
- avg. inventory before delivery
- avg. inventory after delivery
Performance – Tour efficiency

Vehicle capacity 500

Capacity 500 deliver 500

100 miles

B

Capacity 1000 deliver 500

100 miles

A

Plant

10 miles

A

Capacity 1000 deliver 500

10 miles

Plant
Performance – Tour efficiency

Vehicle capacity 500

plant

A

capacity 100
deliver 100

B

capacity 400
deliver 400

100 miles

10 miles

plant

A

capacity 400
deliver 400

B

capacity 100
deliver 100

100 miles

100 miles

100 miles

100 miles

100 miles
Performance – Tour efficiency

- **Plant**
  - Capacity: 400
  - Deliver: 400

- **B**
  - 10 miles
  - Capacity: 400
  - Deliver: 400

- **C**
  - 100 miles
  - Capacity: 100
  - Deliver: 100

- **D**
  - 10 miles
  - Capacity: 400
  - Deliver: 400

- **A**
  - 10 miles
  - Capacity: 100
  - Deliver: 100

- **Vehicle capacity: 500**
Vehicle capacity 500

- Plant A to B: 100 miles
  - Capacity 400
  - Deliver 400

- Plant B to A: 100 miles
  - Capacity 400
  - Deliver 100

- Plant to A: 10 miles
  - Deliver 100

- Plant to B: 10 miles
  - Deliver 400
Performance – Tour efficiency

Vehicle capacity 500

Capacity 400
Deliver 400

10 miles

Capacity 600
Deliver 400

10 miles

Capacity 400
Deliver 100

10 miles

Capacity 600
Deliver 100

10 miles

Plant
Performance – Tour efficiency

- Proposed efficiency measures

  - Miles: $D$

  - Volume per mile: $\sum_i \frac{v_i}{D}$

  - Weighted volume per mile:

    \[
    \frac{\sum_i v_i}{Q} \sum_i \frac{v_i}{\min\{C_i, Q\}} \frac{2d_i}{D}
    \]

    | Truck Utilization | Importance Level | Relative Distance |
Solution Approaches

- Deterministic
  - Based on average product usage

- Stochastic
  - Based on probability distribution of product usage
Deterministic Solution Approach

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
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:

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

Resource constraints:

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

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

\[
\begin{align*}
\min & \sum_t \sum_r c_r x_r^t \\
L_i^t & \leq \sum_{1 \leq s \leq t} d_{i,s}^t \leq U_i^t \quad \forall i \in N, \ t = 1, \ldots, T, \\
\sum_{i : i \in r} d_{i,r}^t & \leq Q x_r^t \quad \forall r \in R, \ t = 1, \ldots, T, \\
\sum_{r : r \in R} T_r x_r^t & \leq |M| \quad t = 1, \ldots, T.
\end{align*}
\]
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

- Feasibility
  - Minimum delivery quantity
    \[ q_{ri}^{\text{min}} = d_i \]
  - Earliest time a delivery can be made
    \[ t_{ri}^{\text{early}} = \max \left\{ t_{rp(i)}^{\text{early}} + tt_{p(i),i}, \left( q_{ri}^{\text{min}} - C_i + I_i \right)/w_i \right\} \]
Insertion Heuristic

- Feasibility (cont.)
  - Latest time a delivery can be made

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

- Maximum delivery quantity

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

- Profitability
  - The maximum volume deliverable on a route
  - The change in maximum volume deliverable on a route
  - *Cannot be computed in constant time !!!*
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

- 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.
Delivery Volume Optimization

Tank capacity or Truck capacity

Usage rate

Pump rate

Earliest delivery time

Latest delivery time
Delivery Volume Optimization

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)
Instance Characteristics

Production Facility A
Instance Characteristics

Production Facility B
Instance Characteristics

Deliveries Per Week for Plant A Customers

Number of deliveries

Customer
Instance Characteristics

Deliveries Per Week for Plant B Customers

Diagram showing the relationship between customer number and number of deliveries per week.
Instance Characteristics

Tank Sizes for Plant A

- Tank sizes range from 0 to 90,000.
- Customers are numbered from 1 to 46.

The graph shows an increasing trend in tank sizes for each customer.
Instance Characteristics

Tank Sizes for Plant B

[Graph showing tank sizes for Plant B with customer numbers and tank sizes labeled.]
Computational Experience

Comparison of optimization approach and industry approach:

<table>
<thead>
<tr>
<th>Approach</th>
<th>V</th>
<th>T</th>
<th>S</th>
<th>U</th>
<th>Vol</th>
<th>Mile</th>
<th>V/M</th>
<th>aV/M</th>
<th>wV/M</th>
<th>Bef</th>
<th>Aft</th>
</tr>
</thead>
<tbody>
<tr>
<td>O 1</td>
<td>3</td>
<td>85</td>
<td>118</td>
<td>95.72</td>
<td>2613027</td>
<td>18841</td>
<td>128.09</td>
<td>410</td>
<td>18858</td>
<td>24.54</td>
<td>81.04</td>
</tr>
<tr>
<td>I 2</td>
<td>4</td>
<td>67</td>
<td>90</td>
<td>89.58</td>
<td>2519989</td>
<td>18988</td>
<td>132.71</td>
<td>309</td>
<td>18357</td>
<td>11.41</td>
<td>81.27</td>
</tr>
<tr>
<td>O 3</td>
<td>3</td>
<td>81</td>
<td>106</td>
<td>90.26</td>
<td>30283480</td>
<td>14226</td>
<td>2128.74</td>
<td>8013</td>
<td>215035</td>
<td>15.91</td>
<td>92.43</td>
</tr>
<tr>
<td>I 4</td>
<td>3</td>
<td>81</td>
<td>93</td>
<td>85.41</td>
<td>28856473</td>
<td>15042</td>
<td>1905.10</td>
<td>7018</td>
<td>208276</td>
<td>9.83</td>
<td>88.37</td>
</tr>
</tbody>
</table>

Value of delivery amount optimization:

<table>
<thead>
<tr>
<th>Setting</th>
<th>V</th>
<th>T</th>
<th>S</th>
<th>U</th>
<th>Vol</th>
<th>Mile</th>
<th>V/M</th>
<th>aV/M</th>
<th>wV/M</th>
<th>Bef</th>
<th>Aft</th>
</tr>
</thead>
<tbody>
<tr>
<td>N 5</td>
<td>3</td>
<td>83</td>
<td>126</td>
<td>87.56</td>
<td>2537554</td>
<td>19529</td>
<td>127.33</td>
<td>330</td>
<td>17077</td>
<td>24.90</td>
<td>78.00</td>
</tr>
<tr>
<td>Y 6</td>
<td>3</td>
<td>85</td>
<td>118</td>
<td>95.72</td>
<td>2613027</td>
<td>18441</td>
<td>138.89</td>
<td>410</td>
<td>18858</td>
<td>24.54</td>
<td>81.64</td>
</tr>
<tr>
<td>N 7</td>
<td>3</td>
<td>81</td>
<td>111</td>
<td>85.81</td>
<td>28789623</td>
<td>15274</td>
<td>1884.88</td>
<td>6012</td>
<td>202435</td>
<td>15.12</td>
<td>85.10</td>
</tr>
<tr>
<td>Y 8</td>
<td>3</td>
<td>81</td>
<td>106</td>
<td>90.26</td>
<td>30283480</td>
<td>14226</td>
<td>2128.74</td>
<td>8013</td>
<td>215035</td>
<td>15.51</td>
<td>92.49</td>
</tr>
</tbody>
</table>
**Computational Experience**

Value of randomization:

<table>
<thead>
<tr>
<th>Setting</th>
<th>V</th>
<th>T</th>
<th>S</th>
<th>U</th>
<th>Vol</th>
<th>Mile</th>
<th>V/M</th>
<th>aV/M</th>
<th>wV/M</th>
<th>Bef</th>
<th>Aft</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
<td>67</td>
<td>116</td>
<td>55.70</td>
<td>2838700</td>
<td>19315</td>
<td>136.51</td>
<td>419</td>
<td>18559</td>
<td>24.78</td>
<td>82.77</td>
</tr>
<tr>
<td>25</td>
<td>3</td>
<td>65</td>
<td>118</td>
<td>55.72</td>
<td>2813027</td>
<td>18441</td>
<td>138.69</td>
<td>410</td>
<td>18856</td>
<td>24.54</td>
<td>81.64</td>
</tr>
<tr>
<td>50</td>
<td>4</td>
<td>65</td>
<td>121</td>
<td>55.22</td>
<td>2544803</td>
<td>19170</td>
<td>132.71</td>
<td>327</td>
<td>17793</td>
<td>23.74</td>
<td>77.21</td>
</tr>
<tr>
<td>NO</td>
<td>3</td>
<td>59</td>
<td>95</td>
<td>56.22</td>
<td>27979401</td>
<td>14378</td>
<td>1945.99</td>
<td>6399</td>
<td>200752</td>
<td>15.28</td>
<td>88.34</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>62</td>
<td>111</td>
<td>57.99</td>
<td>30004093</td>
<td>15094</td>
<td>1987.86</td>
<td>6773</td>
<td>201000</td>
<td>19.88</td>
<td>89.13</td>
</tr>
<tr>
<td>25</td>
<td>3</td>
<td>61</td>
<td>106</td>
<td>50.28</td>
<td>30263480</td>
<td>14220</td>
<td>2128.74</td>
<td>8019</td>
<td>215095</td>
<td>19.91</td>
<td>92.49</td>
</tr>
<tr>
<td>50</td>
<td>3</td>
<td>63</td>
<td>111</td>
<td>57.57</td>
<td>30343074</td>
<td>14805</td>
<td>2077.58</td>
<td>7762</td>
<td>211049</td>
<td>19.88</td>
<td>89.61</td>
</tr>
</tbody>
</table>
Real Life Issues

- Drivers:
  - Fixed start times/Variable start times
  - DOT regulations:
    - 10 hours maximum drive time per day
    - 15 hours maximum work time per day
    - 70 hours maximum work time in any 8 day window
  - Sleeper teams
- Order only customers
- Exceptional use periods
- Multi-facility
Stochastic IRP

If storage capacities and customer demands are sufficiently large relative to vehicle capacity, it is often optimal to deliver full or nearly full truck loads to customers.
DDIRP: Sequence of Events

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

- Discrete time $t = 0, 1, 2, \ldots$
- State $x$
  - inventory level at each customer
- Action $a$
  - amount to deliver at each customer, which determines number of vehicles to be sent to each customer
- Single stage reward $g(x, a)$
- Objective

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

- **Optimality Equation**
  
  \[ V^*(x) = \max_{a \in A(x)} E \left[ g(x, a) + \alpha V^*(X_{t+1}) \mid X_t = x, A_t = a \right] \]

- **Computational effort**
  - Computing optimal value function \( V^* \)
  - Computing expected value \( E[.\] \)
  - Computing optimal action

  \[ a^*(x) \in \arg \max_{a \in A(x)} \{ \cdot \} \]
Solving Problems Exactly

- **Algorithms**
  - Value iteration (successive approximation)
  - Policy iteration
  - Modified policy iteration

- **Dilemma**
  - Computational effort proportional to number of states
  - Problems with $10,000 - 1,000,000$ states can be solved exactly
  - Number of states grows exponentially with number of customers
    \[ \#\text{states} = (\#\text{inv levels})^{\#\text{cust}} \]
  - With 10 inventory levels per customer, problems with 4 – 6 customers can be solved
  - Typical plant serves 20 – 100 bulk customers
Solving Problems Exactly

- Algorithm: Modified 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>
Approximation Methods

- Idea

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

- Motivation

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

$$\hat{\pi}(x) \in \arg \max_{a \in A(x)} E \left[ g(x, a) + \alpha \hat{V}(X_{t+1}) \mid X_t = x, A_t = a \right]$$

then

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

- single stage costs are separable
  \[ g(x, a) = \sum_{i} g_i(x_i, a_i) \]
- IRP is not separable

\[
\max_{k_i} \sum_{i} V_i^*(x_i, k_i) \\
\text{s.t. } \sum_{i} k_i \leq K
\]

Optimal value of a single customer MDP
Single Customer MDP

- **State** \((x_i, k_i)\)
  - inventory at customer
  - number of vehicles allocated
- **Action** \(a_i\)
  - number of vehicles dispatched
  - amount of product delivered
- **Transition probability**

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

Single customer MDPs can be optimally solved very quickly using modified policy iteration
Nonlinear Knapsack Problem

\[
\text{Max} \quad \sum_{i=1}^{n} V_{i}(x, k) \\
\text{subject to} \quad \sum_{i=1}^{n} k_i \leq K
\]
Parametric Value Function Approximation

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

- \( V_i^*(x_i, k_i^*) \): optimal value from single customer SDPs
- \( r_i \): parameter associated with customer

- Optimally solve single customer problems
- Select appropriate \( k_i^* \) for \( V_i^*(x_i, k_i^*) \) by solving a nonlinear knapsack problem
Computing Parameters
Method I

- **Objective Function**

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

- 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
  \[ \delta^t = g(x^t, \pi(x^t)) + \alpha \bar{V}(x^{t+1}, r^t) - \bar{V}(x^t, r^t) \]
- Eligibility vector
  \[ z^{t+1} = \alpha \lambda z^t + \nabla_r \bar{V}(x^t, r^t) \]

Convergence typically very slow
**Computing Parameters**

**Method II**

- Value function for policy $\pi$

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

- Looks like weighted least squares regression problem

Cannot be computed for large problems

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

- Simulate system under policy $\pi$
- Sample path $x^0, x^1, \ldots, x^t, \ldots$
- 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

Parameters vs. Time (hours)
Estimating Expected Value

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

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 $\delta$ of best value with probability $1 - \alpha$
- 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. Simulate policy $\pi^0$ to estimate $p_i^0(k_i | y_i)$
2. Set $t \leftarrow 0$
3. Solve subproblems to compute $V^*_i(x_i, k_i)$
4. Simulate policy $\pi^t$ to estimate $p_i^t(k_i | y_i)$ and $\beta^{\pi^t}$
5. Use $\beta^{\pi^t}$ to define $\pi^{t+1}$

$$\pi^{t+1}(x) \in \arg \max_{a \in A(x)} \left\{ g(x, a) + \alpha \sum_{y \in \mathcal{X}} P[y | x, a] \hat{V}(y, \beta^{\pi^t}) \right\}$$

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

<table>
<thead>
<tr>
<th>Instance</th>
<th>Max percentage difference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\pi_1$</td>
</tr>
<tr>
<td>1</td>
<td>0.18%</td>
</tr>
<tr>
<td>2</td>
<td>1.93%</td>
</tr>
<tr>
<td>3</td>
<td>1.75%</td>
</tr>
<tr>
<td>4</td>
<td>0.84%</td>
</tr>
</tbody>
</table>

The graph shows the value *10^3 against the inventory at customer 3 ($X_3$) for different instances and policies, with the optimal value and policies 1, 2, 3, and CBW policy represented.
# 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>
Approximating the Value Function

- Decompose the problem into subproblems
  - Markov decision process involving a subset of customers
- Solve the subproblems
  - Modified policy iteration
- Combine the subproblem results using an optimization problem to produce the approximate value function
  - Cardinality constrained partitioning problem
Approximating the Value Function

- Example: subset of two customers $i$ and $j$
  - State $(x_i, x_j, v_{ij})$
  - decision $a_{ij}$
  - single stage cost $g_{ij}(x_i, x_j, a_{ij})$
  - transition probabilities $q_{ij}(y_i, y_j, w_{ij} \mid x_i, x_j, d_i(a_{ij}), d_j(a_{ij}))$

- Optimal value function: $V_{ij}^*(x_i, x_j, v_{ij})$
Approximating the Value Function

- State:
  - Inventory levels at customers
  - Vehicle availability to customers (0,1, or 2)

- A single vehicle available:
  - exclusive delivery to i
  - exclusive delivery to j
  - exclusive delivery to i and j (no deliveries to other customers)
  - fraction of vehicle capacity delivered to i and no delivery to j
  - fraction of vehicle capacity delivered to j and no delivery to i
  - fraction of vehicle capacity delivered to i and j plus delivery to other customers
Approximating the Value Function

- Two vehicles available:
  - exclusive delivery to i and j (no deliveries to other customers),
  - exclusive delivery to i, fraction of vehicle capacity delivered to j
  - exclusive delivery to j, fraction of vehicle capacity delivered to i
  - fraction of vehicle capacity delivered to i and fraction of vehicle capacity delivered to j (with different vehicles visiting i and j, each also delivering to other customers)
Approximating the Value Function

- Cardinality constrained partitioning problem

\[
\hat{V}(x) = \max_y \sum_{i \in \mathcal{N}} V_i^*(x_i, 0) y_{i0} + \sum_{S \in S} V_S^*(x_S, 1) y_{S1}
\]

\[
\text{s.t.} \quad y_{i0} + \sum_{S \in S_i} y_{S1} = 1 \quad \forall \ i \in \mathcal{N}
\]

\[
\sum_{S \in S} y_{S1} \leq M
\]

\[
y_{i0} \in \{0, 1\} \quad \forall \ i \in \mathcal{N}
\]

\[
y_{S1} \in \{0, 1\} \quad \forall \ S \in S
\]
Action Determination

- Construct an initial solution consisting only of direct deliveries
- For each existing route, rank all the customers not on the route by an initial estimate of the value of adding the customer to the route
- For each route evaluate more accurately the value of adding the customer to the route starting with the most promising and stopping when the values no longer improve. Identify the route and the customer that lead to the maximum improvement and add the customer to the route