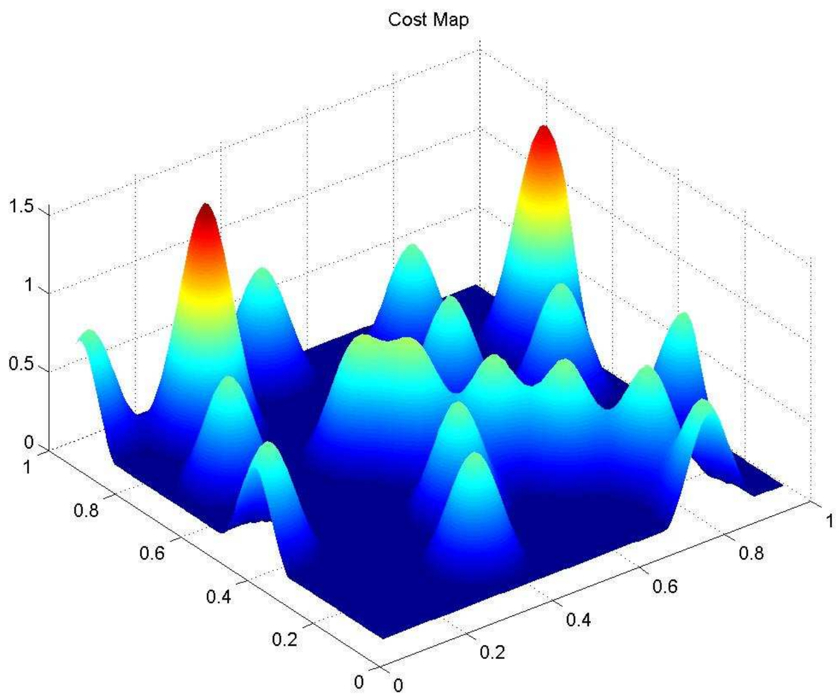
Path Planning with Fast Marching Methods

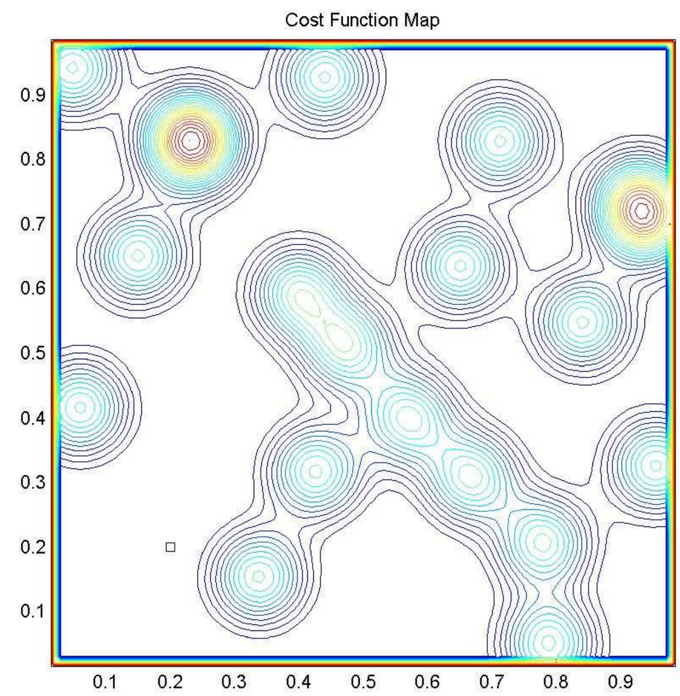
Ian Mitchell
Department of Computer Science
The University of British Columbia

research supported by
National Science and Engineering Research Council of Canada
ONR Computational Methods for Collaborative Control MURI (N00014-02-1-0729)

Basic Path Planning

- Find the optimal path $p(s)$ to a target (or from a source)
- Inputs
  - Cost to pass through each state in the state space
  - Set of targets or sources (provides boundary conditions)





Dynamic Programming Principle

$$V(x) = \min_{y \in N(x)} [V(y) + c(y \to x)]$$

- Value function $V(x)$ is "cost to go" from x to the nearest target
- $V(x)$ at a point x is the minimum over all points y in the neighborhood $N(x)$ of the sum of
  - the cost $V(y)$ at point y
  - the cost $c(y \to x)$ to travel from y to x
- Dynamic programming applies if
  - Costs are additive
  - Subsets of feasible paths are themselves feasible
  - Concatenations of feasible paths are feasible

Eikonal Equation

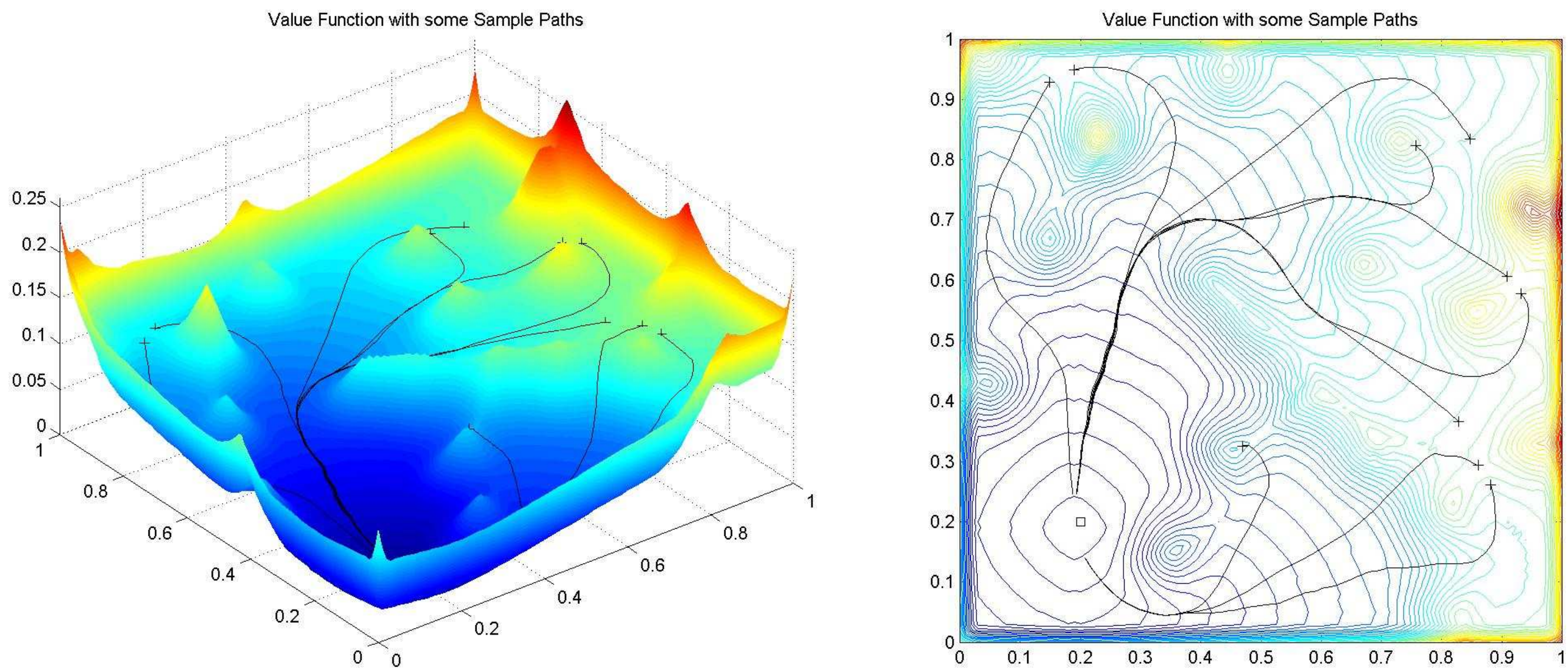
$$\|\nabla V(x)\| = c(x)$$

- Value function is viscosity solution of Eikonal equation
- Dynamic Programming Principle applies to Eikonal Equation
- Fast Marching Method: a continuous Dijkstra's algorithm
  - Node update equation is consistent with continuous PDE (and numerically stable)
  - Nodes are dynamically ordered so that each is visited a constant number of times

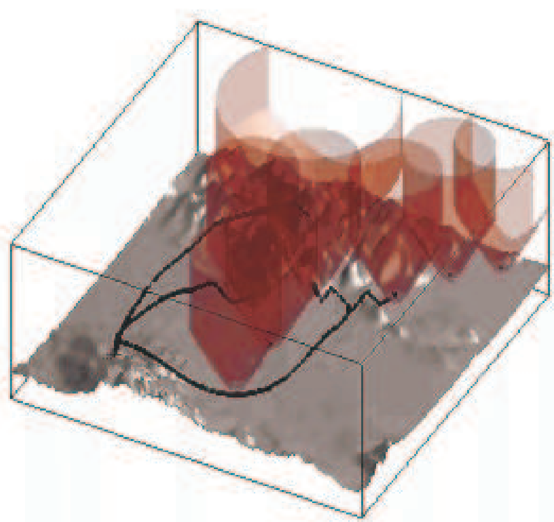
Path Generation

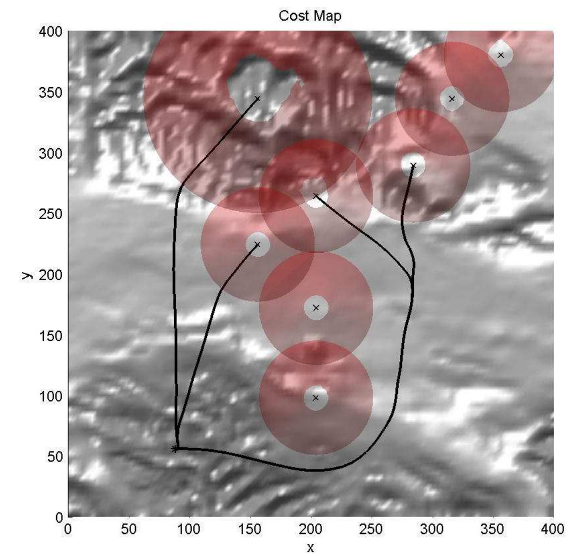
- Optimal path $p(s)$ is found by gradient descent
  - Value function $V(x)$ has no local minima, so paths will always terminate at a target

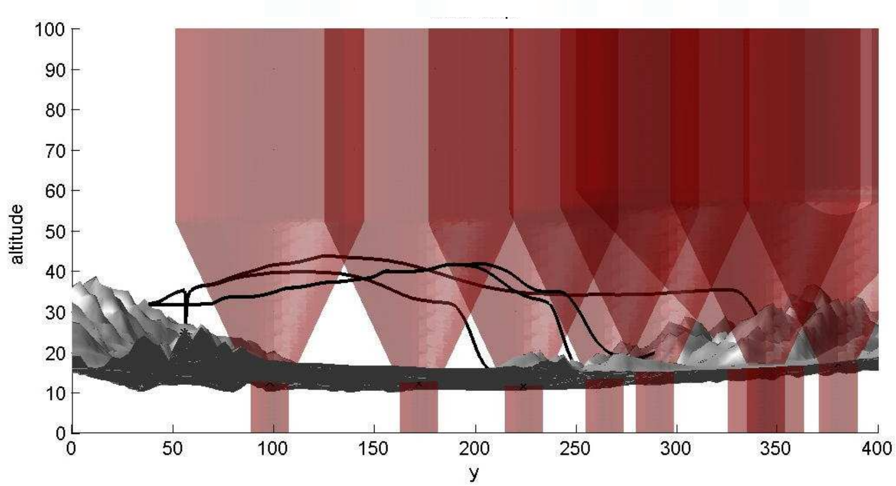
$$\frac{dp}{ds} = \frac{\nabla V(x)}{\|\nabla V(x)\|}$$



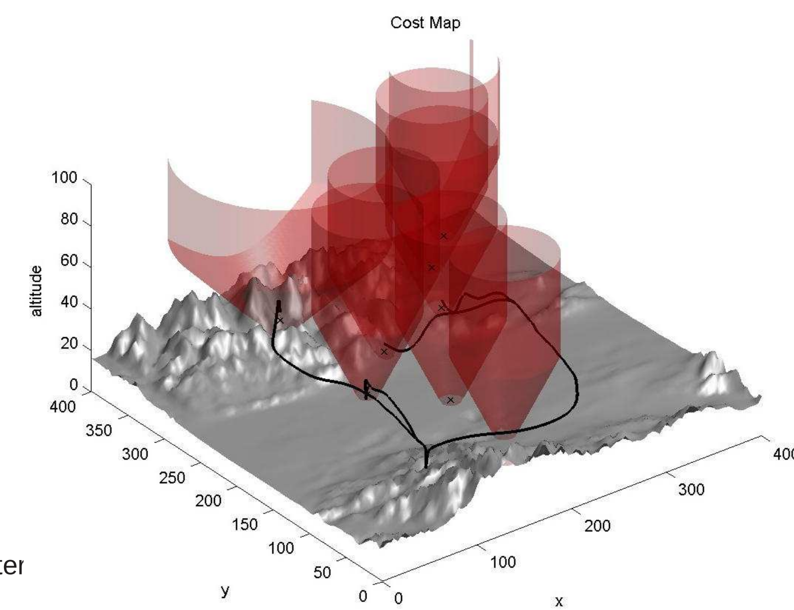
Demanding Example? No!







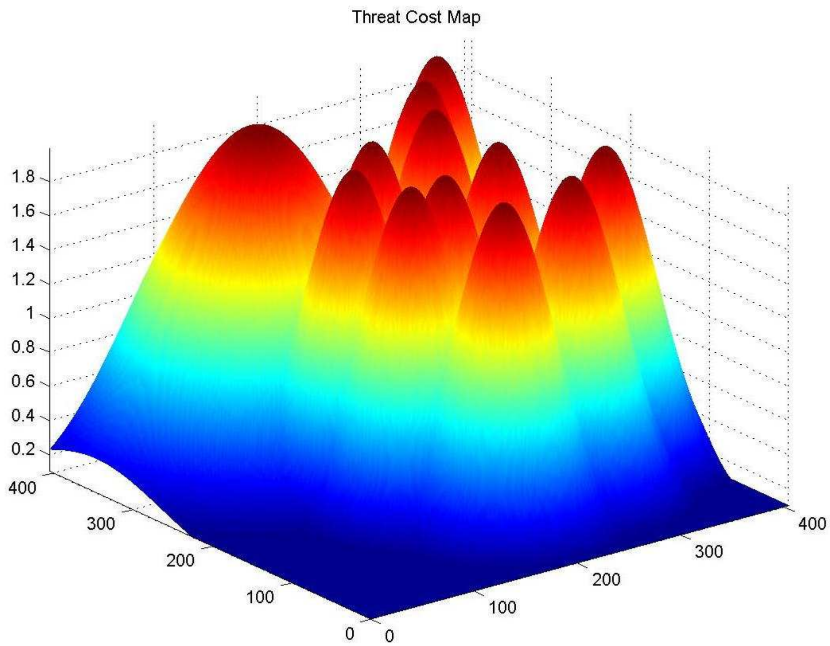
C Computer



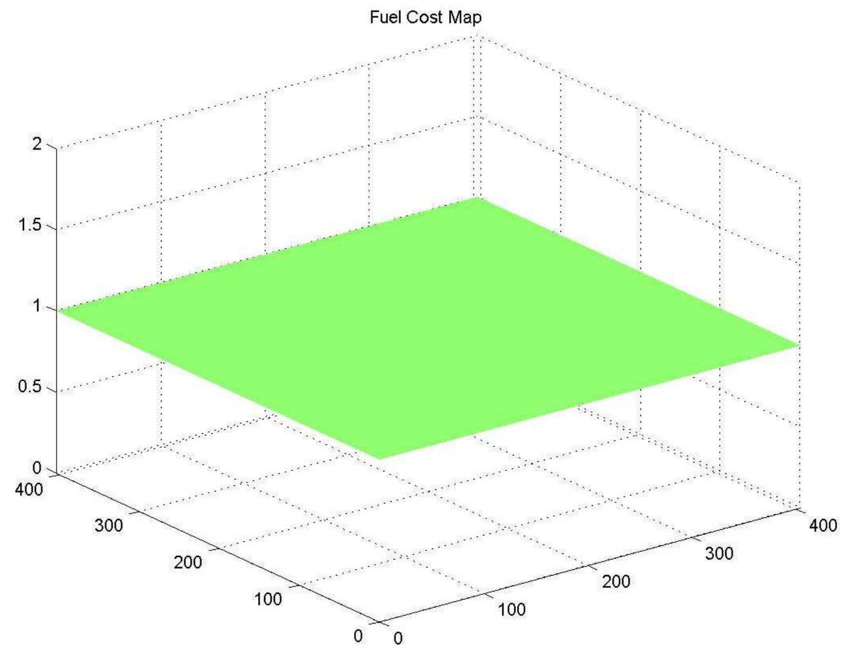
Constrained Path Planning

- Input includes multiple cost functions $c_i(x)$
- Possible goals:
  - Find feasible paths given bounds on each cost
  - Optimize one cost subject to bounds on the others
  - Given a feasible/optimal path, determine marginals of the constraining costs

Variable cost (eg threat level)



Constant cost (eg fuel)



Path Integrals

- To determine if path $p(t)$ is feasible, we must determine

$$P_i(x) = \int_0^T c_i(p(s))ds, \text{ where } \begin{cases} p(0) = \texttt{target}, \\ p(T) = x \end{cases}$$

- If the path is generated from a value function $V(x)$, then path integrals can be computed by solving the PDE

$$\nabla P_i(x) \cdot \nabla V(x) = c_i(x)c(x)$$

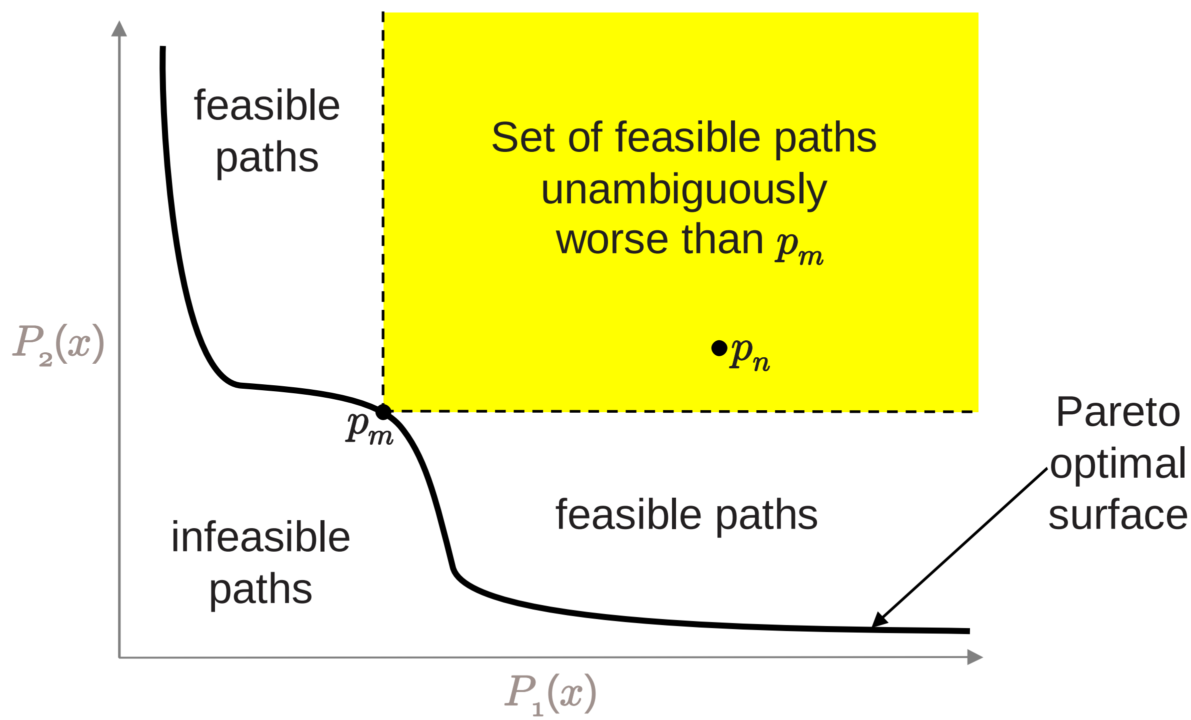
- The computation of the $P_i(x)$ can be integrated into the FMM algorithm that computes $V(x)$

Pareto Optimality

- Consider a single point x and a set of costs $c_i(x)$
- Path p_m is unambiguously better than path p_n if

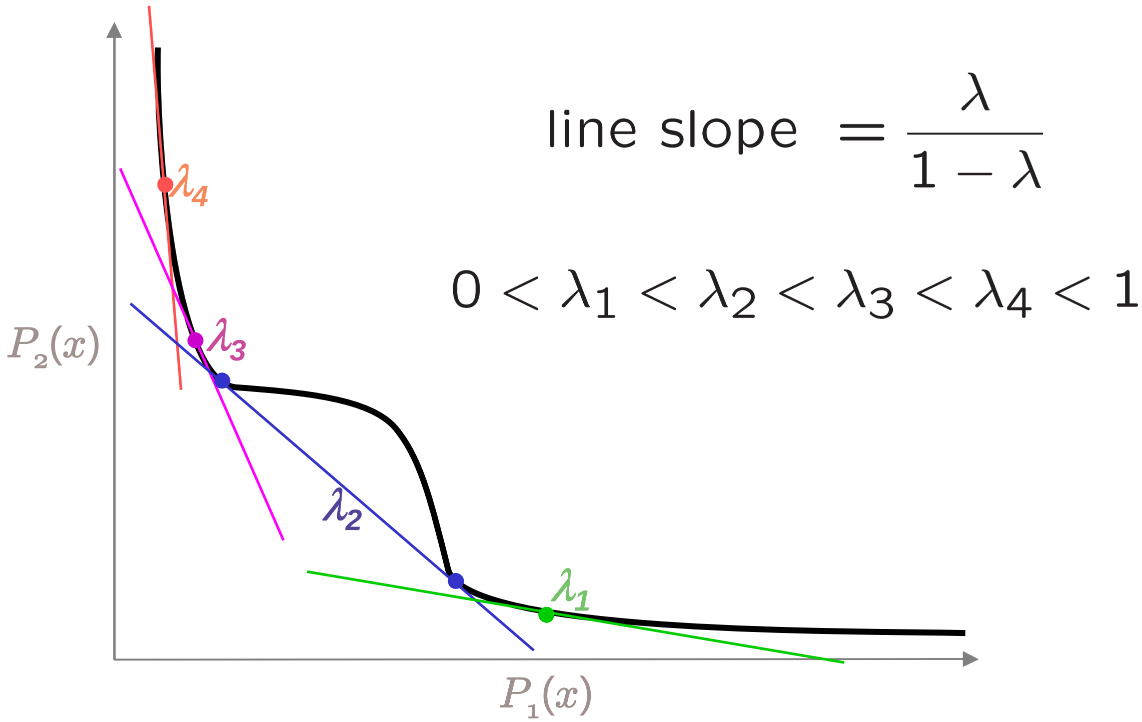
$$P_i(x; p_m) \leq P_i(x; p_n) \text{ for all } i$$

- Pareto optimal surface is the set of all paths for which there are no other paths that are unambiguously better



Exploring the Pareto Surface

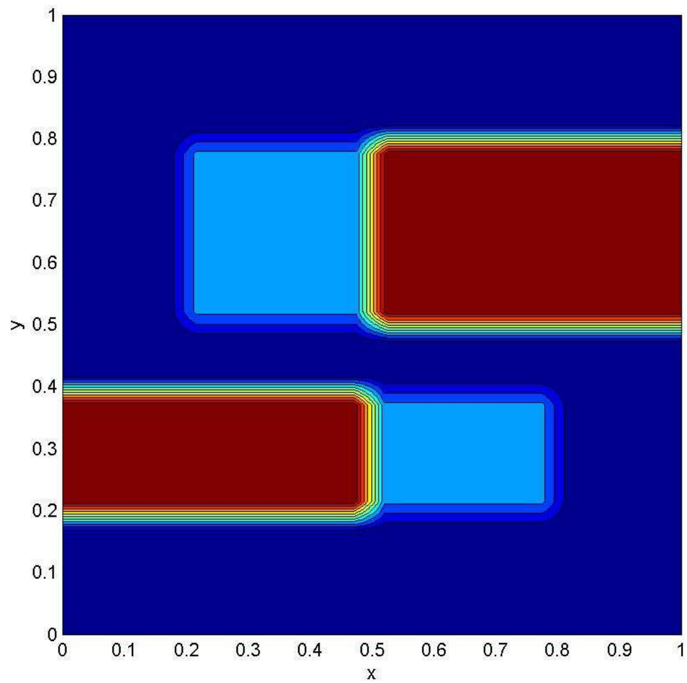
- Compute value function for a convex combination of cost functions
  - For example, let $c(x) = \lambda c_1(x) + (1 - \lambda)c_2(x)$, $\lambda \in [\,0,1\,]$
- Use FMM to compute corresponding $V(x)$ and $P_i(x)$
- Constructs a convex approximation of the Pareto surface for each point x in the state space

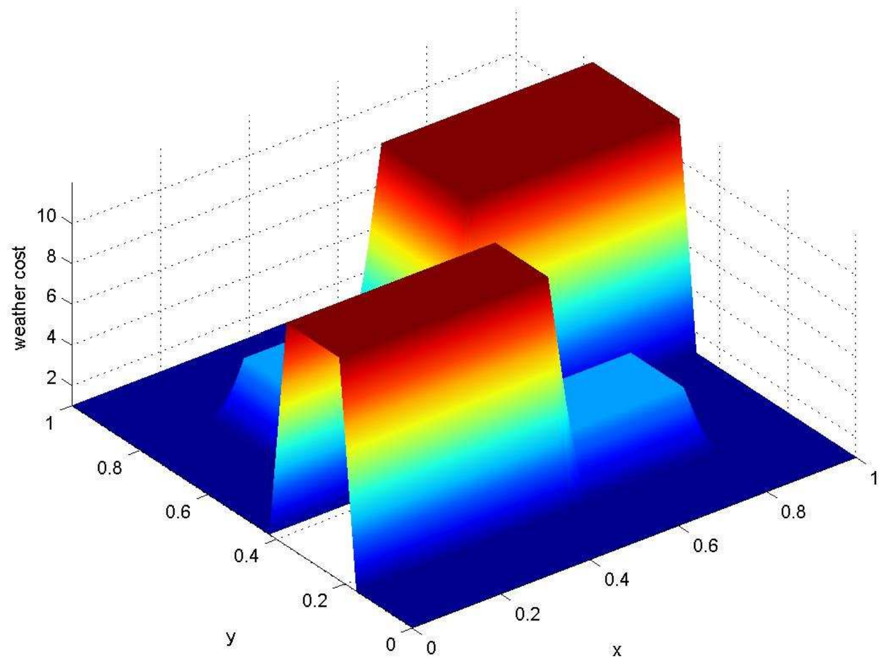


Constrained Path Planning Example

- Plan a path across Squaraguay
  - From Lowerleftville to Upper Right City
  - Costs are fuel (constant) and threat of a storm

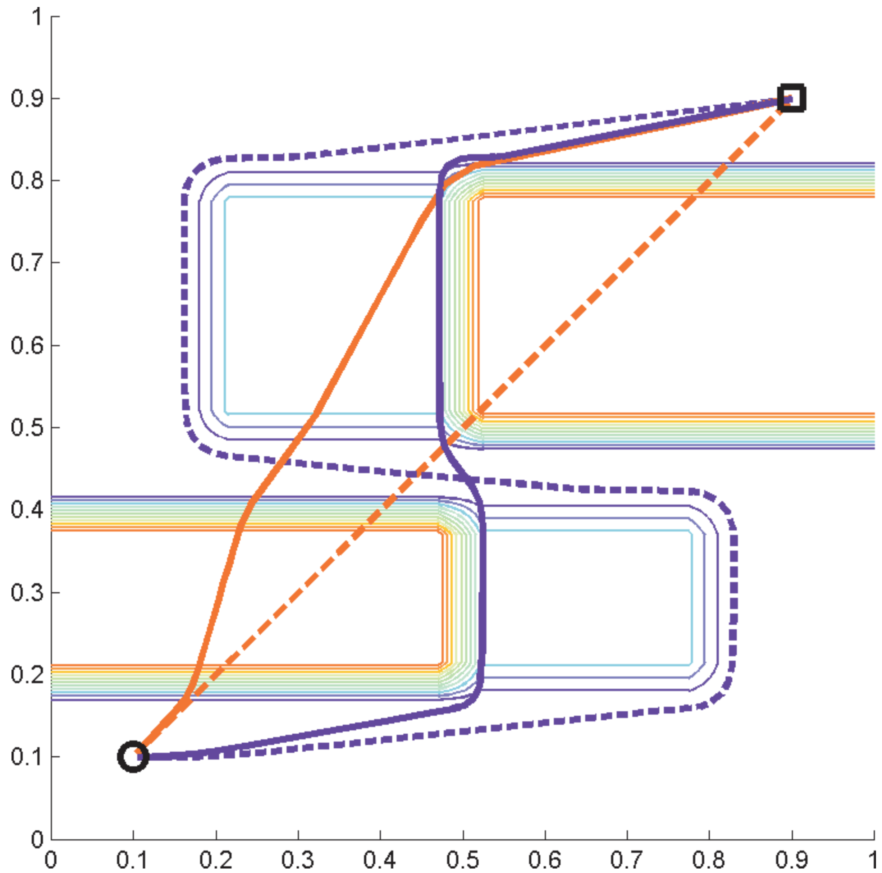
Weather cost (two views)





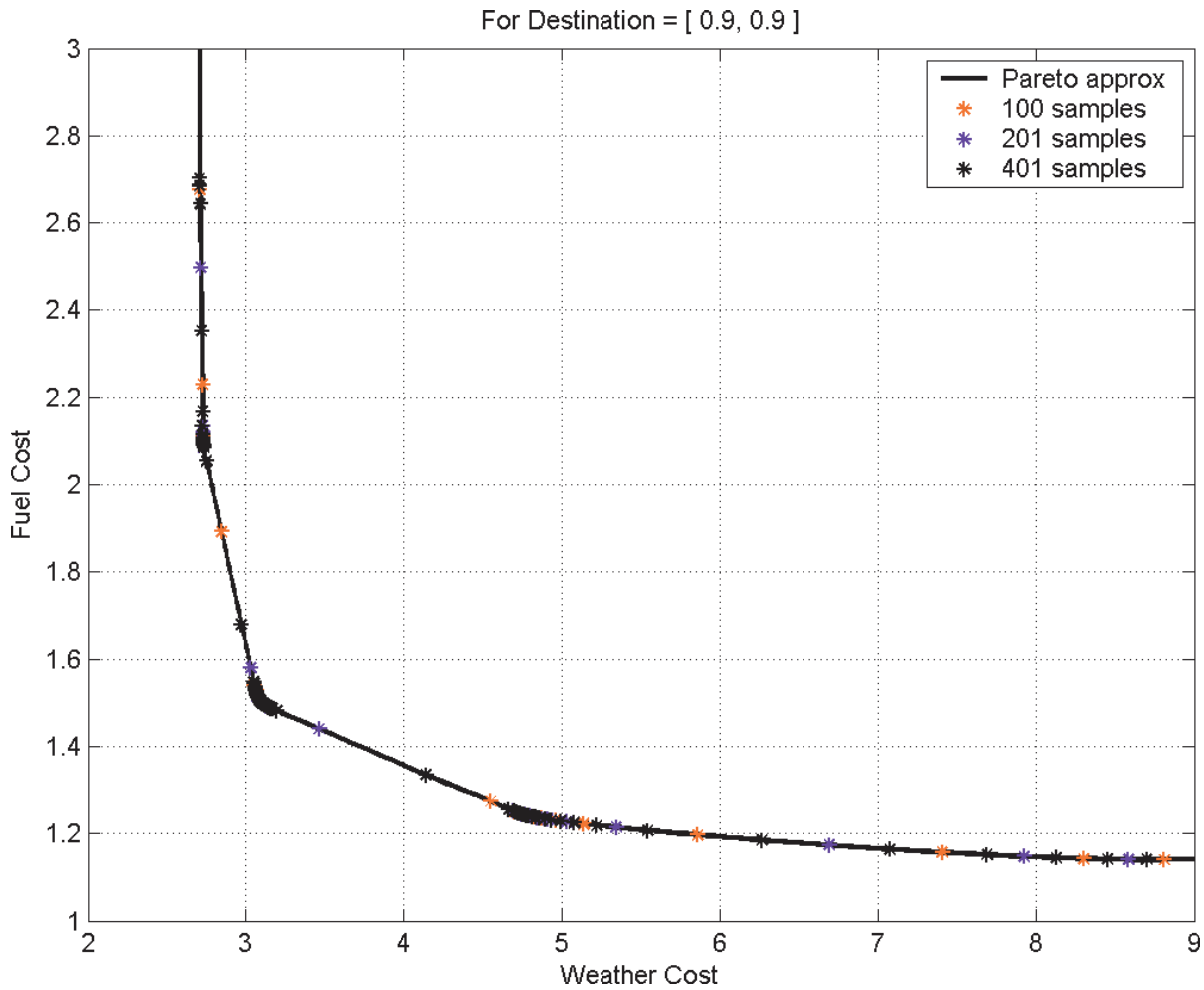
Weather and Fuel Constrained Paths

line type	minimize what?	fuel constraint	fuel cost	weather cost
- - - - - (orange dashed)	fuel	none	1.14	8.81
——— (orange solid)	weather	1.3	1.27	4.55
——— (purple solid)	weather	1.6	1.58	3.03
- - - - - (purple dashed)	weather	none	2.69	2.71



Pareto Optimal Approximation

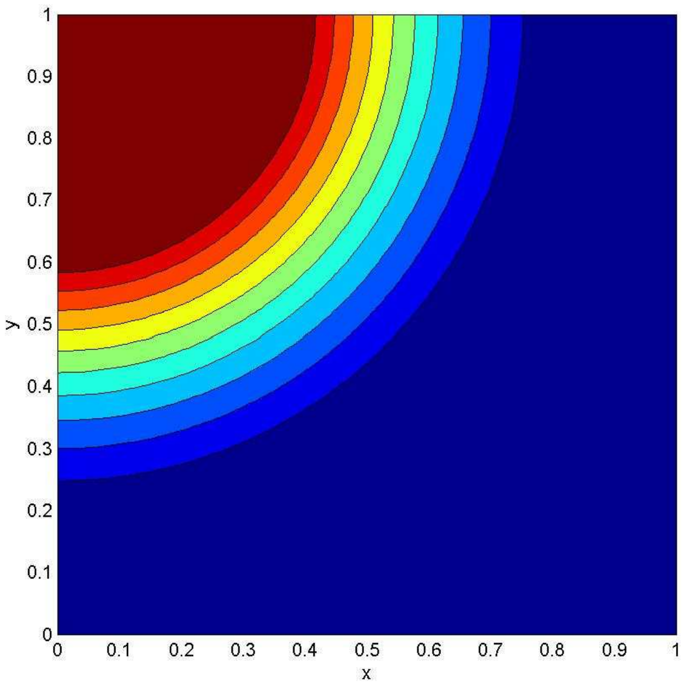
- Cost depends linearly on number of sample λ values
  - For 201^2 grid and 401 λ samples, execution time 53 seconds

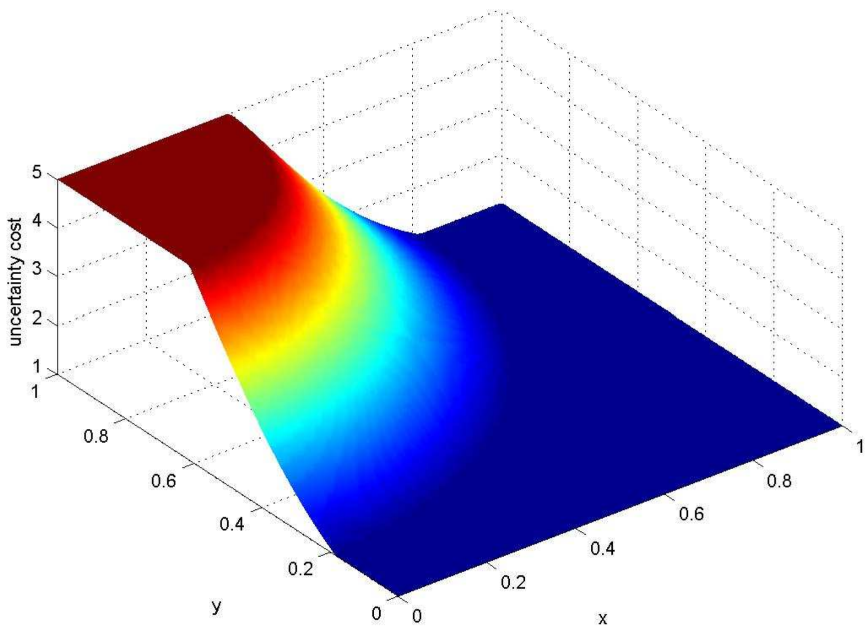


More Constraints

- Plan a path across Squaraguay
  - From Lowerleftville to Upper Right City
  - There are no weather stations in northwest Squaraguay
  - Third cost function is uncertainty in weather

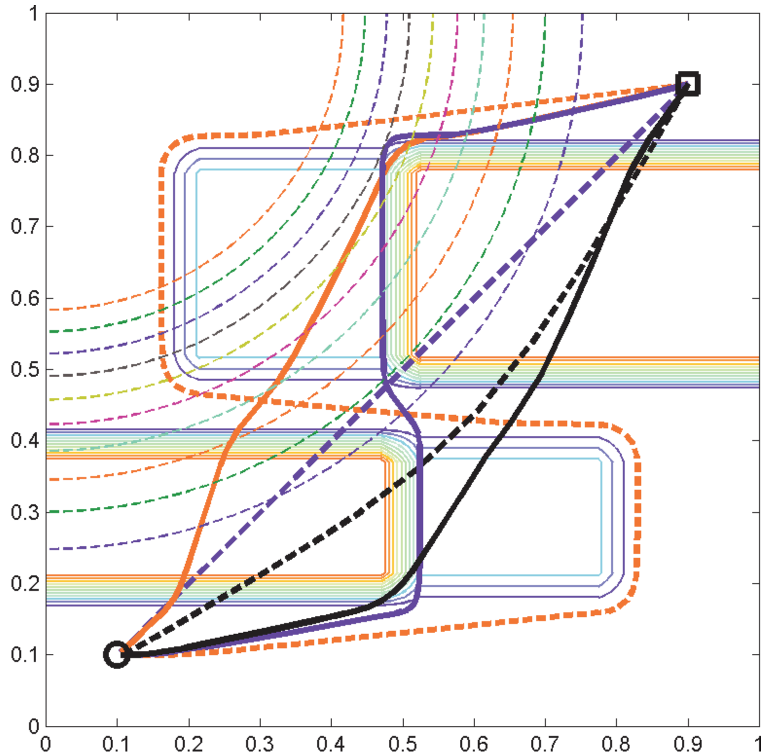
Uncertainty cost (two views)





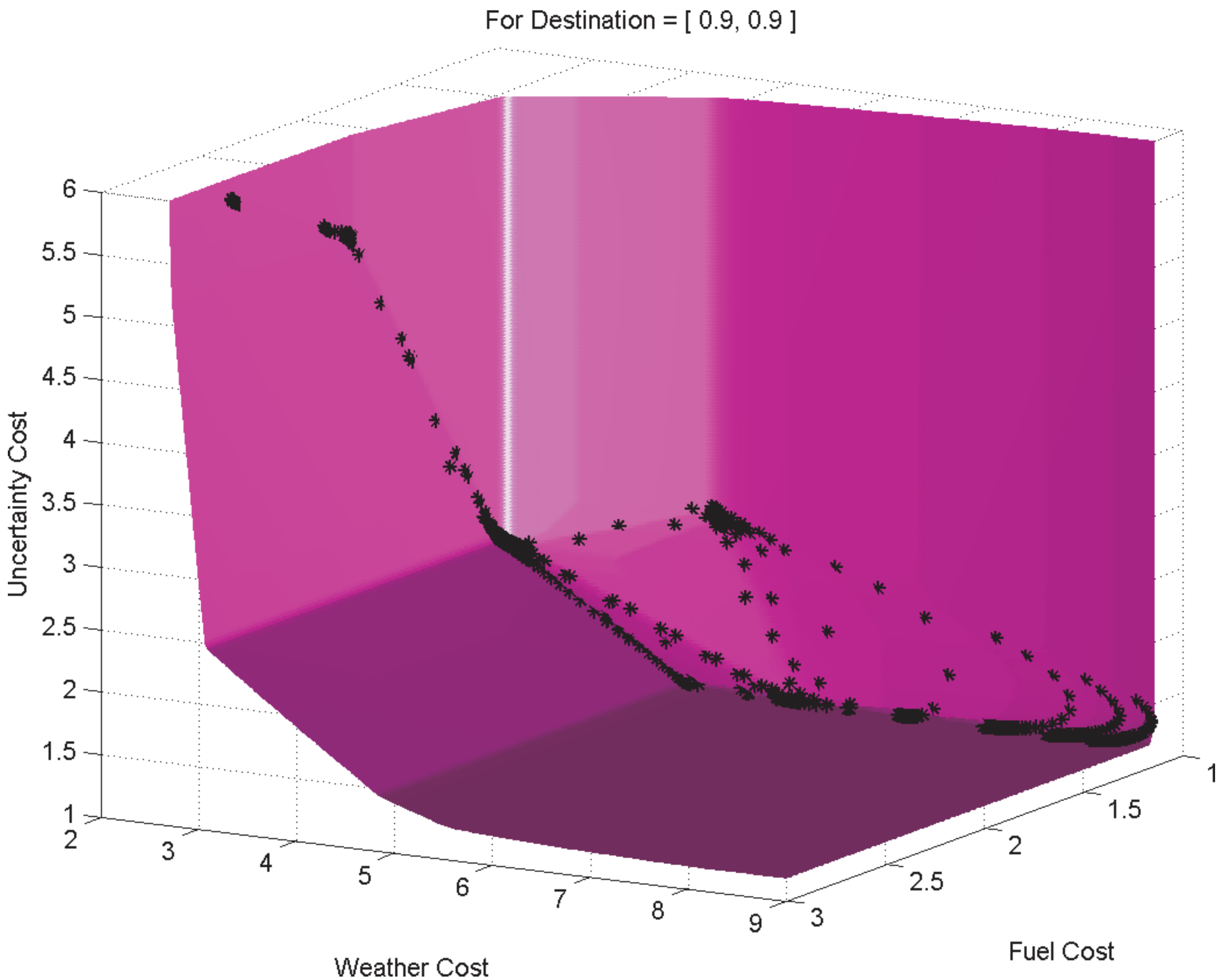
Three Costs

line type	minimize what?	fuel constraint	weather constraint	fuel cost	weather cost	uncertainty cost
- - - - - (purple)	fuel	none	none	1.14	8.81	1.50
- - - - - (orange)	weather	none	none	2.69	2.71	5.83
- - - - - (black)	uncertainty	none	none	1.17	8.41	1.17
——— (purple)	weather	1.6	none	1.60	3.02	2.84
——— (orange)	weather	1.3	none	1.30	4.42	2.58
——— (black)	uncertainty	1.3	6.0	1.23	5.84	1.23



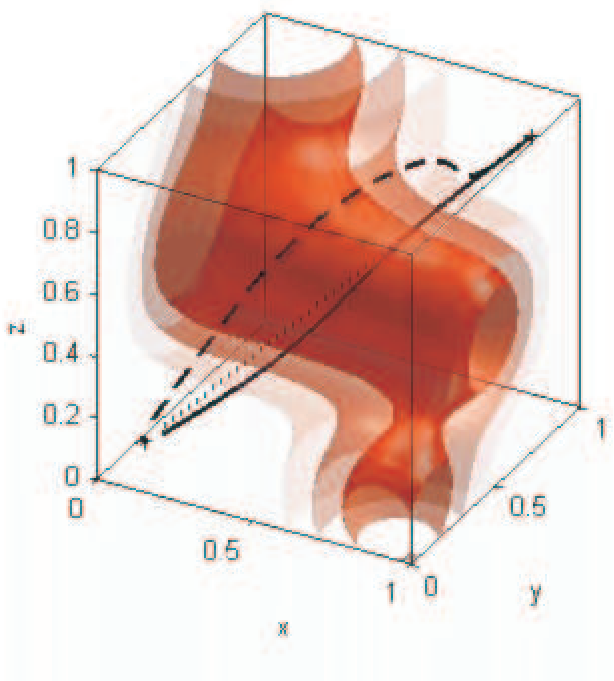
Pareto Surface Approximation

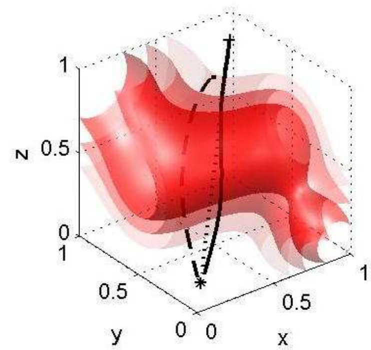
- Cost depends linearly on number of sample λ values
  - For 201^2 grid and 101^2 λ samples, execution time 13 minutes

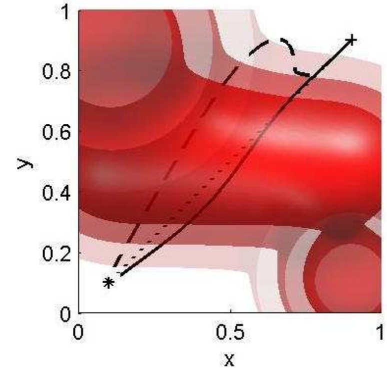


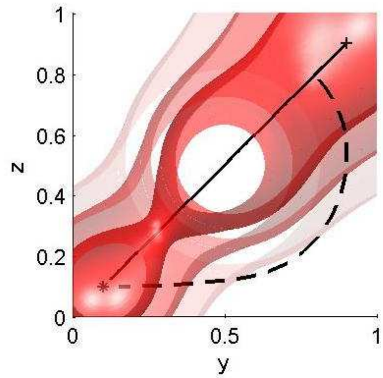
Three Dimensions

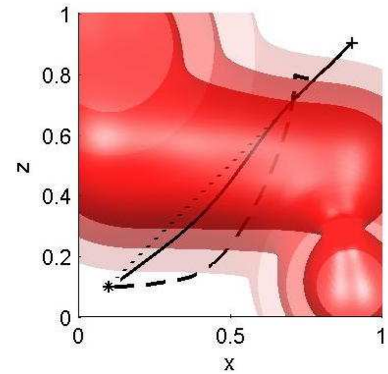
line type	minimize what?	fuel constraint	fuel cost	weather cost
- - - - -	fuel	none	1.14	3.54
— — —	weather	none	1.64	1.64
———	weather	1.55	1.55	2.00





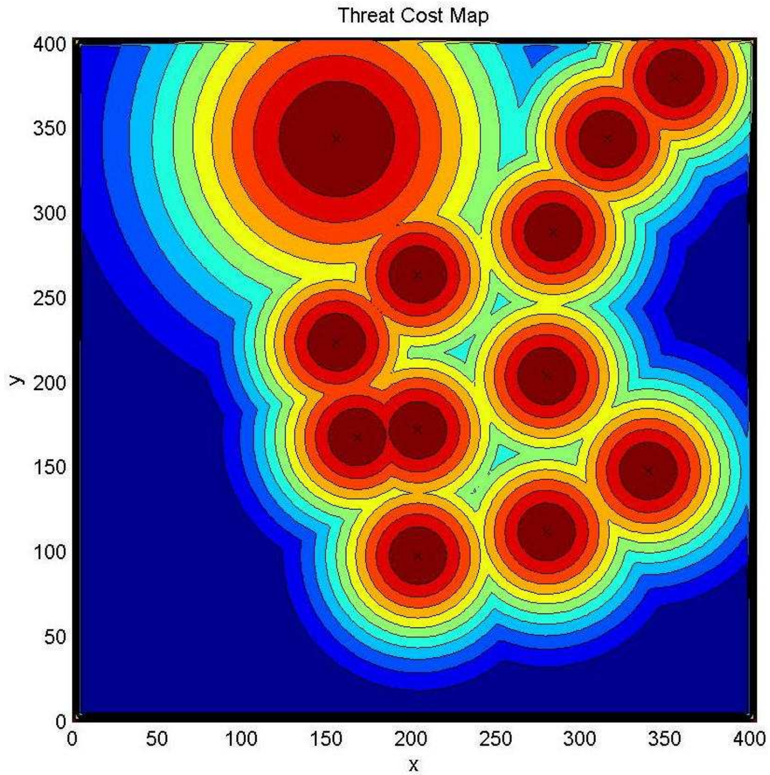




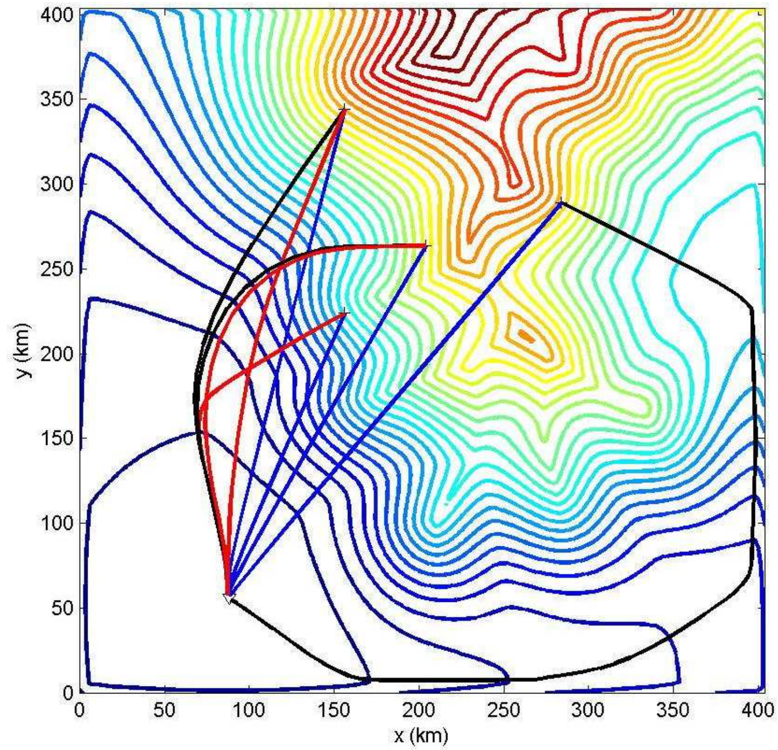


Constrained Example

- Plan path to selected sites
  - Threat cost function is maximum of individual threats
- For each target, plan 3 paths
  - minimum threat, minimum fuel, minimum threat (with fuel ≤ 300)



threat cost



Paths (on value function)

Fast Enough?

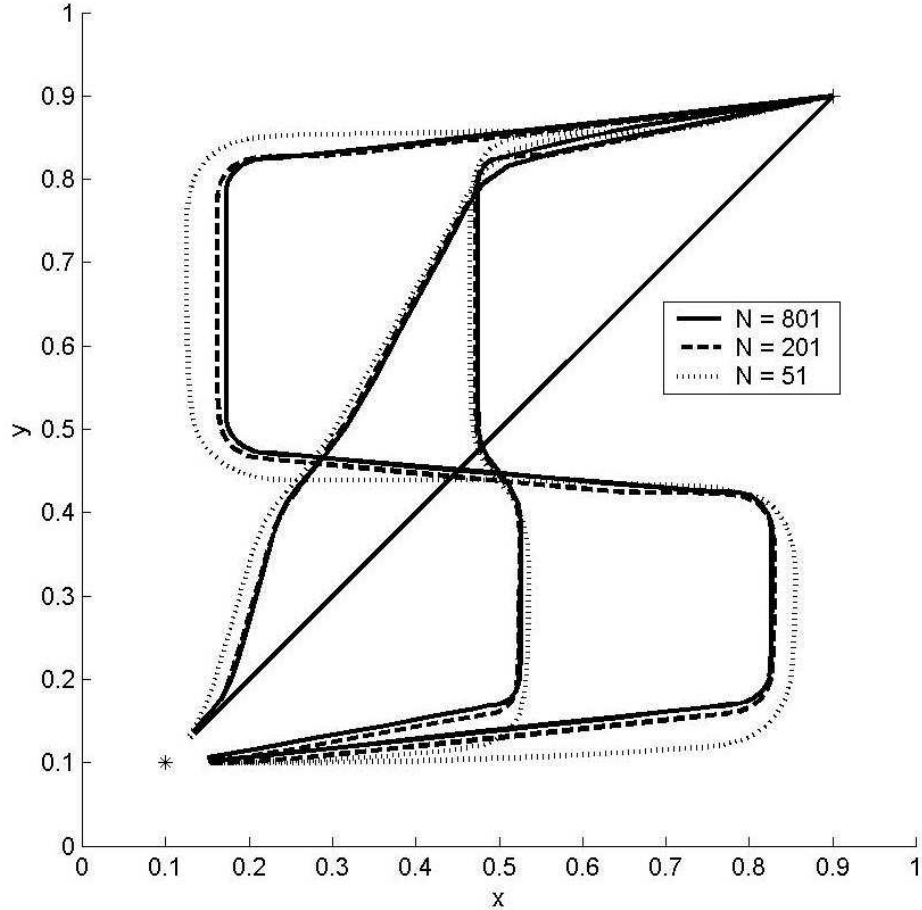
- Platform details
  - 2 GHz Mobile Pentium 4, 1 GB memory, Windows XP Pro
  - Value function by compiled C++
  - Path generation by interpreted m-file integration

Value Function (single objective)		
dim	grid size	time (s)
2	101^2	0.04
	201^2	0.10
	401^2	0.43
	801^2	1.87
	1601^2	9.33
3	51^3	0.90
	101^3	9.78
	201^3	94.91
4	51^4	166.76

Path Generation (25 random targets)			
dim	grid size	mean (s)	σ
2	101^2	0.57	0.32
	201^2	0.62	0.38
	401^2	0.72	0.51
	801^2	0.82	0.60
	1601^2	1.05	0.75
3	51^3	0.92	0.38
	101^3	0.89	0.49
	201^3	0.95	0.48
4	51^4	1.62	0.57

Grid Refinement

- As resolution improves, the approximation converges to the analytically optimal path for almost every destination point
  - little qualitative difference if cost function features are resolved



Path Generation Times

- Platform details
  - 2 GHz Mobile Pentium 4, 1 GB memory, Windows XP Pro
  - Value function by compiled C++
  - Path generation by interpreted m-file integration
  - Total cost includes cost function generation, PDE and ODE solves and plotting all the figures

2D cost per sample		
N	time (s) per λ	ratio
51	0.01	
101	0.04	3.24
201	0.13	3.76
401	0.55	4.20
801	2.44	4.41

3D cost per sample		
N	time (s) per λ	ratio
51	1.27	
101	12.66	9.99
201	125.46	9.91

Total cost for each example				
d	***k***	***N***	$\Delta\lambda$	time (m)
2	2	201	0.005	0.5
2	3	101	0.020	1.0
3	2	101	0.010	22.3