SAND2008-3206P

Low-Memory Lagrangian Relaxation for Sensor Placement in Municipal Water Networks

Jonathan W. Berry
Erik Boman
Cynthia A. Phillips
Lee Ann Riesen

Sandia National Laboratories
Computing and Information Sciences
May 13, 2008

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

The Sensor Placement Problem

Issue: Contamination released in a municipal water network

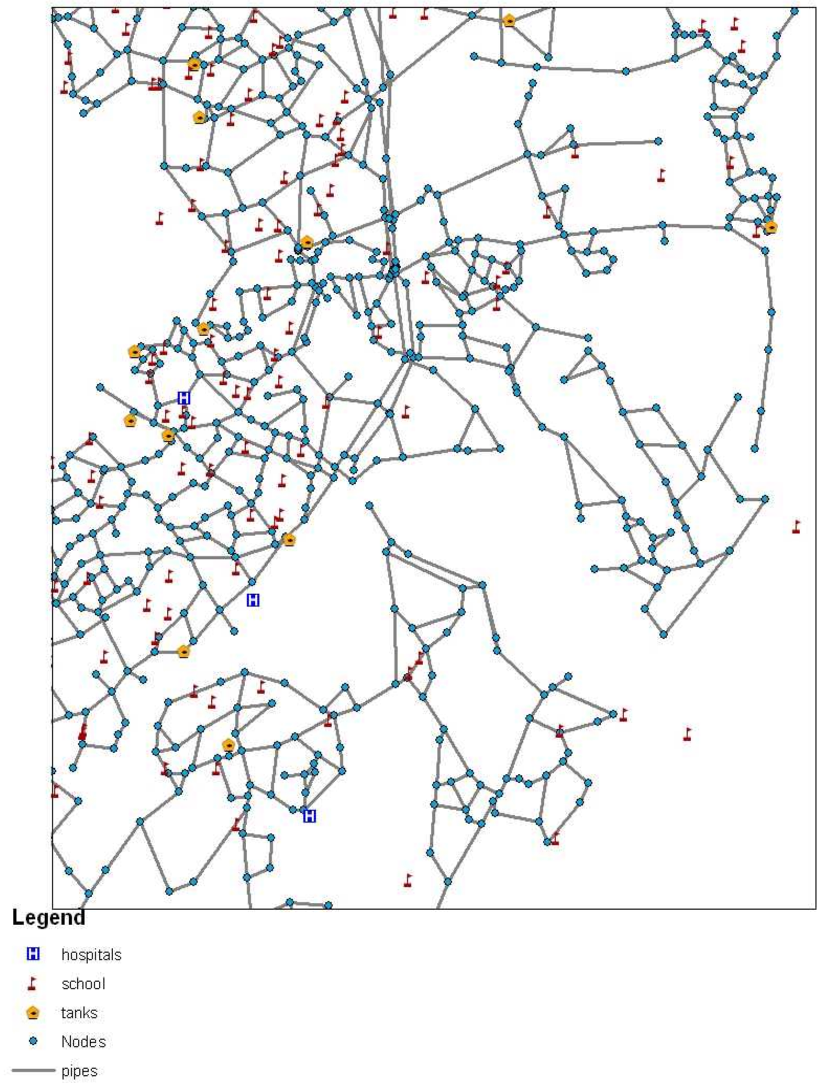
Goal: develop early warning system

- Protect human populations
- Limit network remediation costs

Place sensors on

- Utility-owned infrastructure
- Schools
- hospitals

- Sensors are expensive
  - Cost of sensors
  - Cost of installation



Modeling Assumptions

- Sensors are perfect
- Sensors raise a general alarm
  - Can model a response delay
- Fixed set of demand patterns for “typical” day
  - Seasonal variations
  - Special events
  - Weekday/weekend

Contaminant Transport Modeling

Water movement (direction, velocity in each pipe) determined by

- Demand (consumption)
- Pumps
- Gravity
- Valves
- Sources/tanks

Current (most trusted) simulator

- EPANET code computes hydraulic equations to determine flows
- Discrete-event simulation for contaminant movement

Modeling Events

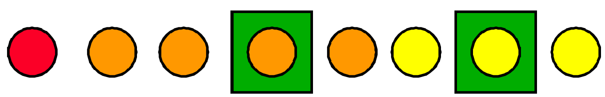
- Given: Set of events = (location, time) pairs
- Simulate the evolution of a contaminant plume
- For each event determine
  - Where/when event can be observed
  - Amount of damage prior to that observation

- Measures of damage/impact:
  - Population exposed
  - # deaths
  - Volume of contaminant release
  - Total pipe length contaminated
  - Time to detection
  - # failed detections

Witnessing an Event

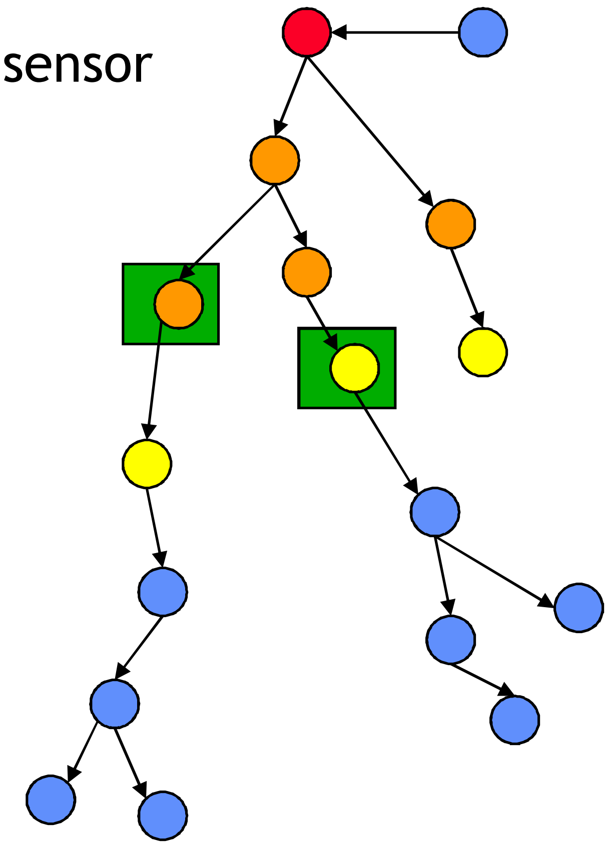
Simulator gives ordered list of nodes where a sensor could witness contamination

Witnesses:



This example has two (green) sensors.

Perfect sensor model: first sensor in list detects the event.



One Sensor Placement IP for Water Networks

Variables:

$$y_i = \begin{cases} 1 & \text{if we place a sensor at location } i \in \mathcal{L}, \\ 0 & \text{Otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if location } i \text{ raises the alarm during event } j \\ 0 & \text{Otherwise} \end{cases}$$

Extreme points will have integer values for x_{ij} if the y_i are integral.

Each event has a dummy location to mark failure to detect

Objective function

Compromise across all “likely” event scenarios to minimize expected damage.

$$\text{minimize} \sum_{j \in A} \sum_{i \in L} \alpha_j w_{ij} x_{ij}$$

α_j – the weight of event $j = (i, t)$

w_{ij} – the total damage from event j if detected at location $i \in \mathcal{L}_j$

x_{ij} – 1 if location i raises alarm in event j, 0 otherwise.

Sensor Placement Mixed Integer Program

minimize $\sum_{j \in A} \sum_{i \in L_j} \alpha_j w_{ij} x_{ij}$

$s.t.$

$\sum_{i \in L_j} x_{ij} = 1 \qquad \forall j \in A \qquad$ (every event witnessed)

$x_{ij} \leq y_i \qquad \forall j \in A, i \in \mathcal{L}_j \qquad$ (need sensor to witness)

$\sum_{i \in L} y_i \leq k \qquad$ (sensor count limit)

$y_i \in \{0,1\}$

$0 \leq x_{ij} \leq 1$

Sensor Placement = k-median

- Place k sensors (on network nodes). Sensors = Facilities
- Events = Customers to be “served” (witnessed)
- “Distance” from an event *j* to a node *i* = impact if a sensor at node *i* witnesses event *j*.

Scalability Challenge

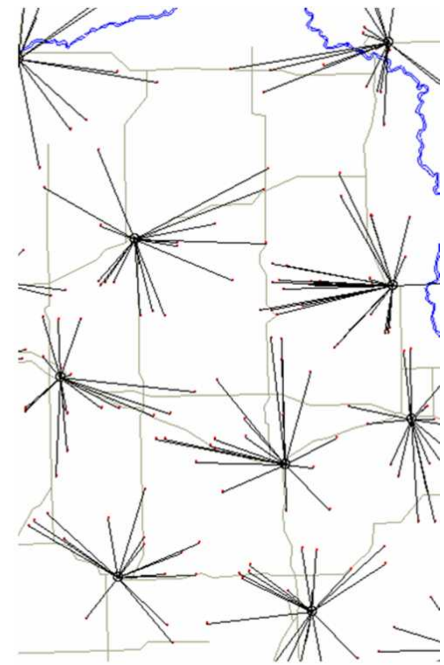
- Full-size problems have
  - 10,000+ nodes
  - 100's of interesting times of day
  - Multiple seasons
  - Weekends/weekdays
  - Special events
  - Multiple contaminant types

- Lots of witness variables
  - trivial upper bound: (# events) x (# nodes)
- Space can be an issue
  - 64-bit workstations
- Linear programming relaxation can be difficult to solve

Sensor Placement Heuristic Solvers

Grasp: Multistart local search

- Neighborhood swaps sensor location with non-location
- Can rapidly solve problems with 10,000's of junctions (SNL-3 in 154 seconds)
- Heuristic solutions are often optimal
- Uses sparse matrix representation, but still requires superlinear space.



Lagrangian Relaxation for p-Median Problems

- What is the biggest challenge in solving this formulation well?:

$\alpha_j w_{ij}$ → c_{ij}

$$\min \sum_{i,j \in A} c_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{i \in L} x_{ij} = 1, \qquad j \in A$$

“Every event must be witnessed by a sensor.”

$$x_{ij} \le y_i \qquad i \in V, j \in A$$

$$\sum_{i \in L} y_i = p$$

$$y_i \in \{0,1\} \qquad i \in V$$

$$x_{ij} \ge 0 \qquad i \in V, j \in A$$

Lagrangian Relaxation for p-Median Problems

- Solution strategy: lift those tough constraints out of the constraint matrix into the objective. (e.g., Avella, Sassano, Vasil'ev, 2003)

$$\min \sum_{i,j \in A} c_{ij} x_{ij} +$$

s.t.

$$x_{ij} \le y_i \qquad i \in L, j \in A$$

$$\sum_{i \in L} y_i = p$$

$$x_{ij} \ge 0 \qquad i \in L, j \in A$$

$$y_i \in \{0,1\} \qquad i \in V$$

Good news: remaining problem easy to solve!

Bad news: some of the original constraints might not be satisfied.

Lifting the Service Constraints

An example violated constraint (event *j* not fully witnessed):

$$\sum_{i \in L} x_{ij} < 1$$

For *j*th service constraint (violation), λ_j is a Lagrange multiplier:

$$\lambda_j \sum_{i \in L} x_{ij} < \lambda_j$$

Multiplier λ_j weights cost of violating *j*th service constraint:

$$\lambda_j - \lambda_j \sum_{i \in L} x_{ij}$$

New objective: $$\min \left(\sum_{i \in L, j \in A} c_{ij} x_{ij} \right) + \sum_{j \in A} \left(\lambda_j - \lambda_j \sum_{i \in L} x_{ij} \right) =$$

$$\min \left(\sum_{i \in L, j \in A} \left(\mathrm{c_{ij}} - \lambda_j \right) x_{ij} + \sum_{j \in A} \lambda_j \right)$$

Solving the Relaxed Problem

New problem:

$$\min\left(\sum_{i\in L, j\in A}(\mathrm{c_{ij}}-\lambda_j)x_{ij}\right)+\sum_{j\in A}\lambda_j$$

subject to:

$$x_{ij}\le y_i \quad \text{for } i\in L, j\in A$$

$$0\le x_{ij}\le 1, \quad y\in\{0,1\}$$

$$\text{For fixed } \lambda_{j,} \text{ let } \rho(i)=\sum_{j\in A}\max\left(0, c_{ij}-\lambda_j\right)$$

Set y_i = 1 for the p locations with lowest values of $\rho(i)$.

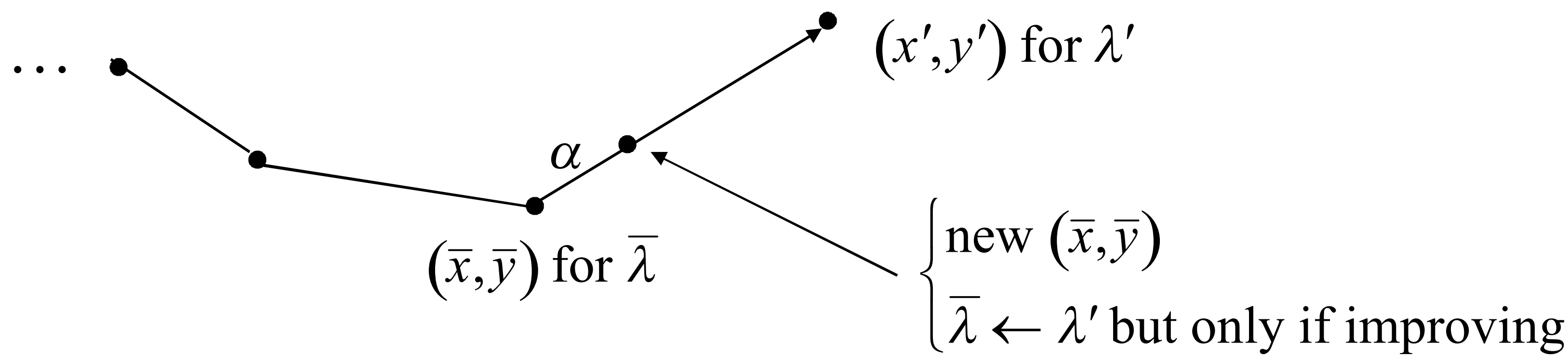
Set x_{ij} = 1 if y_i=1 and c_{ij}-λ_j < 0, x_{ij}=0 otherwise.

- Gives valid lower bound on the best p-median cost
- Linear space, O(W + pn) time for n locations, W potential witnesses.

Finding a Good Solution

- Vol algorithm (Barahuda, Anbil)
  - Subgradient method
  - Finds a feasible LP Solution
- Modified the Vol code for unconstrained facility location from COIN
  - Use sparse data structures
  - p-median instead of ufl

Summary of Vol Algorithm



$v_j = 1 - \sum_{i \in L} \bar{x}_{ij}$ (violation, subgradient)

$\lambda' = \bar{\lambda} + \mathrm{sv}$

$(\mathrm{x}', \mathrm{y}')$ is solution for λ'

step $s = \beta \dfrac{U - L(\bar{\lambda})}{\|v\|^2}$

U: upper bound on LP

$L(\bar{\lambda})$: lower bound for multipliers $\bar{\lambda}$

α, β decrease after nonimproving steps

β increases with improving step

Finding a feasible solution

- Vol provably converges to LP optimal
- Find a feasible solution by randomized rounding
  - Have $\sum_{i \in L} y_i = p$
  - Taking sensor *i* with probability y_i, we *expect* p sensors
  - But the actual probability of getting p sensors is small
  - Use algorithm by Berry and Phillips to efficiently sample directly from the “lucky” distribution.

Simple computational example

- Network with about 3358 nodes, 1621 events, 5 sensors
- Memory: Lagrangian: 45028kb, Heuristic: 154424kb
- Memory increases rapidly, by 13,000 events, Heuristic ~1Gb

Objective	Solver	Gap	Time
Pop. Exposed	Lagrangian	.008	84.3s
Pop. Exposed	Heuristic	0	33.8
Extent Contam.	Lagrangian	0	73.1s
Extent Contam.	Heuristic	0	33.2
Mass Consumed	Lagrangian	.049	85.4s
Mass Consumed	Heuristic	0	41.7
Vol. Consumed	Lagrangian	.641	104.7s
Vol. Consumed	Heuristic	0	44

Handling Side-Constrained p-Median Problems

- We can lift a side constraint too:

An example violated side constraint (d_{ij} impacts too high):

$$\sum_{i \in L, j \in A} d_{ij} x_{ij} > B$$

Violation, weighted by new single Lagrange multiplier λ_s :

$$\lambda_s \sum_{\mathrm{i} \in \mathrm{L}, \mathrm{j} \in \mathrm{A}} d_{ij} x_{ij} - \lambda_s \mathrm{B}$$

New objective function:

$$\sum_{i \in L, j \in A} \left(\mathrm{c}_{\mathrm{ij}} + d_{ij}\lambda_s - \lambda_j\right) x_{ij} + \sum_{j \in A} \lambda_j - \lambda_s$$

Future Work

- Update the multipliers differently for service constraints vs side constraints
- Can the Lagrangian method take advantage of a "free" dummy sensor?
  - Selecting p+1 not always correct
- Sensitive to scaling

Simple Multiobjective Example

Solver	EC Goal	VC Goal	MC	EC	VC
Optimal individual			638,344.7	40,867	217,001
Lagrang.	45,000	250,000	678,175	49,016	256,615
Heuristic	50,000		670,399	49,827	326,943
Heuristic		250,000	640,791	66,728	243,605

- Lagrangian ~ 2min. Heuristic ~3min

Still Lots of Work To Do

- Mc objective. EC Goal = 45,000, VC Goal = 250,000
- With huge goal same answers as with no PE goal. Memory 114k

PE Goal	MC (648345)	VC (217002)	EC (40867)	PE (2653)	time
10000000	678175.5	256615.3	49016.9	3093	147s
10000	678175.5	256615.3	49016.9	3093	151s
4000	714817	411050.5	47669.7	3158	150s
3100	720030.8	477922.6	73912.9	5403.3	88s
2700	715783.6	411934.9	46759	3047	182s