Modeling, Simulation and Optimization to Guide Radiological and Nuclear Searches

Claudio Santiago

Thomas Edmunds, Daniel Faissol, David Lange, Karl Nelson, Kevin Ryan, Richard Wheeler

Lawrence Livermore National Laboratory

May 13th, 2015

Lawrence Livermore National Laboratory

LLNL-PRES-**670564**

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Outline

- Overview of current effort: OPTUS (Optimization Planning Tool for Urban Search)
- Approach and results of previous 2012 effort
- Optimization models and solution approaches for OPTUS
- Experiments
- Conclusion

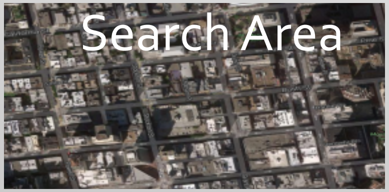
Objective: Develop modeling/simulation techniques to enhance urban radiological and nuclear searches

- Search teams "clear" areas of radiological and nuclear threats prior to special events with vehicle-mounted and human-portable detectors
- Current search plans are based on simple estimates of detector and CONOP effectiveness resulting in inefficient use of resources and uncertain performance
  - A systematic quantitative-based approach to improve performance is needed
- 2012 LLNL study developed a discrete optimization approach that showed significant reductions in search time when clearing building interiors with a minimum signal-to-noise ratio

OPTUS will integrate data, models & algorithms to enable planners to develop more effective searches

OPTUS will be made up of four major components

1. Background radiation characteristics of search area based on radiation transport models and/or measurements
2. Models of attenuation through building walls of different construction
3. Combined measurements detection algorithm to optimally exploit measurements from multiple passes and vantage points
4. **Optimization and computational models to compute search plan that maximizes the probability of detection given available resources**



+

- City/building databases
- Site-specific measurements
- Threat information
- Available search assets

Detection algorithms
Background models

OPTUS

Attenuation models
Optimization models

High interest locations

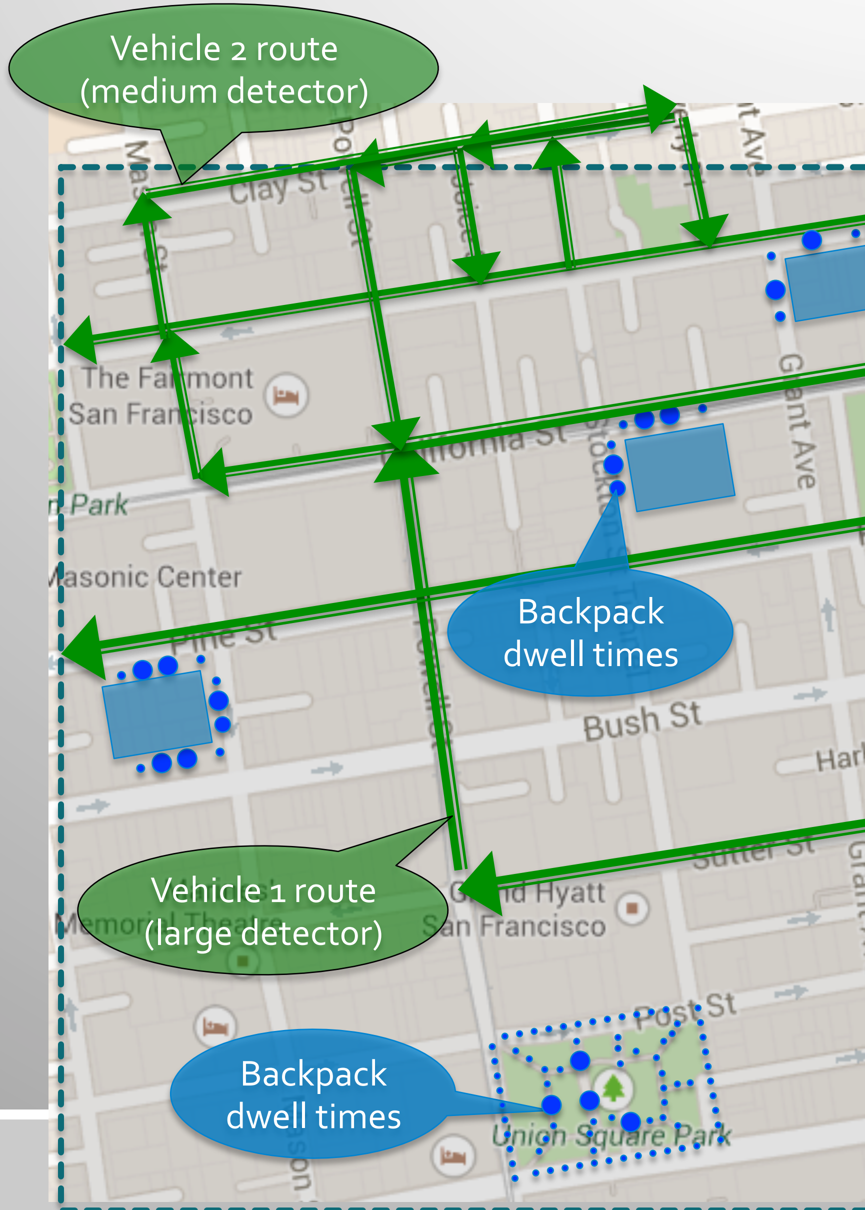
Optimal dwell times per location for human portable detector(s)

+

Wide area exterior

Optimal routes for vehicle-mounted detector(s)

Sample OPTUS output



<u>Inputs</u>

- Urban infrastructure data for search area
- Total search time available
- Source assumptions
- Specification of *High Interest Locations*
  - Specify relative probability that source is in each (optional)
- Number of human portable and vehicle-mounted detectors
  - Size and max/min speeds for each

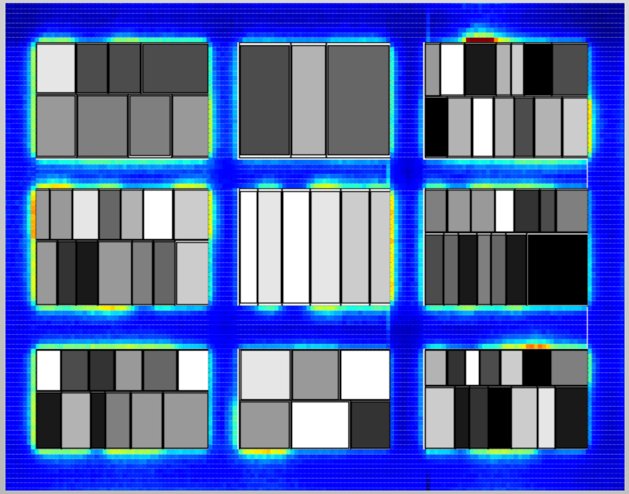
<u>Outputs</u>

- **For *High Interest Locations***
  - **dwell times and locations to achieve the highest probability of detection across all high interest buildings**
- **For *Wide Area Exterior***
  - **Search routes for each vehicle that maximizes probability of detection**

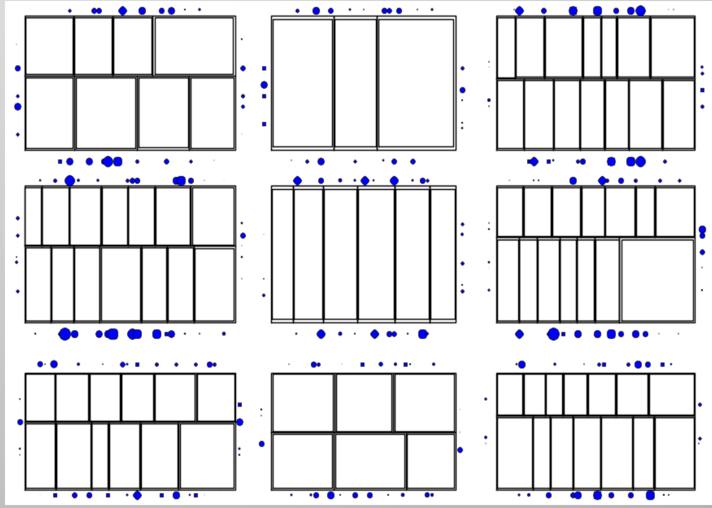
Approach and results from 2012 pilot study

- Focus of pilot study was on "clearing" building interior from roads and sidewalks in minimum time
- Optimal search performs better than constant speed, except when there is extreme variation in wall thicknesses *and the thicknesses are unknown and poorly estimated*
  - 3-30 faster when wall thicknesses are known or appropriately estimated

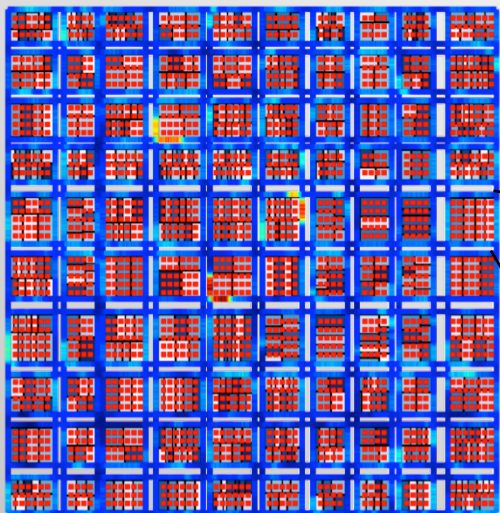
Urban Scene Simulator Output



Optimal Solution



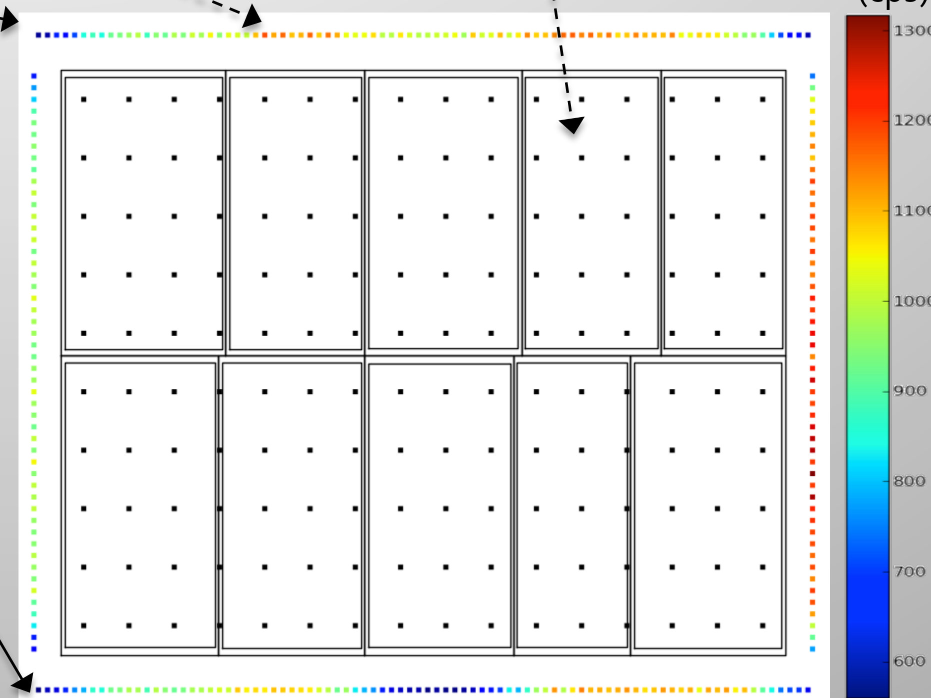
Approach: discretize potential detector and threat locations to solve for optimal search times/velocities



urban scene
10 x 10 city
blocks

Discrete detector locations along exterior of building on roads

Discrete potential source locations inside building



single urban block

Each detector location has an associated:

1. Background radiation estimate
2. Signal attenuation from every potential threat source location

Optimization model developed to find dwell times at detector locations for arbitrary geometry, background, and attenuation

SNR at a potential source location j is

$$SNR_j = \frac{\sum S_{ij} t_i}{\sqrt{\sum B_i t_i}}$$

such that detector location i contributes to clearing potential source location j

t_i = time spent at detector location i

S_{ij} = expected attenuated signal at i from j

B_i = background rate at i

Optimization formulation

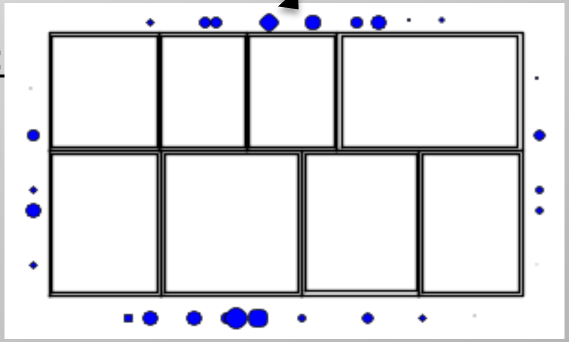
$$\text{Minimize } \sum t_i \text{ such that}$$

$$SNR_j \geq SNR_{desired} \text{ for all } j$$

$$t_i \geq \text{minimum dwell time per detector location}$$

sample solution:

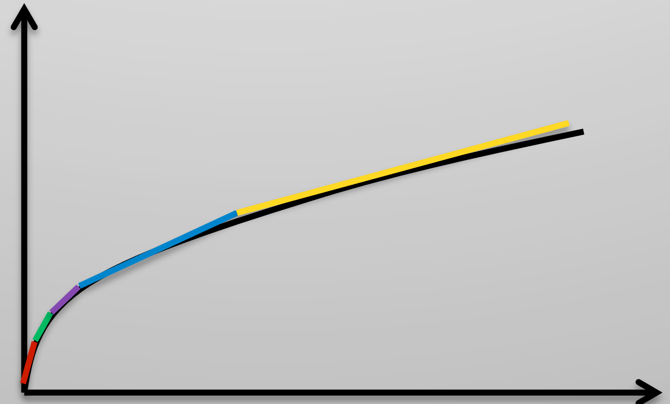
optimal t_i's shown as blue dots (size proportional to dwell time)



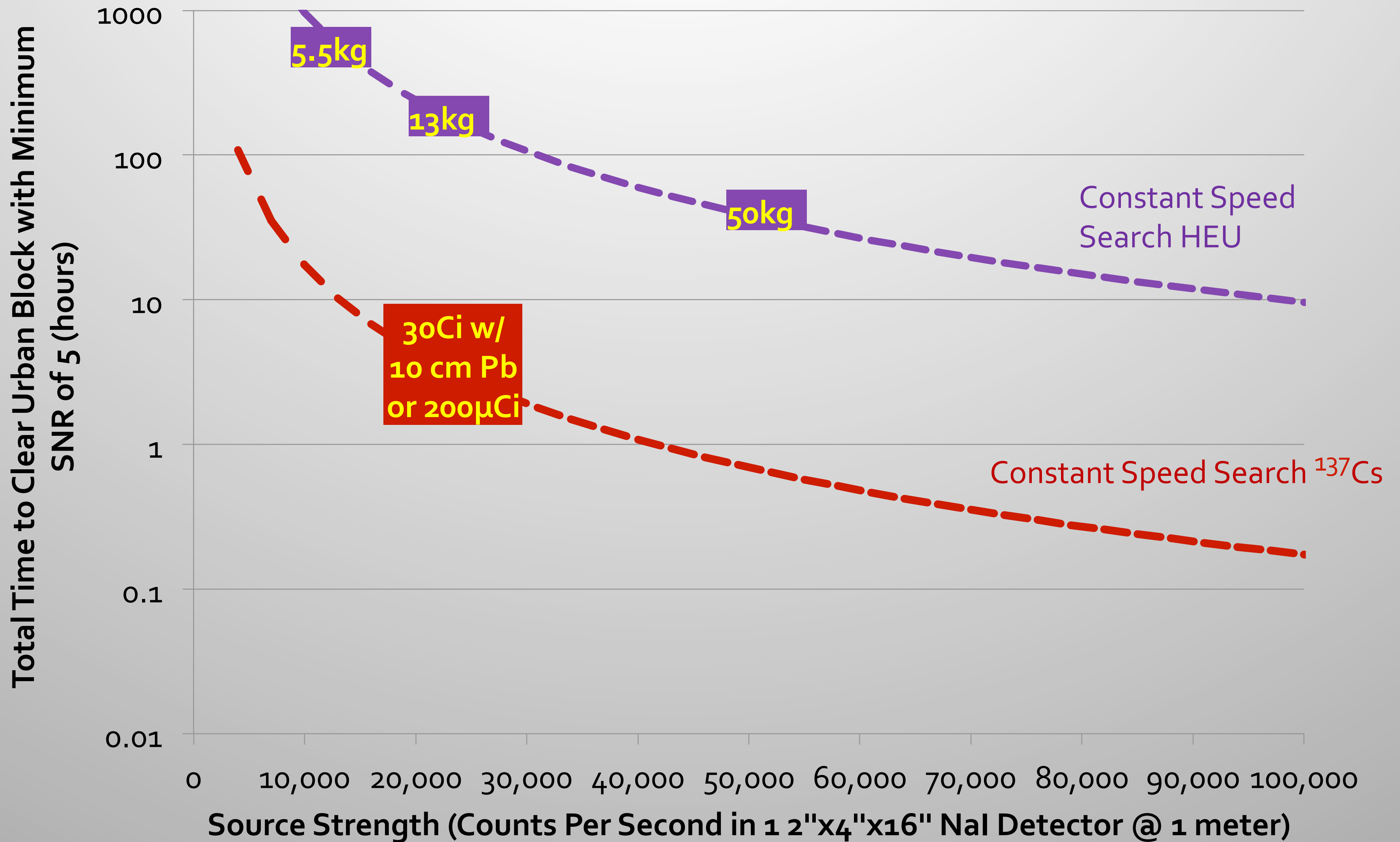
This is a non-convex optimization problem

To solve, we approximate each non-convex function with a piecewise linear function

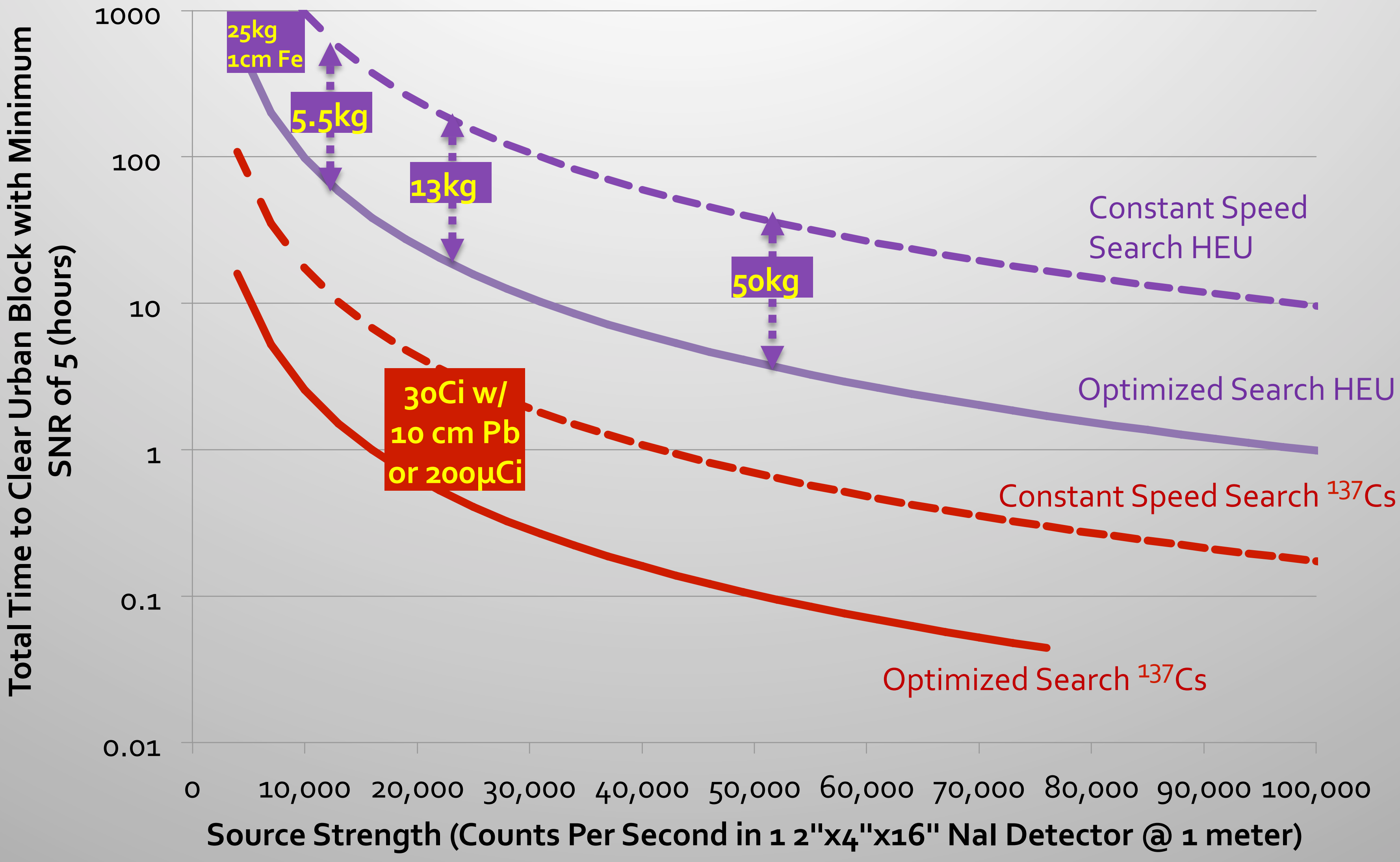
- Can reformulate problem as a mixed binary-linear optimization problem
- Iteratively solve the optimization problem, increasing the number of line segments each iteration
  - Split line segments based on optimal solution from previous iteration
- Approach empirically shown to find an optimal solution efficiently within a specified error tolerance



Optimized search clears urban block 7-10 times faster than constant speed search



Optimized search clears urban block 7-10 times faster than constant speed search



Optimization models for OPTUS

- Wide Area Exterior
  - mathematical programming model
- High Interest Location
  - mathematical programming model
  - heuristics
  - results

New HIL optimization problem for OPTUS

Minimize time given minimum desired signal-to-noise ratio (SNR)

$$(P)\min \sum_i t_i$$

$$s.t.$$

$$\sum_d S_{ds} t_d \geq SNR_{desired} \sqrt{\sum_d B_d t_d} \quad \forall s$$

Maximize probability of detection given allotted time

$$(Q)\max \sum_{s,r} \left(prior_s u_s^r \right) z_s^r$$

$$s.t.$$

$$\sum_d S_{ds} t_d \geq SNR_{desired} \sqrt{\sum_d B_d t_d} - M_s \left(1 - z_s^r \right) \quad \forall s, r$$

$$\sum_i t_i \leq T$$

z^r defines discrete levels of detection probability (e.g., $z^1 = 0$, $z^2 = .3$, $z^3 = .5$, $z^4 = 1$)
u^r captures marginal increase in probability of detection at each discrete level z^r

Going from (P) to (Q)

- Non-convex inequalities are the same:
  - SNR calculated >= SNR desired
  - for probability of detection it only needs to be satisfied for a subset of points
- (P) and (Q) are equally hard in theory, but (Q) has more variables, making it more time consuming in practice

Solution strategy:

- IP modeling used to solve (P) can be directly used to solve (Q) – SOS2 valid inequalities
- Devised heuristic to solve (P)
- Apply heuristic to solve a sequence of (P) problems in order to obtain a good bound, OR
- Change the heuristic to solve (Q) directly

Preliminary Results for HIL - % difference from IP

	wall thicknesses (cm)	IP (total search time in min)	**Heuristic 1**	**Heuristic 2**	Heuristic 2 + IP
Concrete	5	14	14%	9%	-0.20%
	9	47	14%	14%	0.00%
	13	178	14%	14%	0.10%
	17	697	14%	14%	0.30%
	21	2,734	14%	14%	0.00%
	25	10,702	15%	15%	0.30%
Granite	5	23	16%	13%	-0.20%
	9	86	14%	14%	0.00%
	13	****	****	****	****
	17	1,315	12%	12%	2.40%
	25	20,462	12%	12%	-1.10%
Brick	5	20	16%	14%	0.11%
	9	72	14%	14%	2.26%
	13	285	12%	12%	-1.06%
	17	1,108	13%	13%	-0.10%
	21	4,348	14%	14%	6.96%
	25	17,029	14%	14%	0.90%

Preliminary Results for HIL - % difference from IP

	wall thicknesses (cm)	IP (total search time in min)	**Heuristic 1**	**Heuristic 2**	Heuristic 2 + IP
Concrete	5	14	14%	9%	-0.20%
	9	47	14%	14%	0.00%
	13	178	14%	14%	0.10%
	17	697	14%	14%	0.30%
	21	2,734	14%	14%	0.00%
	25	10,702	15%	15%	0.30%
Granite	5	23	16%	13%	-0.20%
	9	86	14%	14%	0.00%
	13	****	****	****	****
	17	1,315	12%	12%	2.40%
	25	20,462	12%	12%	-1.10%
Brick	5	20	16%	14%	0.11%
	9	72	14%	14%	2.26%
	13	285	12%	12%	-1.06%
	17	1,108	13%	13%	-0.10%
	21	4,348	14%	14%	6.96%
	25	17,029	14%	14%	0.90%

Preliminary Results for HIL - % difference from IP

	wall thicknesses (cm)	IP (total search time in min)	**Heuristic 1**	**Heuristic 2**	Heuristic 2 + IP
Concrete	5	14	14%	9%	-0.20%
	9	47	14%	14%	0.00%
	13	178	14%	14%	0.10%
	17	697	14%	14%	0.30%
	21	2,734	14%	14%	0.00%
	25	10,702	15%	15%	0.30%
Granite	5	23	16%	13%	-0.20%
	9	86	14%	14%	0.00%
	13	****	****	****	****
	17	1,315	12%	12%	2.40%
	25	20,462	12%	12%	-1.10%
Brick	5	20	16%	14%	0.11%
	9	72	14%	14%	2.26%
	13	285	12%	12%	-1.06%
	17	1,108	13%	13%	-0.10%
	21	4,348	14%	14%	6.96%
	25	17,029	14%	14%	0.90%

<16% <15%

Preliminary Results for HIL – Time to solve (sec)

	wall thicknesses (cm)	IP	**Heuristic 1**	**Heuristic 2**	Heuristic 2 + IP
Concrete	5	0.96	0.18	3.76	0.51
	9	2.43	0.15	1.6	1.18
	13	5.27	0.13	1.55	7.73
	17	225.88	0.11	1.51	80.61
	21	56.52	0.1	2.31	300.11
	25	300.14	0.1	2.34	31.45
Granite	5	1.47	0.19	4.4	0.99
	9	5.54	0.14	1.61	5.6
	13	300.11	0.14	1.62	300.1
	17	300.12	0.12	1.57	300.09
	25	3.11	0.1	2.31	2.4
Brick	5	1.39	0.19	4.49	0.87
	9	1.9	0.15	1.72	1.76
	13	84.53	0.14	1.73	4.08
	17	54.3	0.12	1.59	169.03
	21	64.29	0.11	2.38	300.13
	25	6.18	0.1	2.4	4.52

Preliminary Results for HIL – Time to solve (sec)

	wall thicknesses (cm)	IP	**Heuristic 1**	**Heuristic 2**	Heuristic 2 + IP
Concrete	5	0.96	0.18	3.76	0.51
	9	2.43	0.15	1.6	1.18
	13	5.27	0.13	1.55	7.73
	17	225.88	0.11	1.51	80.61
	21	56.52	0.1	2.31	300.11
	25	300.14	0.1	2.34	31.45
Granite	5	1.47	0.19	4.4	0.99
	9	5.54	0.14	1.61	5.6
	13	300.11	0.14	1.62	300.1
	17	300.12	0.12	1.57	300.09
	25	3.11	0.1	2.31	2.4
Brick	5	1.39	0.19	4.49	0.87
	9	1.9	0.15	1.72	1.76
	13	84.53	0.14	1.73	4.08
	17	54.3	0.12	1.59	169.03
	21	64.29	0.11	2.38	300.13
	25	6.18	0.1	2.4	4.52

Preliminary Results for HIL – Time to solve (sec)

	wall thicknesses (cm)	IP	**Heuristic 1**	**Heuristic 2**	Heuristic 2 + IP
Concrete	5	0.96	0.18	3.76	0.51
	9	2.43	0.15	1.6	1.18
	13	5.27	0.13	1.55	7.73
	17	225.88	0.11	1.51	80.61
	21	56.52	0.1	2.31	300.11
	25	300.14	0.1	2.34	31.45
Granite	5	1.47	0.19	4.4	0.99
	9	5.54	0.14	1.61	5.6
	13	300.11	0.14	1.62	300.1
	17	300.12	0.12	1.57	300.09
	25	3.11	0.1	2.31	2.4
Brick	5	1.39	0.19	4.49	0.87
	9	1.9	0.15	1.72	1.76
	13	84.53	0.14	1.73	4.08
	17	54.3	0.12	1.59	169.03
	21	64.29	0.11	2.38	300.13
	25	6.18	0.1	2.4	4.52

<0.2 **<5.0**

Preliminary Results for HIL

%IP solution	Heuristic 1	Heuristic 2	Heuristic 2 + IP
Average	13.88%	13.25%	0.67%
Std. dev.	1.20%	1.44%	1.93%

Time to solve (sec)	IP	Heuristic 1	Heuristic 2	Heuristic 2 + IP
Average	83.18	0.13	2.29	88.89
Std. dev.	117.57	0.03	0.99	128.00

Conclusion

- IP solution times varies considerably across both thickness and material type
- Heuristic 2 did not improve much the solution obtained by Heuristic 1
- Solution times or both Heuristic 1 and 2 did not vary much
- Heuristic 1 produced a solution within 16% of the IP solution in < 0.2s for all instances tested
- IP was not faster with the bound provided by heuristic 2 (Integer Program + Heuristic 2)
- **Both heuristics can be used to solve approximately new HIL problem**

Questions, Comments?

Approach for WAE Problem

- Convert road network into directed graph
- Each arc has a *benefit* representing the increase in probability of detection for traversing it
  - Replicate arcs to represent multiple passes and assign appropriate benefit and cost to each
  - Diminishing returns for each subsequent pass
- Assign *cost* to each arc representing traversal time
- Resembles existing problems
  - Min Max k-vehicle Chinese postman problem
    - cover all arcs at least once minimizing the maximum tour
  - Maximum benefit Chinese postman problem
    - maximize profit across all edges with one vehicle
- But has important differences

WAE problem formulation for 1 vehicle mounted detector

Maximize the total benefit given available time

- b_{ijn} = benefit for traversing arc from node i to j for nth time (decreasing in n)
- t_{ij} = time to traverse arc from node i to j
- *Decision variable* x_{ijn} = 1 if the arc from node i to j gets traversed at least n times
- No need for sub-tour elimination constraints?

$$\max \sum_{i,j,n} b_{ijn} x_{ijn}$$

$$s.t.$$

$$\sum_{k,n} x_{kjn} - \sum_{k,n} x_{jkn} = 0 \;\; \forall\, j$$

$$\sum_{i,j,n} t_{ij} x_{ijn} \leq T$$

$$x_{ijn} \;\; binary \;\; \forall i, j, n$$

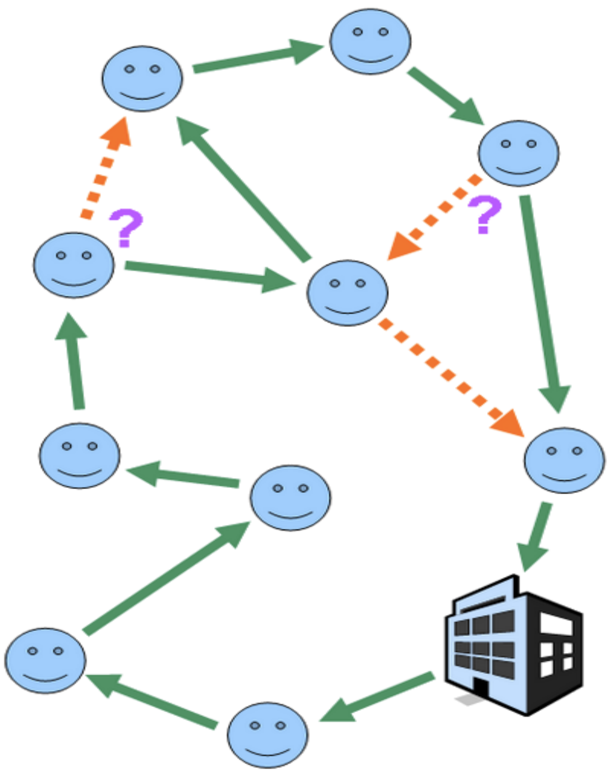
Why is an integer programming problem hard?

Vehicle Routing Problem – More Sites

www.mjc2.com/vehicle-routing-problem-3.htm

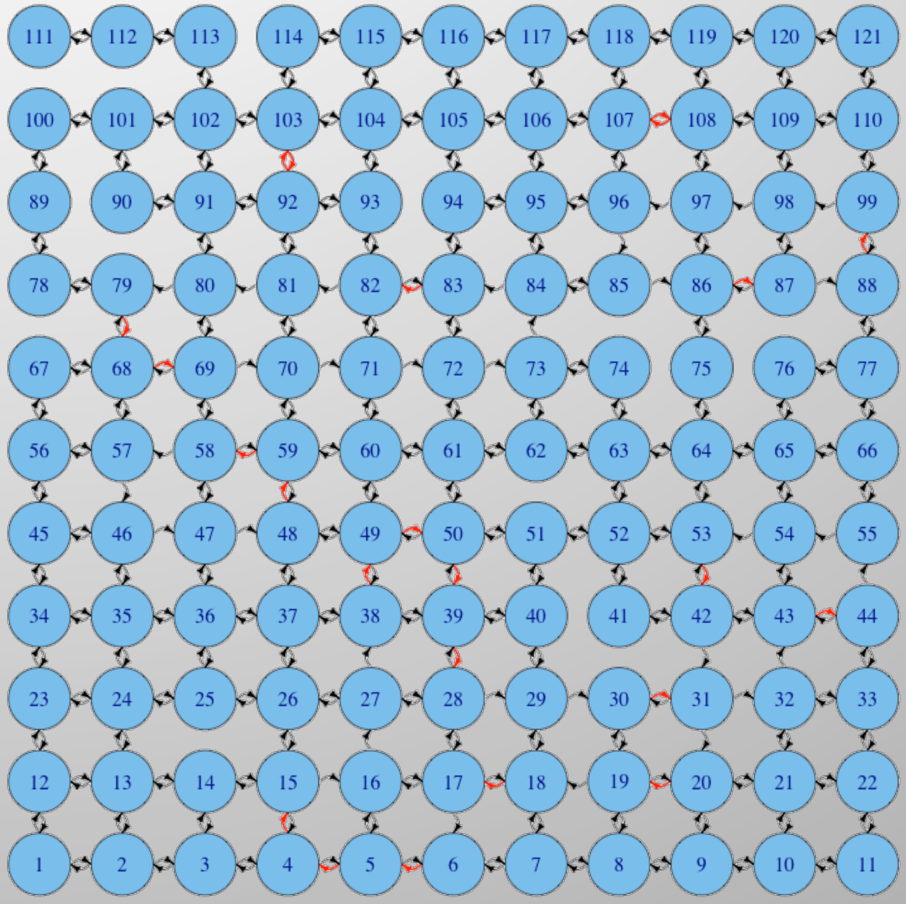
The number of possible routes increases VERY FAST as the number of customers increases (see below). Even for 10 customers there are ~3.6 **million** options. In theory you would need to look at each one and choose the best.

Customers	Number of Routes
3	6
4	24
5	120
6	720
7	8,040
8	40,320
9	362,880
10	3,628,800



WAE model properties

- Solves quickly without the sub tour elimination constraints.
- Extending to multiple vehicles
  - Empirically seems to solve quickly
- Given a set of arc in the solution, a closed Eulerian trial can easily be constructed since model guarantees all nodes have even degree
- Heuristic approach is to solve for only one vehicle at a time and successively remove arcs in the solution (such approach is possible because the graph is Eulerian)



Heuristic for the HIL Optimizer

Step 0: find a feasible solution

start from the zero vector and

select a nonnegative vector as direction

Step 1: obtain new descending direction **d**

compute the gradient at the current solution

*solve an **LP** accounting for minimum time constraints*

Step 2: check feasibility of **d**

Step 3: if **d** is feasible, find new solution

Step 4: if **d** is not a feasible direction, return to Step 1

Step 5: check stop criteria (*time and/or cost*), return to step 1 if needed

Heuristic for the HIL Optimizer

Step 0: find a feasible solution

start from the zero vector and

select a nonnegative vector as direction

Heuristic 1

Step 1: obtain new descending direction **d**

compute the gradient at the current solution

*solve an **LP** accounting for minimum time constraints*

Heuristic 2

Step 2: check feasibility of **d**

Step 3: if **d** is feasible, find new solution

Step 4: if **d** is not a feasible direction, return to Step 1

Step 5: check stop criteria (*time and/or cost*), return to step 1 if needed

Heuristic for the HIL Optimizer

<u>Step 0</u>: find a feasible solution

start from the zero vector and

select a nonnegative vector as direction

<u>Step 1</u>: obtain new descending direction **d**

compute the gradient at the current solution

solve an ***LP*** *accounting for minimum time constraints*

LP solver

<u>Step 2</u>: check feasibility of **d**

<u>Step 3</u>: if **d** is feasible, find new solution

<u>Step 4</u>: if **d** is not a feasible direction, return to Step 1

<u>Step 5</u>: check stop criteria (*time and/or cost*), return to step 1 if needed

Heuristic 1 for the HIL Optimizer

First compute a feasible solution

start from the infeasible solution

choose a direction based on attenuation values

Set current direction to a negative vector

Compute **step size (line search)**

1. If **step size > 0** then direction is feasible

Compute
new solution

Heuristic 1 for the HIL Optimizer

First compute a feasible solution

start from the infeasible solution

choose a direction based on attenuation values

Set current direction to a negative vector

Compute **new solution**

Compute **step size (line search)**

1. If **step size > 0** then direction is feasible

Heuristic 1 for the HIL Optimizer

First compute a feasible solution

start from the infeasible solution

choose a direction based on attenuation values

Set current direction to a negative vector

Set to zero coordinates that are binding

Compute **step size (line search)**

1. If **step size > 0** then direction is feasible
2. If **step size = 0** then

Heuristic 2

Compute
new solution

Try a different direction

Heuristic 2 for the HIL Optimizer

Compute **direction of descent**

- compute the gradient of the current solution
- **<u>solve an LP</u>** that guarantees descent (first order)

Compute **new solution**

Compute step size (**line search**)

1) If **step size > 0** then **direction is feasible**

Heuristic 2 for the HIL Optimizer

Compute **direction of descent**

- compute the gradient of the current solution
- **solve an LP** that guarantees descent (first order)

Compute **new solution**

Compute step size (**line search**)

1) If **step size > 0** then **direction is feasible**
2) If **step size = 0** then try different directions
   - solve LP again (not to optimality)

Heuristic 1

STOP

Questions, Comments?