Inventory Routing Problems

Martin Savelsbergh

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

Goals

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

Inventory Routing

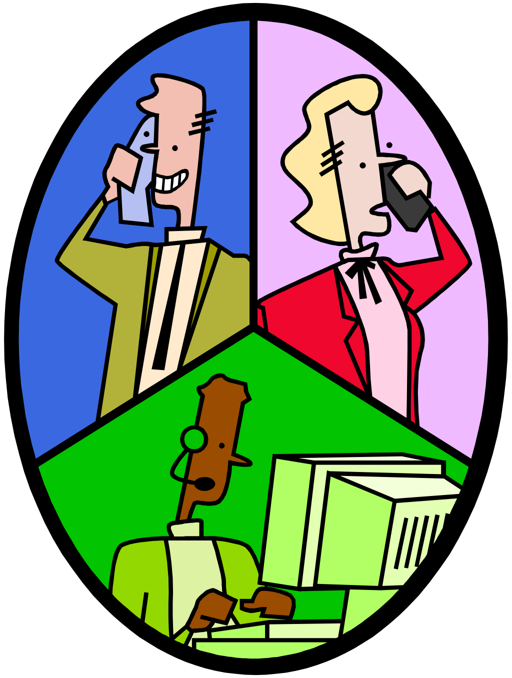
Inventory Management

↕

Vehicle Routing

Conventional Inventory Management

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



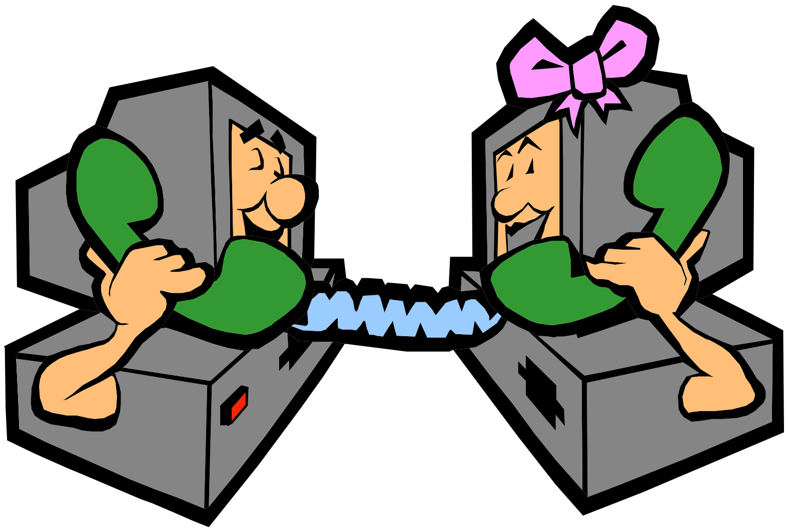
You call – We haul

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

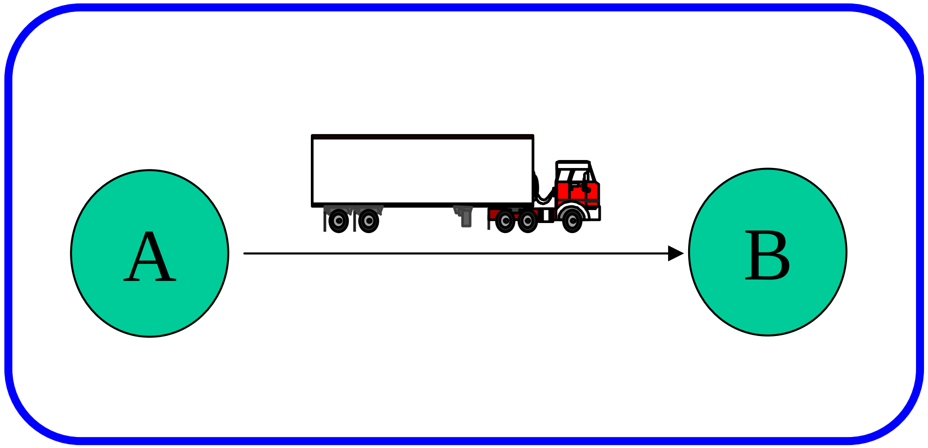
Long-Term Problem

Inventory Costs

+

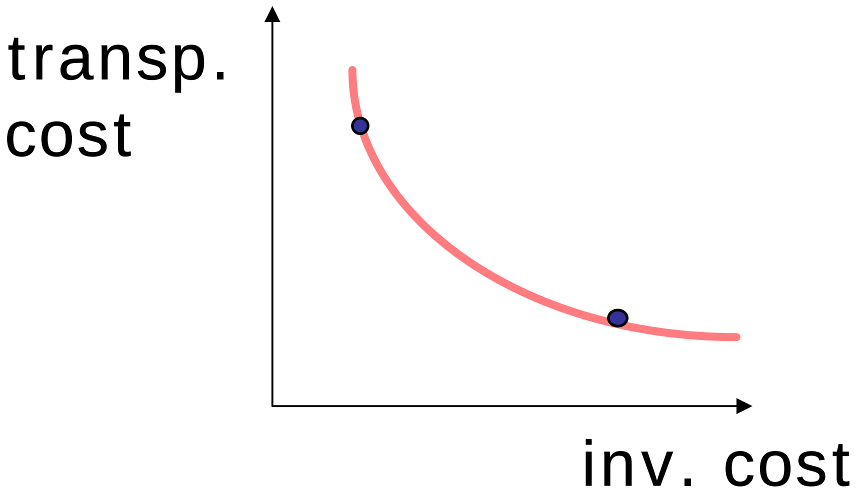
Transportation Costs

Single Customer (Single Link)

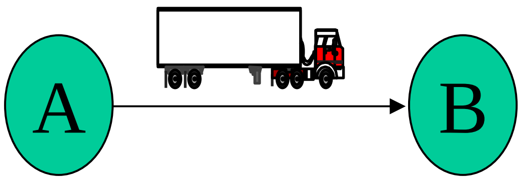


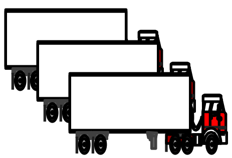
Determine shipping policies that optimize the trade-off between:

- Inventory cost
- Transportation cost



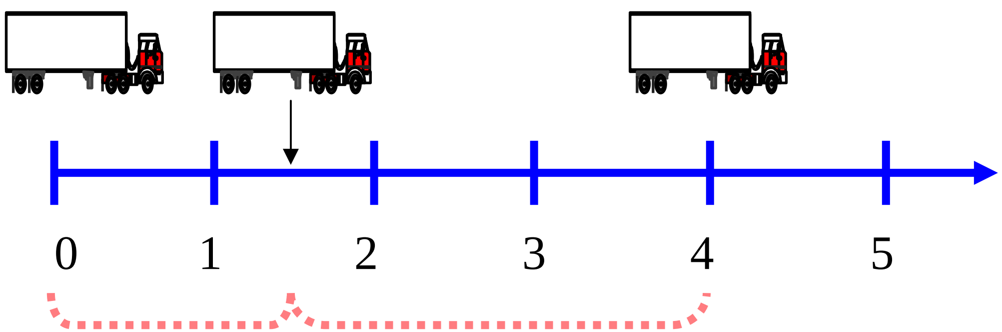
Problem Description



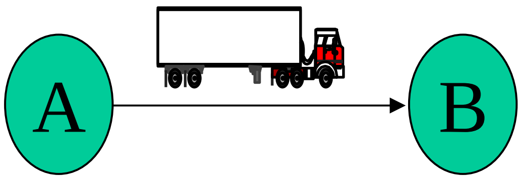


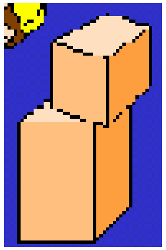
Fleet of vehicles:

- transportation capacity $r = 1$
- transportation cost c (A➔B➔A)



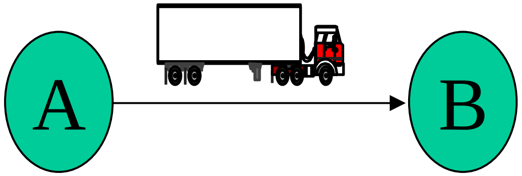
Problem Description





- 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

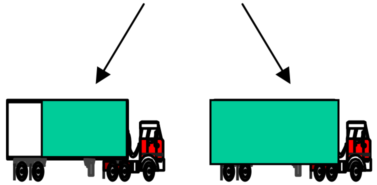
The Continuous Variant

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

$$\min hvt + \frac{c}{t}$$
$$vt \leq r$$
$$t \geq 0$$

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

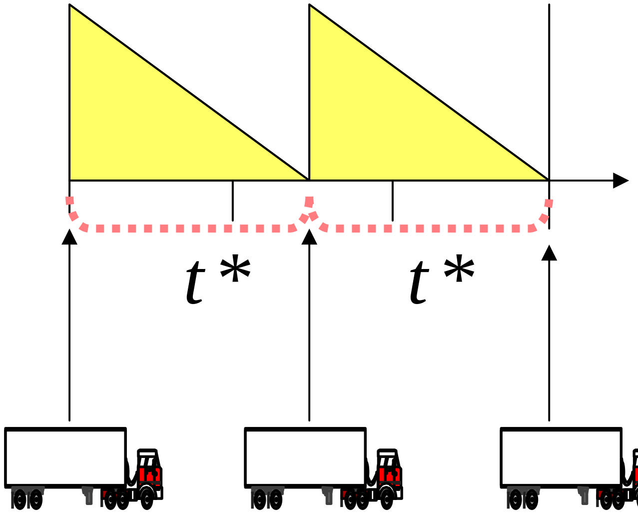
- ZIO (Zero Inventory Ordering)
- FBPS (Frequency Based Periodic Shipping)

Zero Inventory Policy

- ✓ Minimum inter-shipment time
- ✓ Single frequency f
- ✓ Continuous time between shipments t= 1/f

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

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

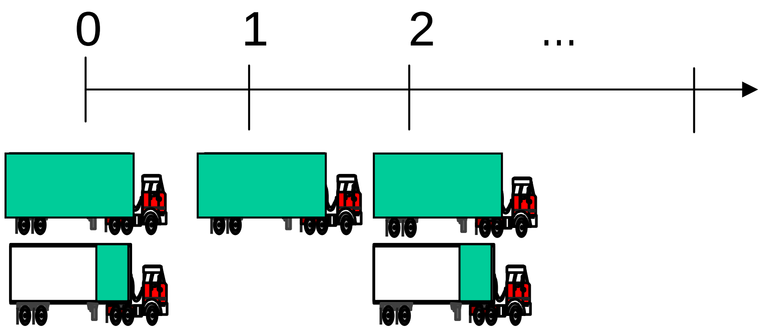


Frequency Based Periodic Shipping Policies

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

Single frequency

Double frequency



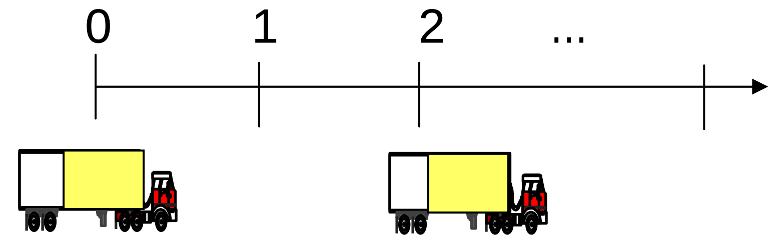
THE LOGISTICS INSTITUTE
Georgia Institute of Technology
www.tli.gatech.edu

Best Single Frequency Policy

$1/k^*$: best single frequency

Lemma: $$1 \le k^* \le \bar{k} = \max\left\{ \left\lceil (\lceil v \rceil - v)\frac{c}{h} \right\rceil, 1 \right\}$$

$$\Rightarrow \quad z^{Best\ \mathrm{SF}} = \min_{1 \le k \le \bar{k}} \left(hk + \frac{c}{k} \lceil vk \rceil \right)$$



Best Double Frequency Policy

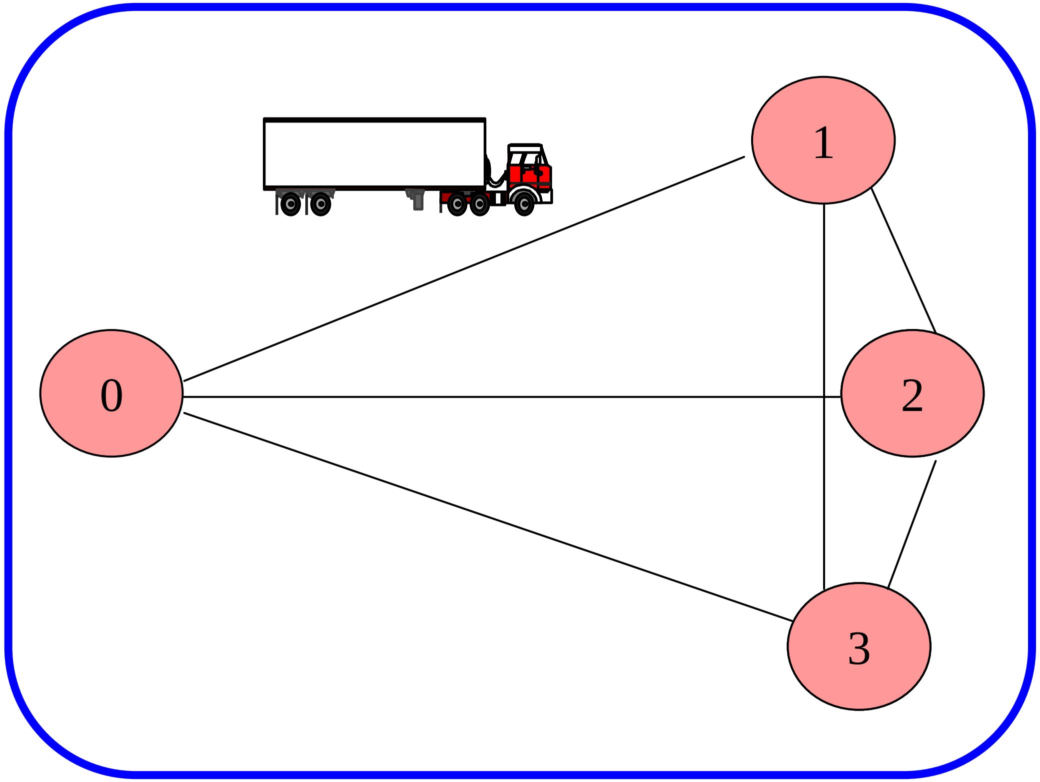
$1/k_1^*, 1/k_2^*$: best frequencies

Lemma: $1 \le k_1^* \le \bar{k}_1 = \ldots$

$k_1^* < k_2^* \le \bar{k}_2 = \ldots$

➔ $$z^{Best\ \mathrm{DF}} = \min_{1 \le k_1 \le \bar{k}_1} \left(\min \left(z^{SF}(k_1), \min_{k_1 < k_2 \le \bar{k}_2} z^{DF}(k_1, k_2) \right) \right)$$

Multiple Customers



Determine shipping policies that optimize the trade-off between:

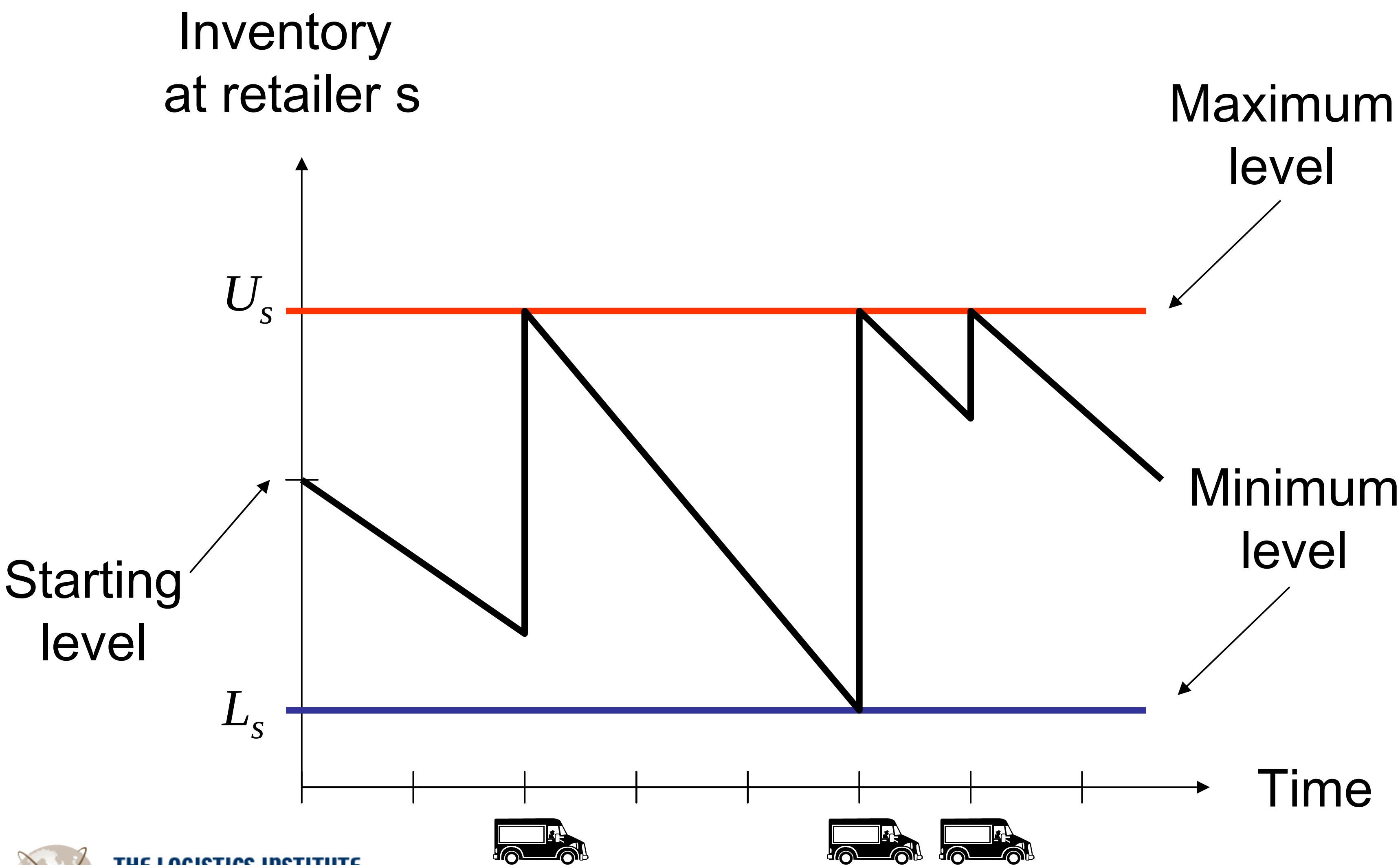
- Inventory cost
- Transportation cost

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



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:

Min   Inv. Plant + Inv. Customer + 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

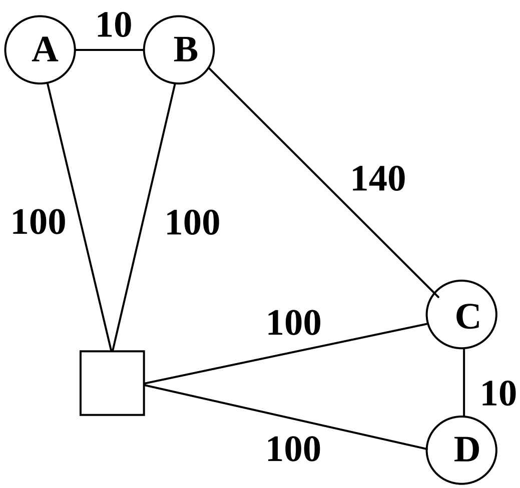
- Objective
  - Minimize distribution costs without causing any stock-outs over a finite horizon ***OR***
  - Maximize the expected total discounted value (rewards minus costs) over an infinite horizon

Inventory Routing Problem

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

Inventory Routing

Even simple situations are non-trivial



	A	B	C	D
Daily Use	1000	3000	2000	1500
Max. Delv.	5000	3000	2000	4000

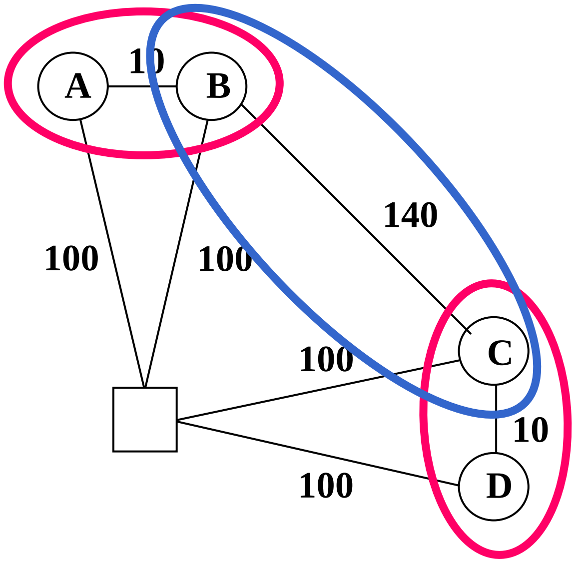
Truck capacity is 5000

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

Inventory Routing



	A	B	C	D
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

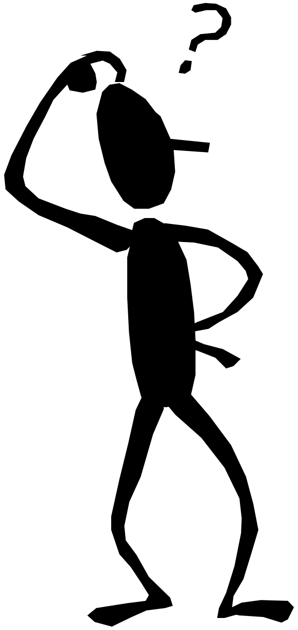
Day 1 schedule
trip 1: deliver 3000 to B & 2000 to C

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?



Single Customer Problem

Deterministic *d*-day policy

$$v_T(d) = \max\left(0, \left\lceil \frac{Tu - I}{\min(C, Q)} \right\rceil\right) c$$

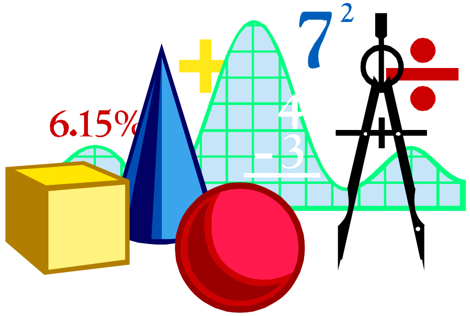
Planning horizon

Usage rate

Initial inventory

Storage capacity

Vehicle capacity



Single Customer Problem

Stochastic *d*-day policy

Probability that stockout occurs on day *j*

Stockout cost

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

Probability that no stockout occurs

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

Delivery cost

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

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

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

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

Single Customer Problem

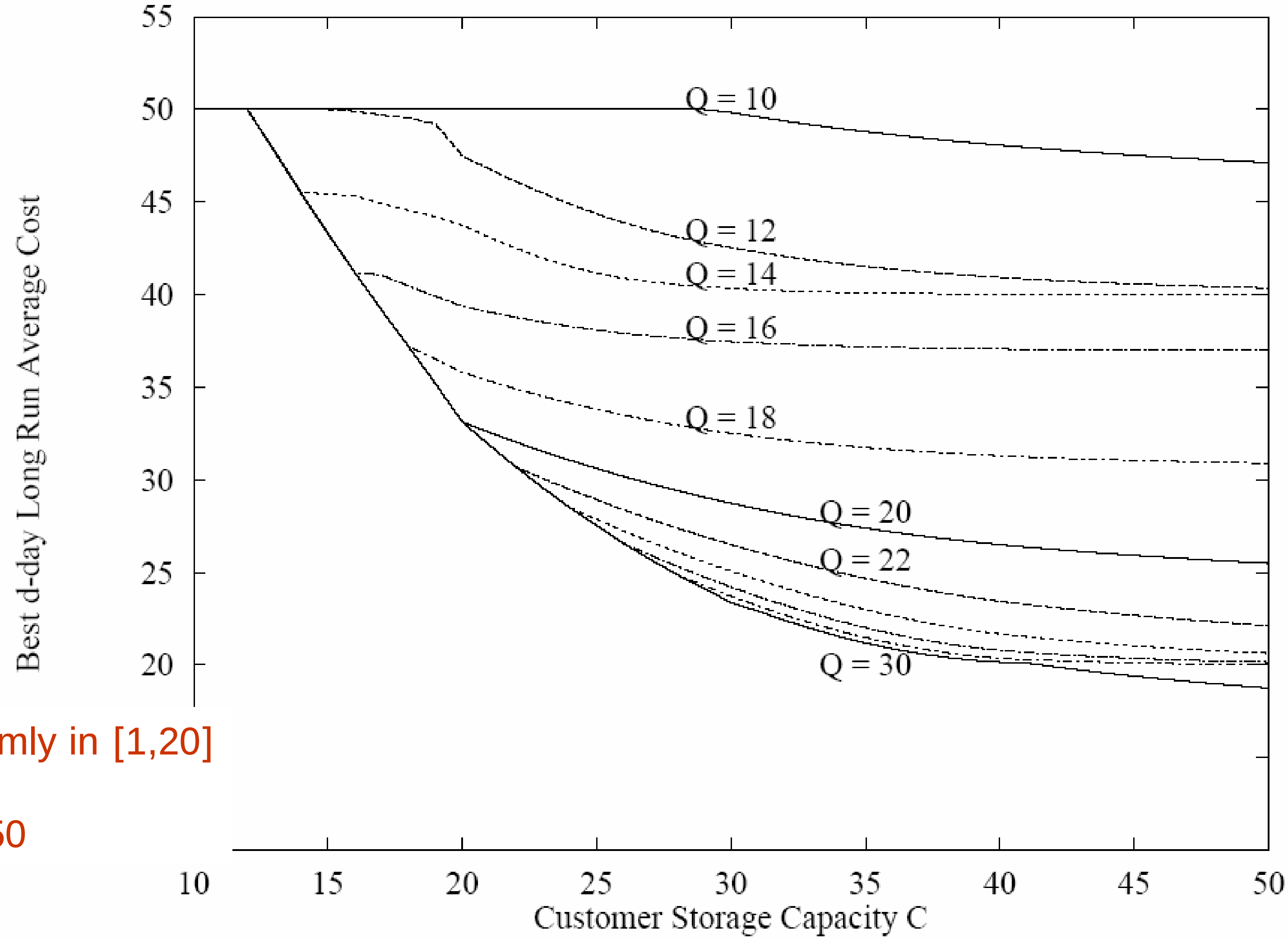
An optimal constant replenishment period strategy over a large *T*-day planning period will correspond to choosing d^* to minimize $\beta(d)$

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

Expected cost of a delivery

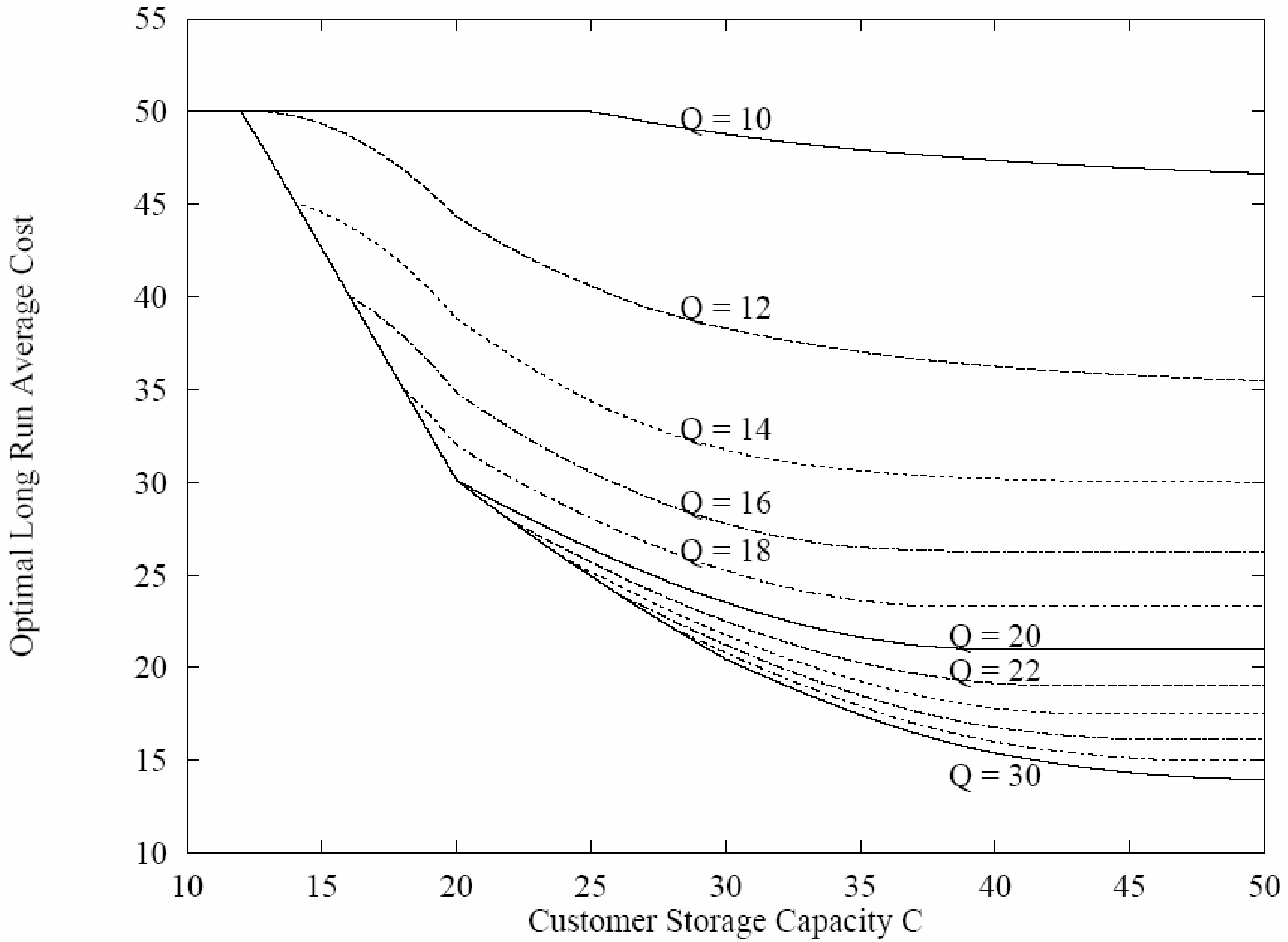
Expected number of days between deliveries

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:

$$v_T = \max(0, \lceil \frac{T u_1}{\min(C_1, Q)} \rceil) c_1 + \max(0, \lceil \frac{T u_2}{\min(C_2, Q)} \rceil) c_2$$

Always together:

$$v_T = \lceil \frac{T}{\min(\frac{C_1}{u_1}, \frac{C_2}{u_2}, \frac{Q}{u_1 + u_2})} \rceil c_{12}$$

Sometimes individually, sometimes together?

What if one cannot take a full truckload?

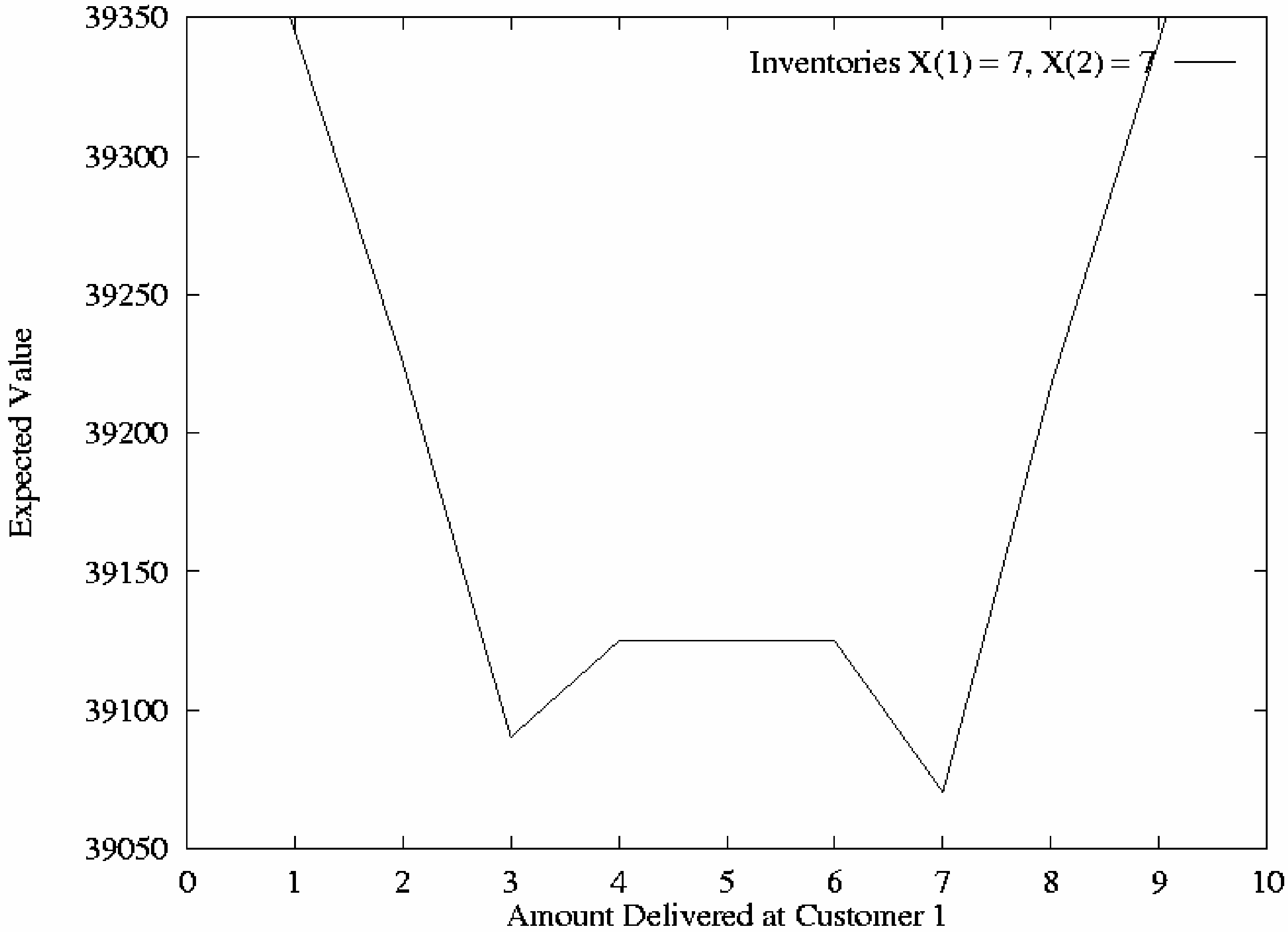
What if the customers are close together?

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

Two Customer Problem

- Stochastic policy:
  - Storage capacity: 20
  - P[demand = 0] = 0.4, P[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



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 $C_i \geq Q$**

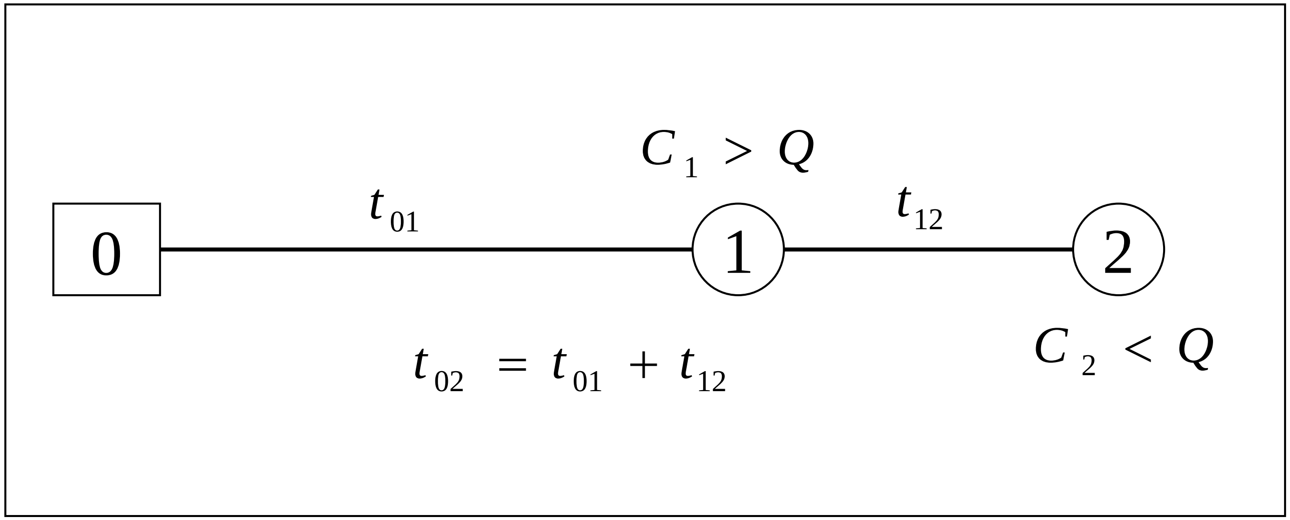
$$LB_1: \sum_{i \in I} \frac{u_i}{Q} 2t_{0i}$$

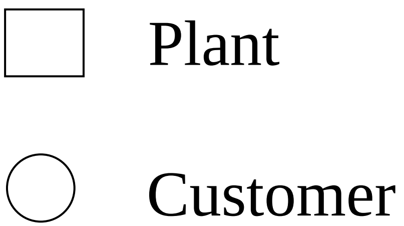
  - **Assume direct delivery**

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

I	Set of customers
u_i	Usage of customer i (period)
Q	Vehicle tank capacity
t_{ij}	Travel distance from i to j

Two Customer Analysis





Case 1 : If $\frac{u_2}{C_2}(Q - C_2) \geq u_1$, then

$$D^* = \frac{u_2}{C_2} 2t_{02}.$$

Case 2 : If $\frac{u_2}{C_2}(Q - C_2) < u_1$, then

$$D^* = \frac{u_2}{C_2} 2t_{02} + \frac{u_1 - \frac{u_2}{C_2}(Q - C_2)}{Q} 2t_{01}.$$

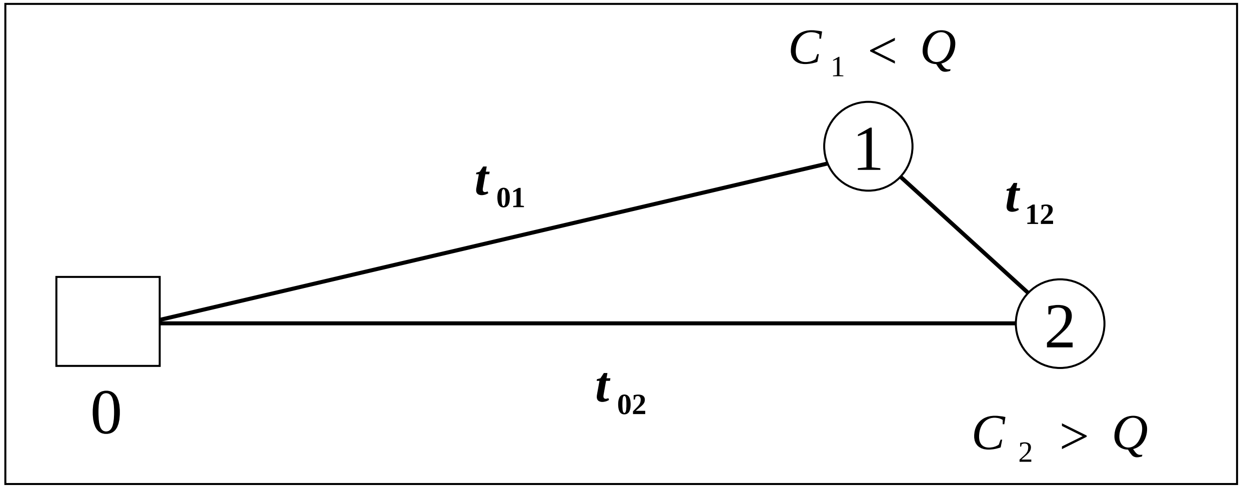
Patterns:

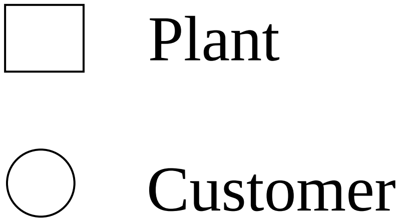
(Q,0)

$(0,C_2)$

$(Q-C_2,C_2)$

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

$$D^* = \frac{u_2}{Q-C_1}(t_{01}+t_{12}+t_{02}) + \frac{u_1 - \frac{u_2}{Q-C_1}C_1}{C_1}2t_{01}$$

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**

$P_j = (d_{j1}, d_{j2}, \ldots, d_{jn})$ **is feasible if**

$\sum_{i \in I} d_{ji} \leq Q$ **and** $0 \leq d_{ji} \leq C_i \ \forall i \in I$.

$\delta(P_j) = \{i \in I : d_{ji} > 0\}$ **: Customers visited in** P_j

$c(P_j)$ **: The cost of delivery pattern** P_j **- optimal TSP value**

$\mathcal{P}$ **: Set of all feasible delivery patterns**

- **Pattern Selection LP**

$$D^* = \min \sum_{P_j \in \mathcal{P}} c(P_j)\ x_j$$

$$\text{s.t. } \sum_{P_j \in \mathcal{P}} d_{ji}\ x_j \geq u_i, \ \forall i \in I$$

$$x_j \geq 0$$

x_j **: How many times should pattern** P_j **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

Obstacle II

- 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 $\min_{i \in I}(C_i) \geq Q/k$, 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 $\lceil Q/\min_{i \in I}(C_i) \rceil \leq k$, then $UB_k = D^*$

    $|\delta(P)| \leq k$ for any base pattern P

For values k=3 and k=4, the TSPs that have to be solved involve at most 4 and 5 stops, respectively, and thus can be solved relatively easily by enumeration

Dominance

- **Do we need base pattern *P* ?**

$$\begin{aligned} z = \quad min \quad & \sum_{\{j:\delta(P_j)\subsetneq\delta(P)\}} c(P_j)\lambda_j \\ s.t. \quad & \sum_{\{j:\delta(P_j)\subsetneq\delta(P)\}} d_{ji}\lambda_j \geq d_i, \quad \forall i \in \delta(P) \\ & \lambda_j \geq 0 \end{aligned}$$

If z ≤ *c(P),*

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

Base pattern *P* can be eliminated from the Pattern Selection LP

Simple Dominance

- Consider base pattern P with $d_4 < C_4$

P

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ d_4 \end{bmatrix}$$

Condition I

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ d_{44} \end{bmatrix}$$

$$d_{44} = min(Q, C_4)$$

Condition II

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ 0 \end{bmatrix} \begin{bmatrix} C_1 \\ 0 \\ 0 \\ d_{14} \end{bmatrix} \begin{bmatrix} 0 \\ C_2 \\ 0 \\ d_{24} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ C_3 \\ d_{34} \end{bmatrix}$$

$$d_{i4} = min(Q - C_i, C_4)$$

Condition III

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ 0 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ 0 \\ d_{124} \end{bmatrix} \begin{bmatrix} C_1 \\ 0 \\ C_3 \\ d_{134} \end{bmatrix} \begin{bmatrix} 0 \\ C_2 \\ C_3 \\ d_{234} \end{bmatrix}$$

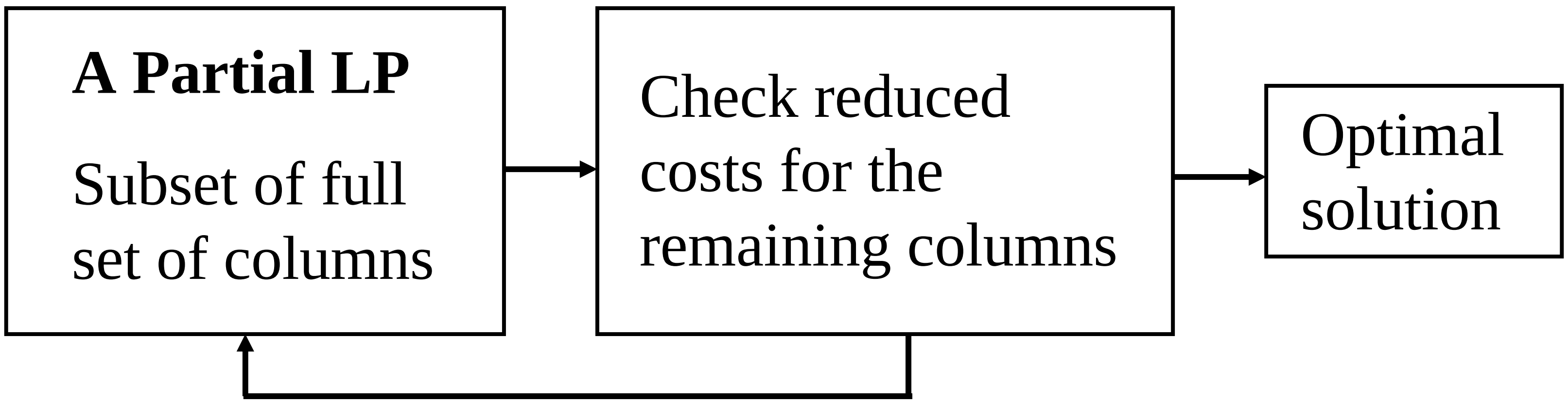
$$d_{ij4} = min(Q - C_i - C_j, C_4)$$

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

Instances	n	before	after
1	136	5,015,046	3,029,980
2	157	7,665,722	4,336,466
3	169	9,086,385	5,420,907
4	147	15,180,701	8,838,137
5	157	14,471,228	8,975,615
6	194	22,575,528	16,640,122

Implementation: Sifting Approach

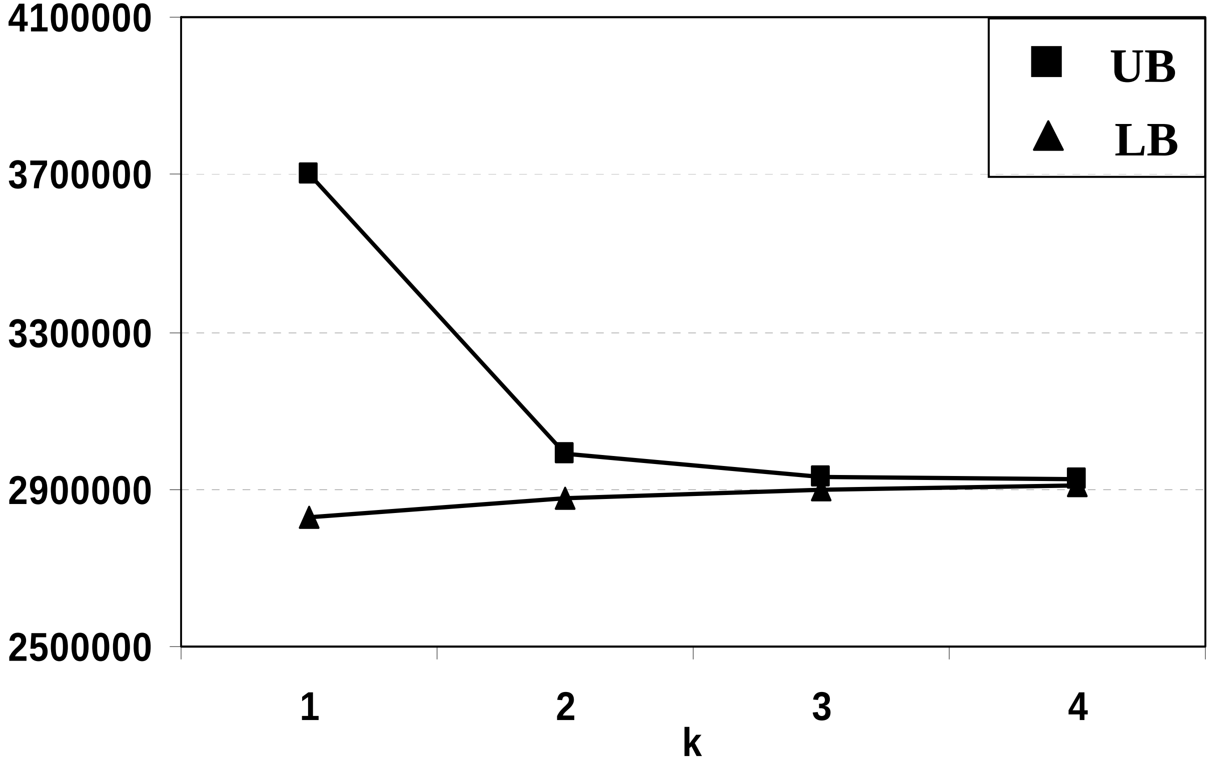
- **Specialized solver for LPs with a large ratio of number of columns to number of rows**



Instance	n	# of patterns	# of iterations	default(sec)	sifting(sec)
1	136	3,029,980	5	85.66	77.09
2	157	4,336,466	6	118.37	121.34
3	169	5,420,907	6	155.29	152.56
4	147	8,838,137	5	360.20	240.83
5	157	8,975,615	6	397.33	254.81
6	194	16,640,122	6	675.98	533.56

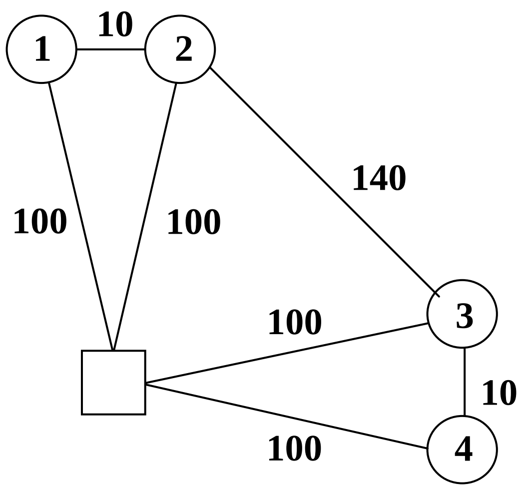
Computational Experiments

- **36 plants, ~2000 customers**



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

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)

Customer	C_i	u_i
1	5000	1000
2	3000	3000
3	2000	2000
4	4000	1500

Q = 5000

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!

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

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:

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

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

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

Vehicle capacity

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

Number of vehicles

Integer Program

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:

$$t_{ri}^{early} = \max \begin{cases} t_{rp(i)}^{early} + tt_{p(i),i} \\ (q_{ri}^{min} - C_i + I_i)/u_i \end{cases}$$

Insertion Heuristic

Latest time a delivery can be made:

$$t_{ri}^{late} = \min \begin{cases} t_{rs(i)}^{late} - tt_{i,s(i)} \\ I_i / u_i \end{cases}$$

Maximum delivery volume:

$$q_{ri}^{max} = \min \begin{cases} Q - \sum_{j \neq i \in r} q_{rj}^{min} \\ C_i \\ C_i - I_i + u_i t_{ri}^{late} - l_i u_i \end{cases}$$

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

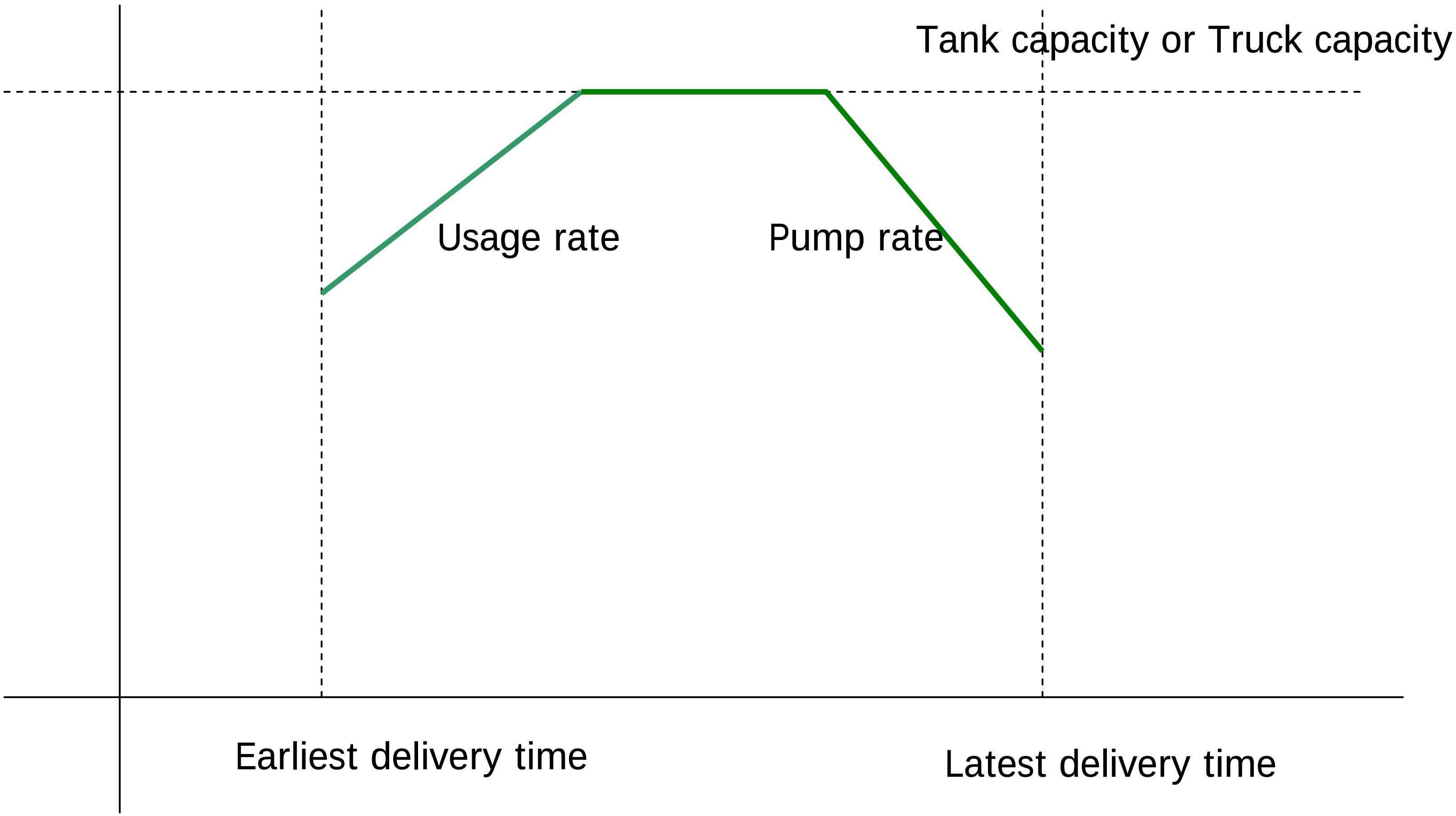
Feasibility check:

- 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 !!

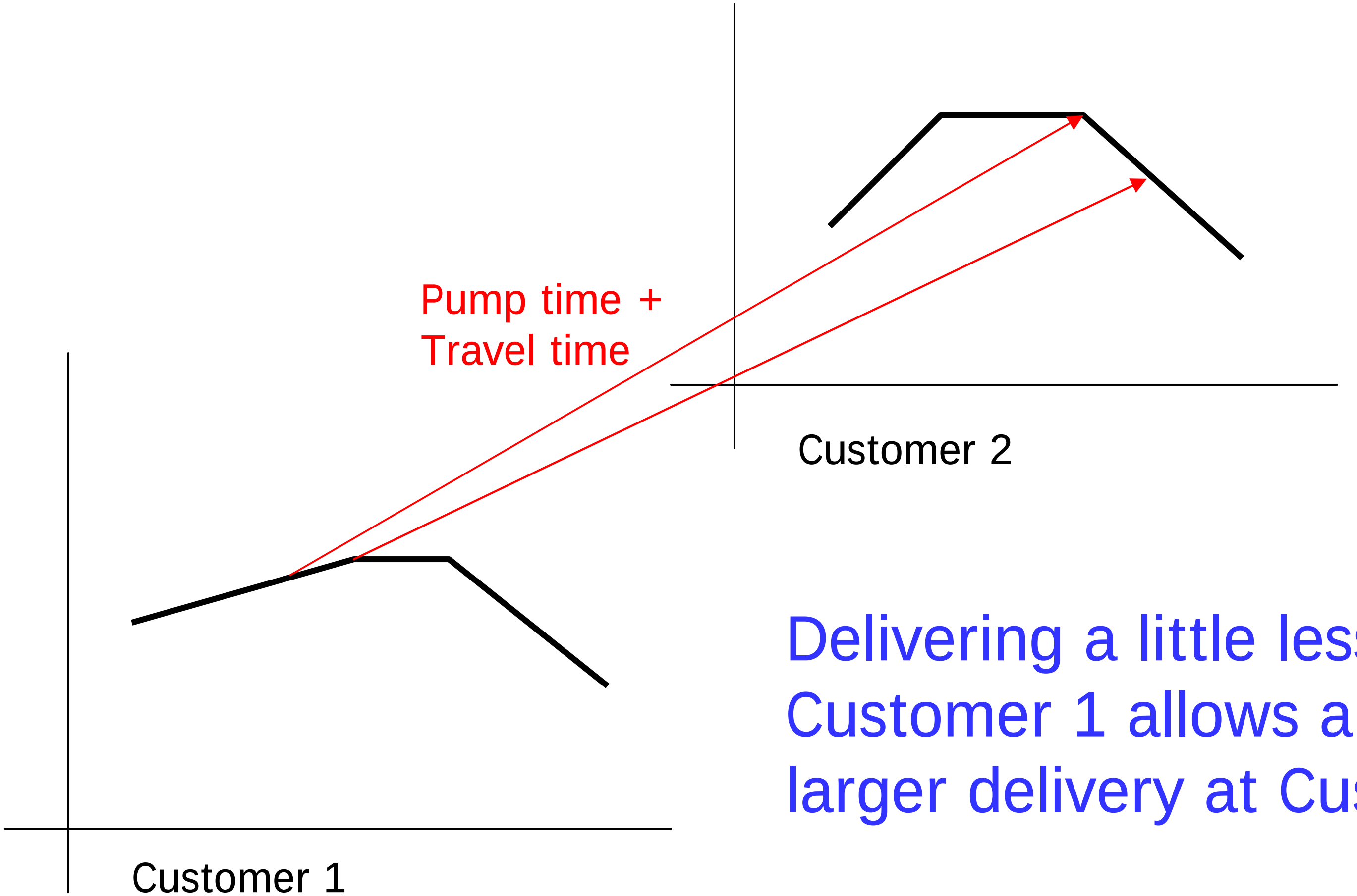
Delivery Volume Optimization



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

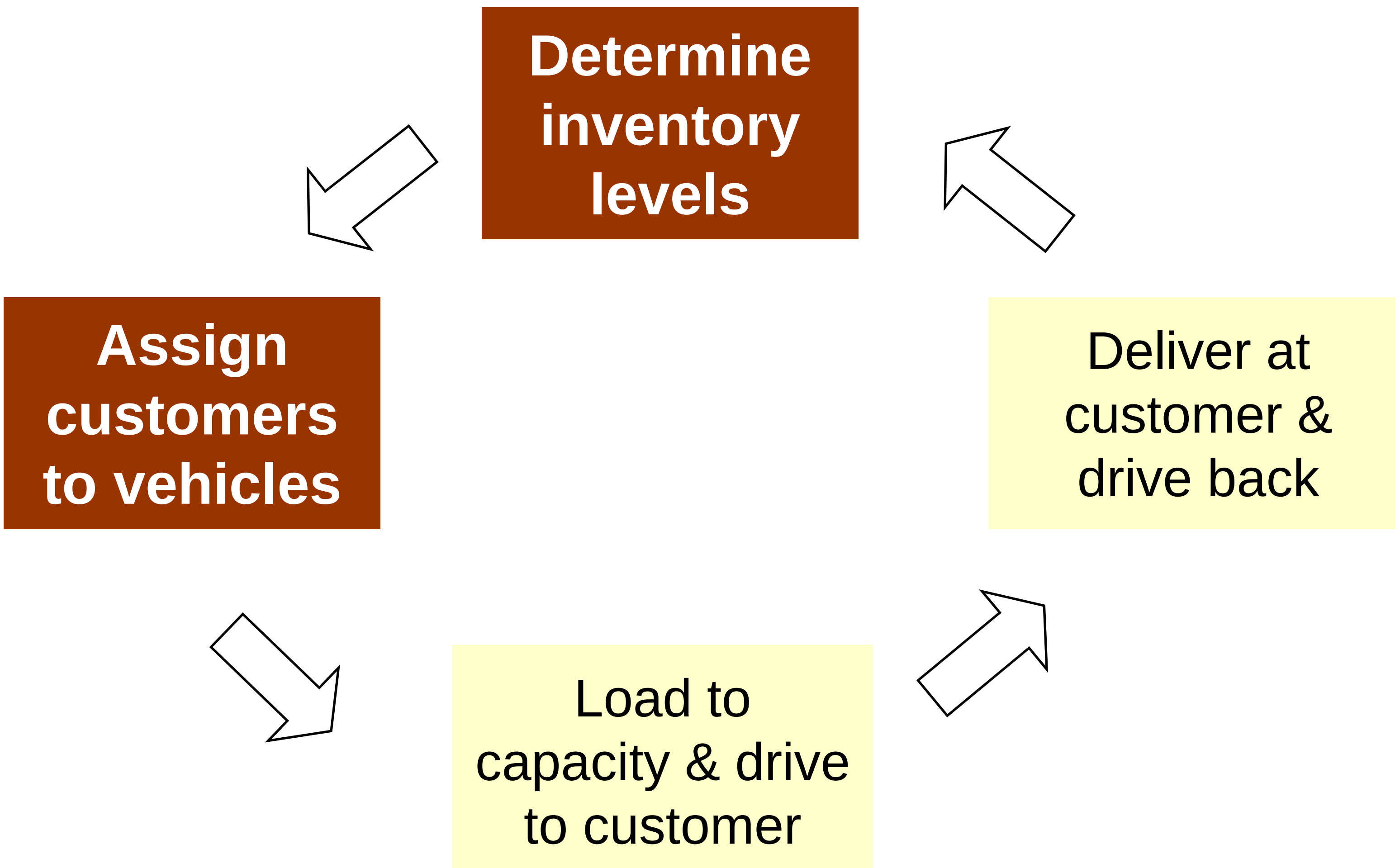


Delivering a little less at Customer 1 allows a much larger delivery at Customer 2

GRASP

- The insertion heuristic is embedded in a Greedy Randomized Adaptive Search Heuristic (GRASP)

Stochastic IRP



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

Instance	Time
2 customers, 1 vehicle	3 sec
3 customers, 2 vehicles	15 min
4 customers, 3 vehicles	24 hr

Direct delivery only

MDP Model: Issues

- Optimality Equation

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

- Computing optimal value function
- Computing expected value
- Computing optimal action

Approximation Methods

- Idea

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

- Motivation

if $\left\| V^* - \widehat{V} \right\| \leq \varepsilon$ and

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

then

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

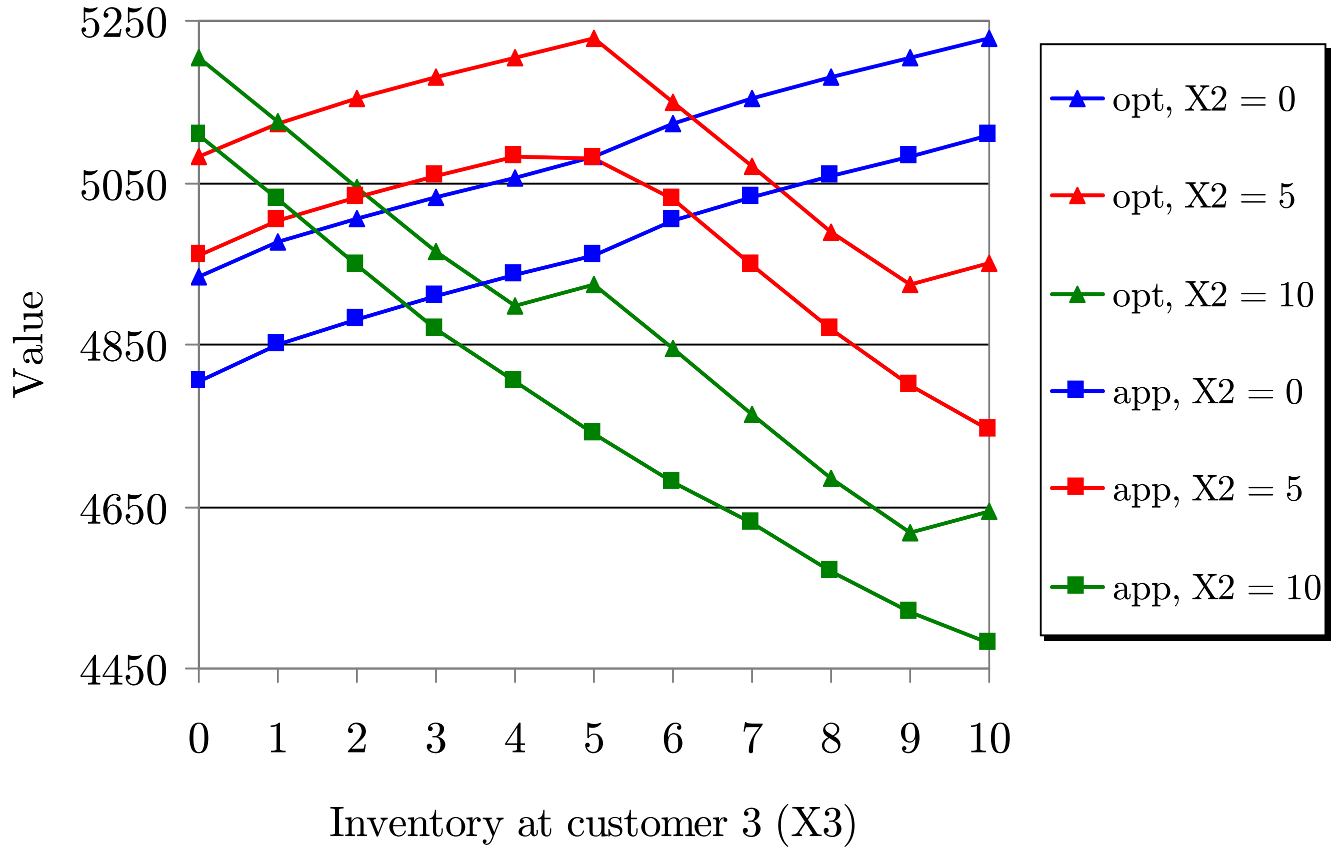
- Parameterized approximation function

$$\widehat{V}(x,\beta) = r_0 + r_1\phi_1(x) \cdots + r_n\phi_n(x)$$

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

Examples for Basis Functions

- Polynomial function
  - inventory level at customers
  - second order effects



THE LOGIS
Georgia Inst
www.tli.gat

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 × 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

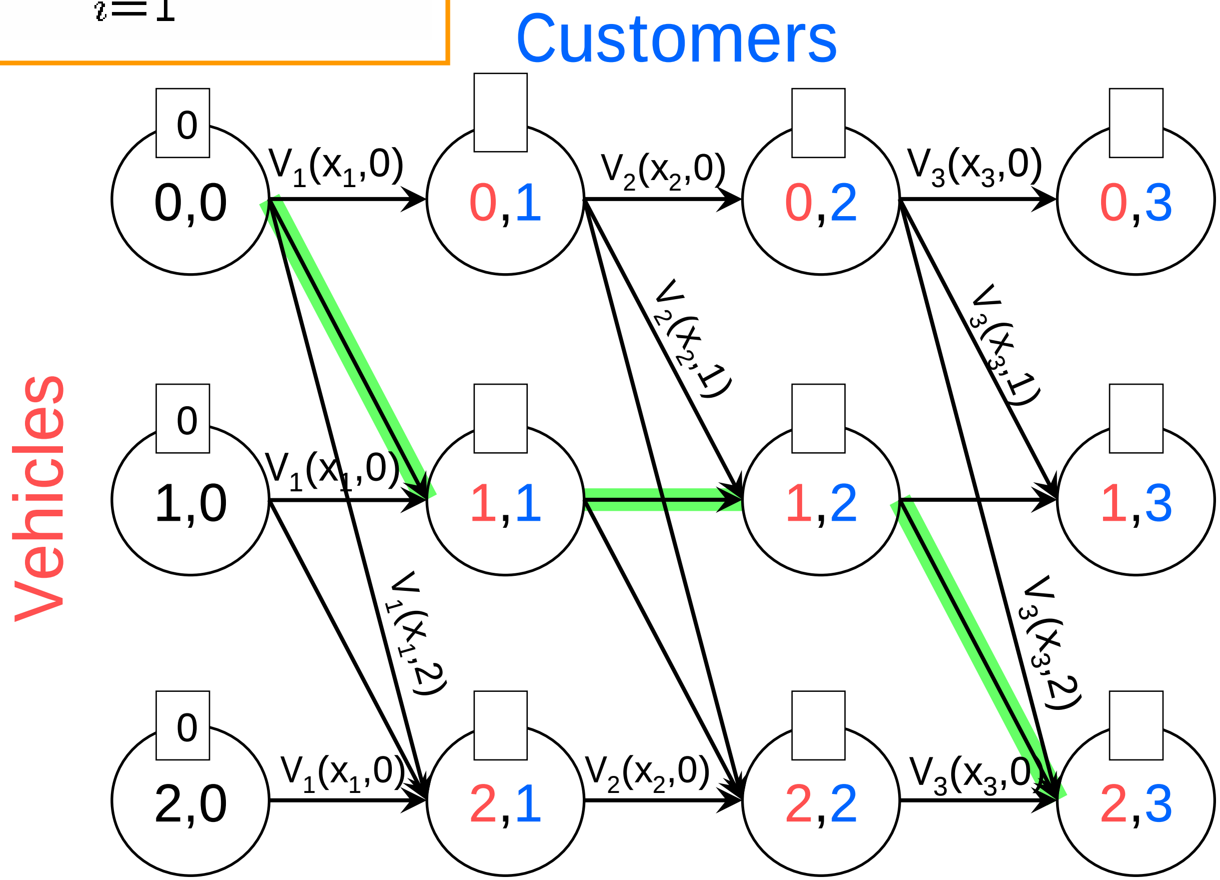
$$\hat{V}(x,r) = r_0 + \sum_{i=1}^{n} r_i V_i^*(x_i, k_i^*)$$

- 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

$$\max_{k_i \in \{0,1,\ldots,K\}} \sum_{i=1}^{n} V_i^*(x_i, k_i)$$

$$\text{subject to} \quad \sum_{i=1}^{n} k_i \leq K$$



Computing Parameters
Method I

- Objective Function

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

- Looks like weighted least squares regression problem

Cannot be computed for large problems

Computing Parameters
Stochastic Approximation Algorithm

- Simulate system under policy π
- Sample path x^0, x^1, …, x^t, ...
- 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 π

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

$$\Downarrow$$

Cannot be computed for large problems

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

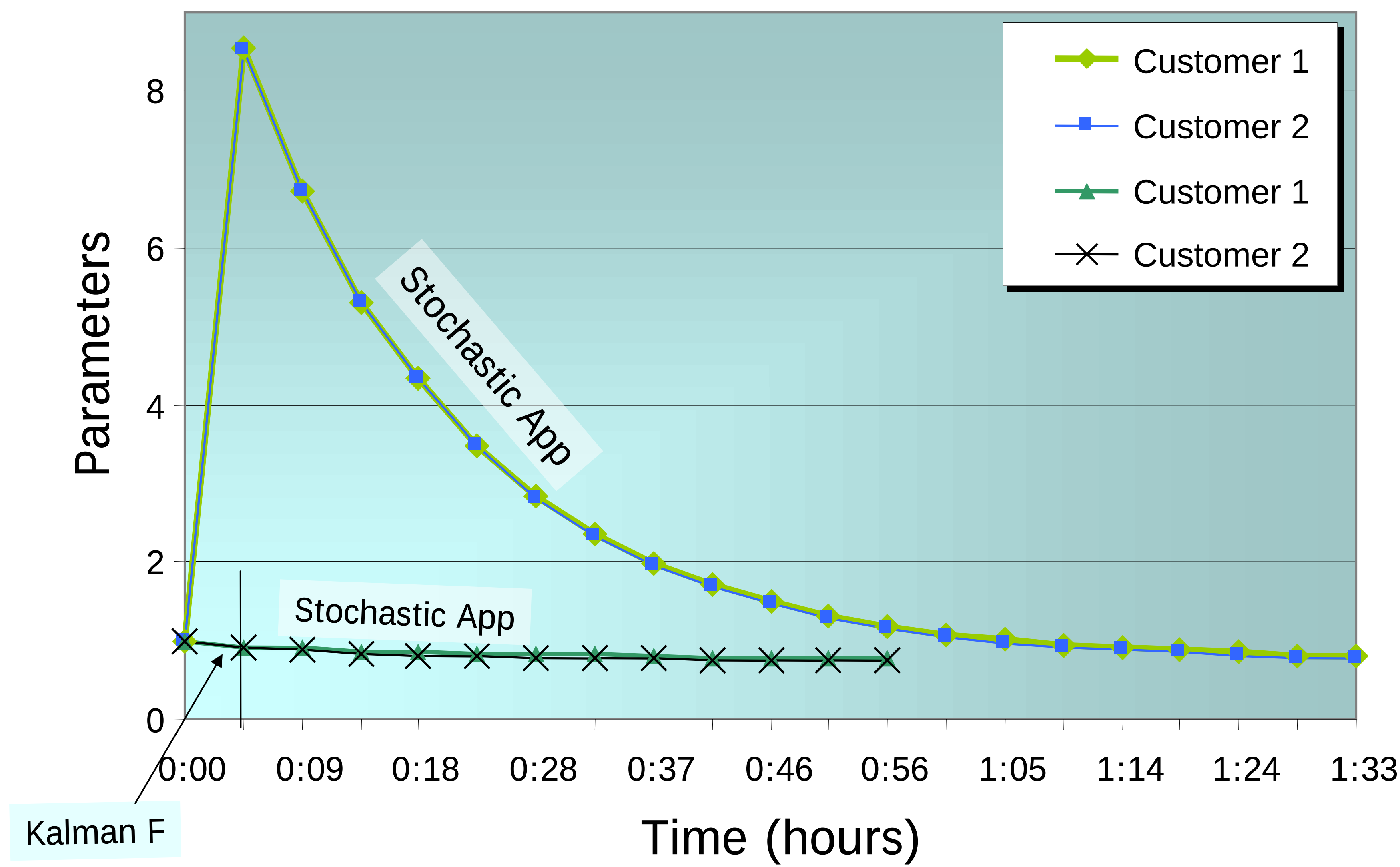
- Looks like weighted least squares regression problem

Computing Parameters
Kalman Filter Algorithm

- Simulate system under policy π
- 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



Estimating Expected Value

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

Randomized methods are better for large d

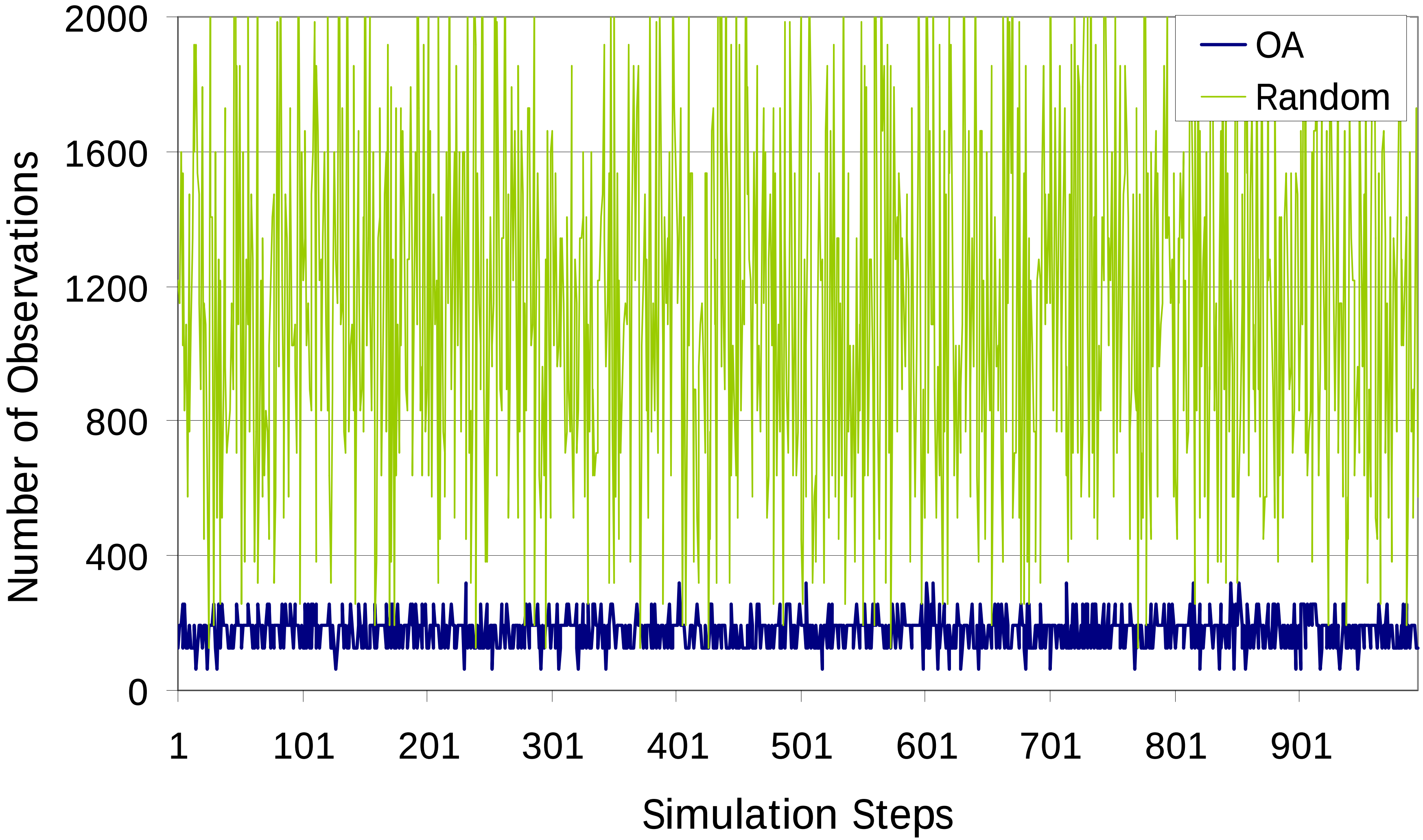
THE LOGISTICS INSTITUTE
Georgia Institute of Technology
www.tli.gatech.edu

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?
- Nelson and Matejcik (1995)
  - 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



Approximate Policy Iteration

1. Initialization. Simulate initial policy π_0 and obtain parameters r^{π_0}

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

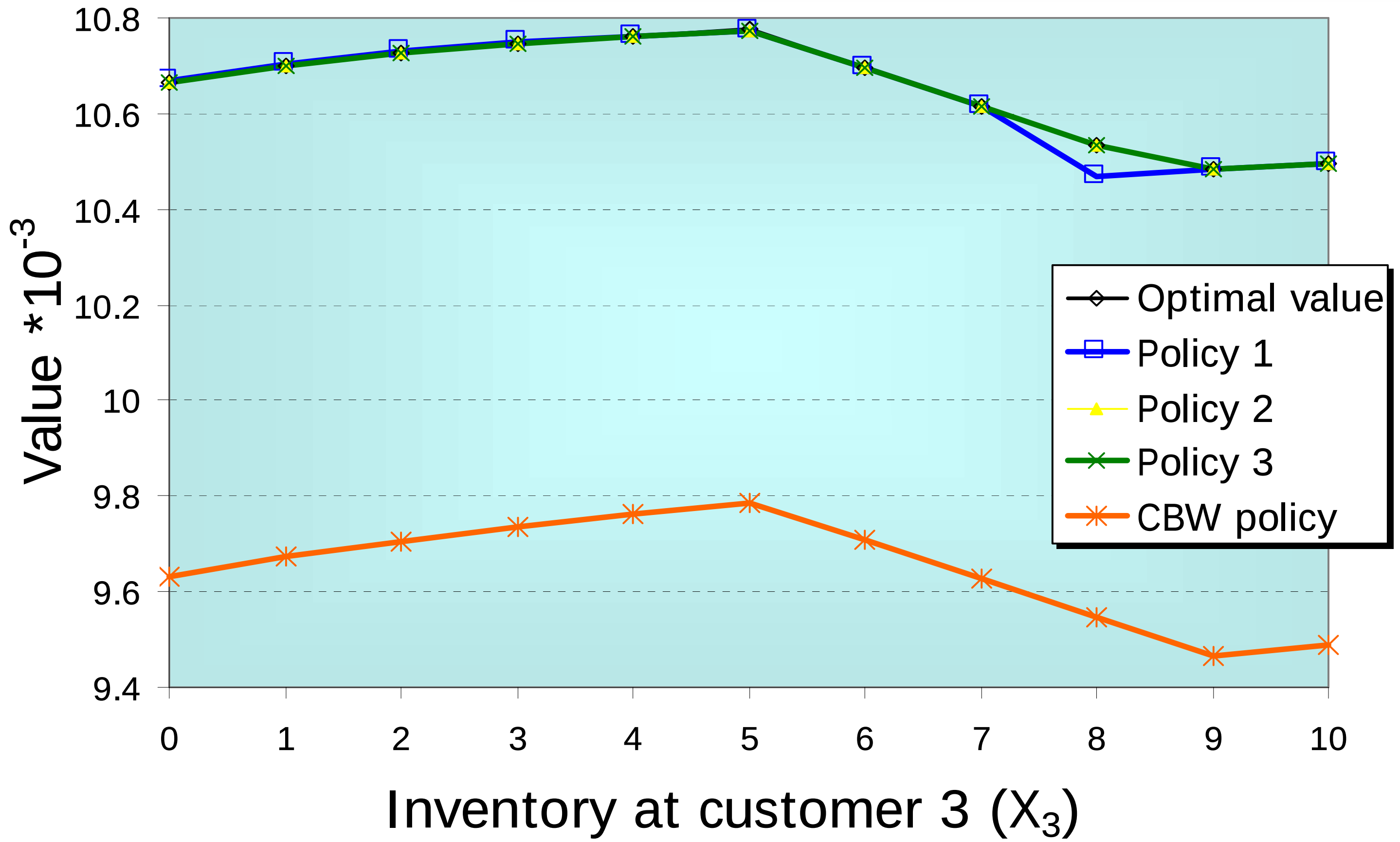
$$\pi_t(x) \in \arg\max_{a \in \mathcal{A}(x)} \left\{ g(x,a) + \alpha \sum_{y \in \mathcal{X}} P[y \mid x,a] \widehat{V}(y, r^{\pi_{t-1}}) \right\}$$

3. Simulate policy π_t and obtain parameters r^{π_t}

4. $t \leftarrow t+1$; go to Step 2

Performance Comparison
Small Instances

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%



Performance Comparison
Large Instances

P	State	π_1			π_{CBW}		
		μ	$\mu - 2\sigma$	$\mu + 2\sigma$	μ	$\mu - 2\sigma$	$\mu + 2\sigma$
1	x_1	400.6	386.7	414.4	380.1	368.7	391.5
	x_2	401.7	388.0	415.4	386.8	374.7	398.9
	x_3	404.1	392.2	416.0	391.0	382.5	399.6
	x_4	403.9	390.2	417.5	392.8	384.7	400.8
2	x_1	892.0	887.5	896.4	854.6	823.3	885.9
	x_2	886.0	871.5	900.4	851.6	820.0	883.2
	x_3	896.4	892.0	900.8	860.0	828.6	891.4
	x_4	891.2	881.9	900.6	856.3	825.7	886.9

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_l, when replenishing a set of customers l.

Price-Direct Replenishment

- Net value for dispatcher

$$\sum_{i \in I} V_i d_i - c_I$$

- Incremental value for dispatcher

$$d_i V_i - \left(c_{I \cup \{i\}} c_I\right)$$

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

$$\max uV$$

$$dV \leq c \quad \forall 0 \leq d \leq min\{C, Q\}$$

replenishment

frequency

usage

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

$$\max uV$$

$$dV \leq c \qquad \forall 0 \leq d \leq min\{C, Q\}$$

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

- P. Jaillet, J. Bard, L. Huang, M. Dror (2002). Delivery cost approximations for inventory routing problems in a rolling horizon framework. TS 36, 292-300.
- A. Campbell, M. Savelsbergh (2004). A decomposition approach for the inventory routing problem. TS 38, 488-502.
- A. Campbell, M. Savelsbergh (2004). Delivery volume optimization. TS 38, 210-223.
- A. Kleywegt, V. Nori, M. Savelsbergh (2002). The stochastic inventory routing problem with direct deliveries. TS 36, 94-118.
- A. Kleywegt, V. Nori, M. Savelsbergh (2004). Dynamic programming approximations for a stochastic inventory routing problem. TS 38, 42-70.
- J.-H. Song, M. Savelsbergh (2006). Performance measurement for inventory routing. TS. To appear.

References

- L. Bertazzi, M.G. Speranza (2002). Continuous and discrete shipping strategies for the single link problem. TS 36, 314-325.
- L. Bertazzi, G. Palletta, M.G. Speranza (2002). Deterministic order-up-to level policies in an inventory routing problem. TS 36, 119-132.
- L. Bertazzi, G. Palletta, M.G. Speranza (2005). Minimizing the total cost in an integrated vendor-managed inventory system. JH 11, 393-419.

References

- D. Adelman (2003). Price-directed replenishment of subsets: methodology and its application to inventory routing. MSOM 5, 348-371.
- V. Gaur, M. Fisher (2004). A periodic inventory routing problem at a supermarket chain. OR 52, 813-822.
- A. Campbell, J. Hardin (2005). Vehicle minimization for periodic deliveries. EJOR 165, 668-684.
- W. Bell, L. Dalberto, M. Fisher, A. Greenfield, R, Jaikumar, P. Kedia, R. Mack, P. Prutzman (1983). Improving the distribution of industrial gases with an online computerized routing and scheduling optimizer. Interfaces 13, 4-23.

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?