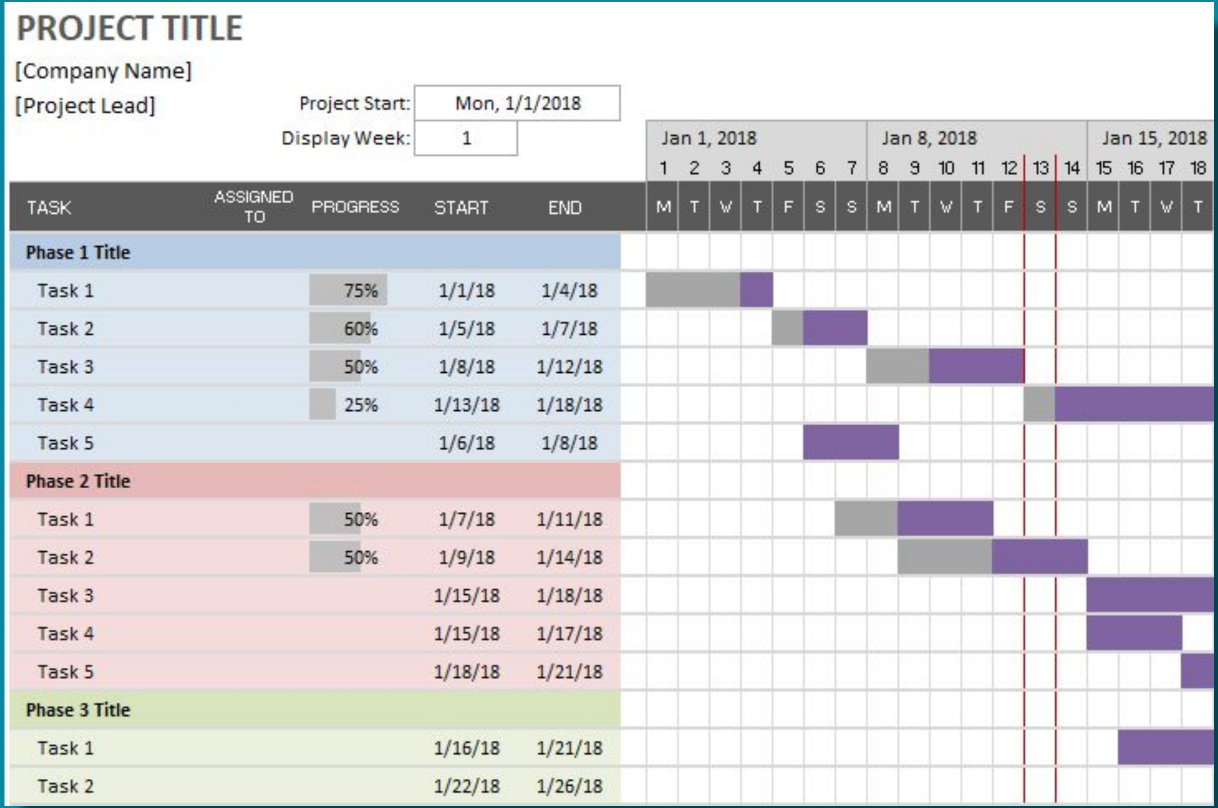
LEARNING TO CRASH:

A Reinforcement Learning Approach to Logistics

Logistics Planning



PROJECT TITLE

[Company Name]

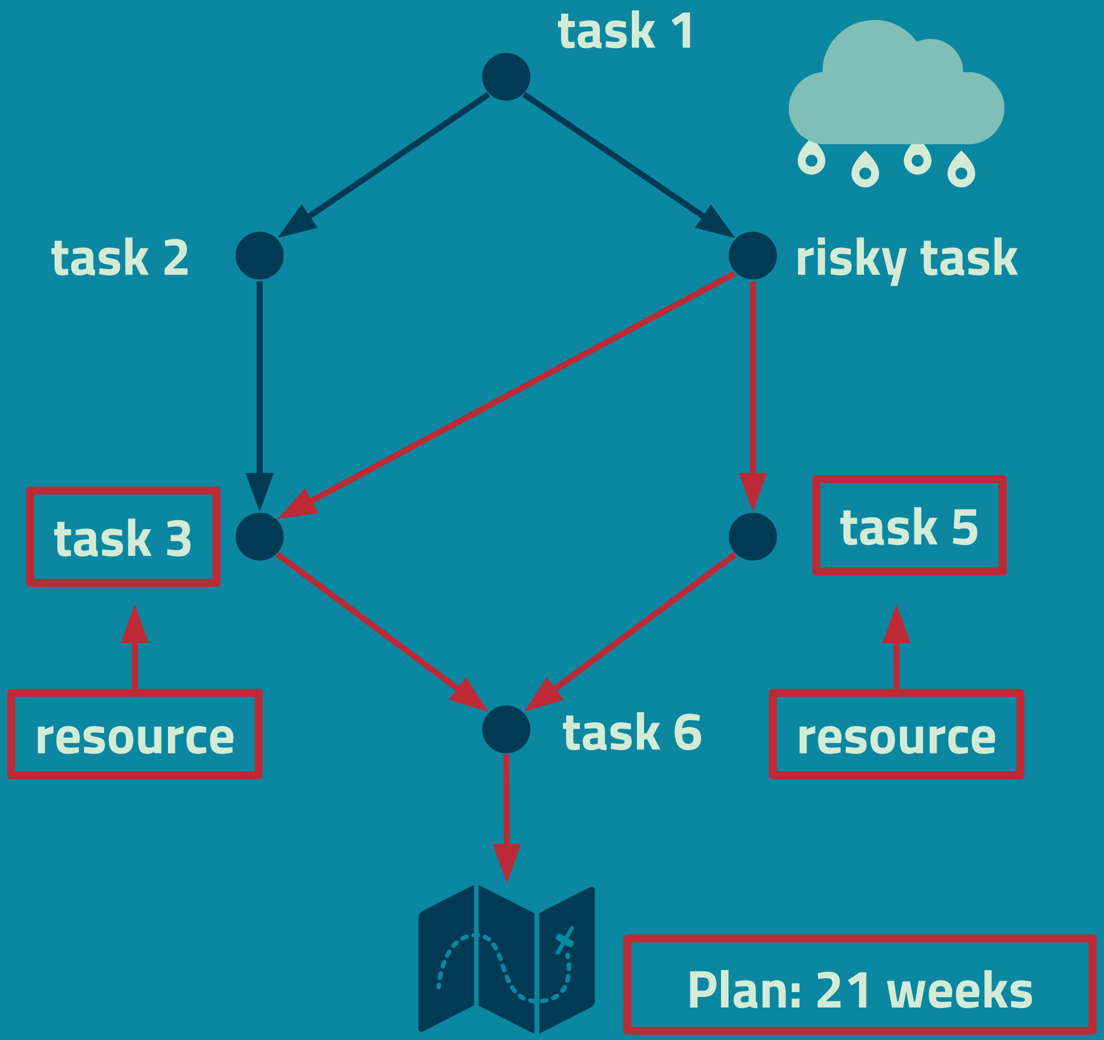
[Project Lead]

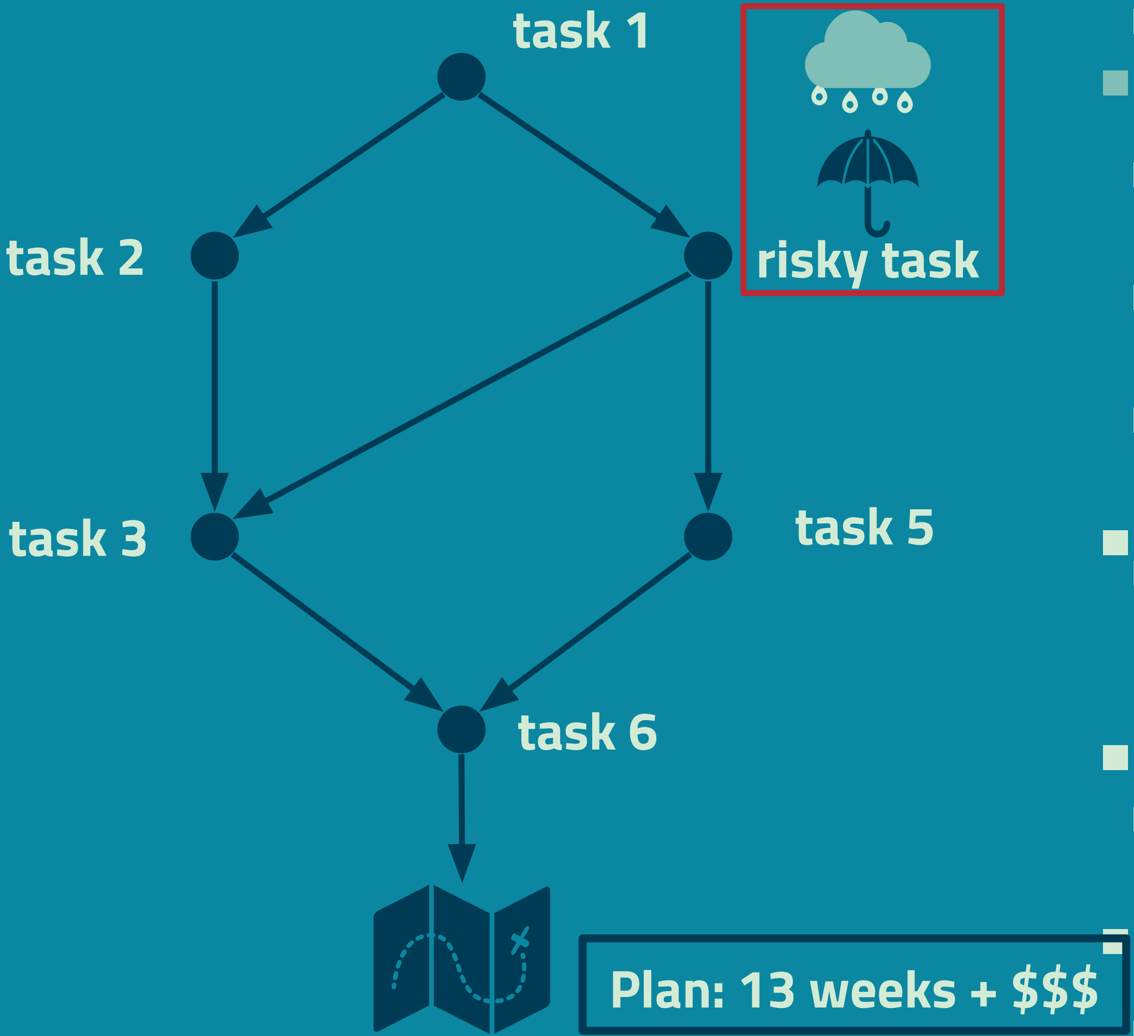
Project Start: Mon, 1/1/2018

Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 1 Title				
Task 1		75%	1/1/18	1/4/18
Task 2		60%	1/5/18	1/7/18
Task 3		50%	1/8/18	1/12/18
Task 4		25%	1/13/18	1/18/18
Task 5			1/6/18	1/8/18
Phase 2 Title				
Task 1		50%	1/7/18	1/11/18
Task 2		50%	1/9/18	1/14/18
Task 3			1/15/18	1/18/18
Task 4			1/15/18	1/17/18
Task 5			1/18/18	1/21/18
Phase 3 Title				
Task 1			1/16/18	1/21/18
Task 2			1/22/18	1/26/18

Plan: 13 weeks


task 1
risky task
task 2
task 3
task 5
resource
task 6
resource
Plan: 21 weeks


task 1
risky task
task 2
task 3
task 5
task 6
Plan: 13 weeks + $$$

Problem Statement

$$\min_{x_i \in \{0,1\}} \mathbb{E}_{\Omega}[\Lambda(x)] + \sum_{i=1}^{m} c_i x_i$$

Problem Statement

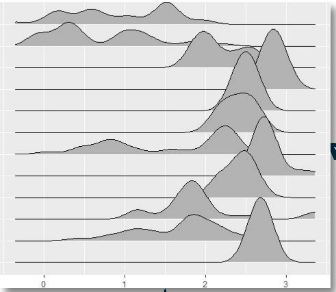
loss penalty **crash cost**

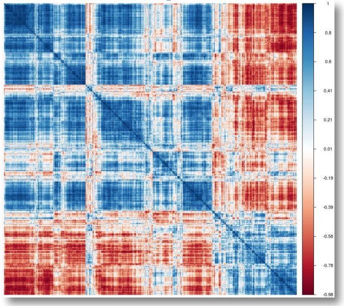
$$\min_{x_i \in \{0,1\}} \mathbb{E}_{\Omega}\left[\Lambda(x)\right] + \sum_{i=1}^{m} c_i x_i$$

crashing decisions **infinite scenarios**

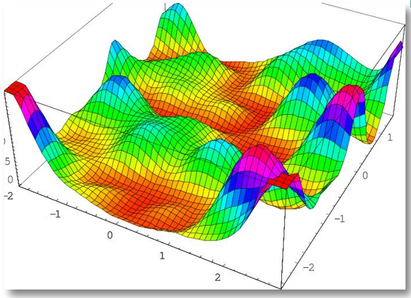
Data Components

Task Duration Distributions





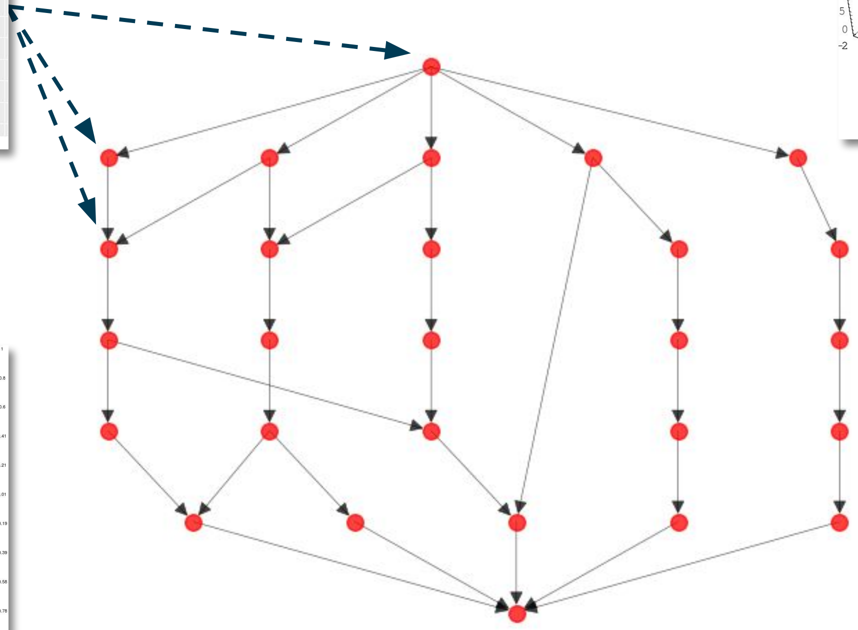
Covariances



Cost & Penalty Function

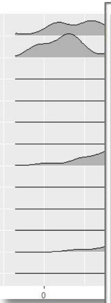
Activity	Description	Predecessor	Time (Days)
A	Clear the location	-	1
B	Bring the tool to the location	-	2
C	Conducting excavations	A	1
D	Pour the foundation	C	2
E	Plumbing installation from outside	B, C	6
F	Create a home frame	D	10
G	Installation of electric wire	F	3
H	Install the floor	G	1
I	Installing the roof	F	1
J	Installation of plumbing from inside	E, H	5
K	Installing the roof	I	2
L	Installation of the outer wall layers	F, J	1
M	Installing windows and outside doors	F	2
N	Build a wall	L, M	4
O	Isolate the walls and ceiling	G, J	2
P	Close the walls and ceiling	O	2
Q	Isolate the roof	I, P	1
R	Finish the interior	P	7
S	Finish the exterior	I, N	7
T	Build a garden	S	3

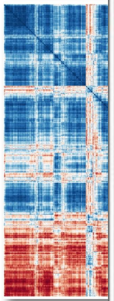
Crash Description

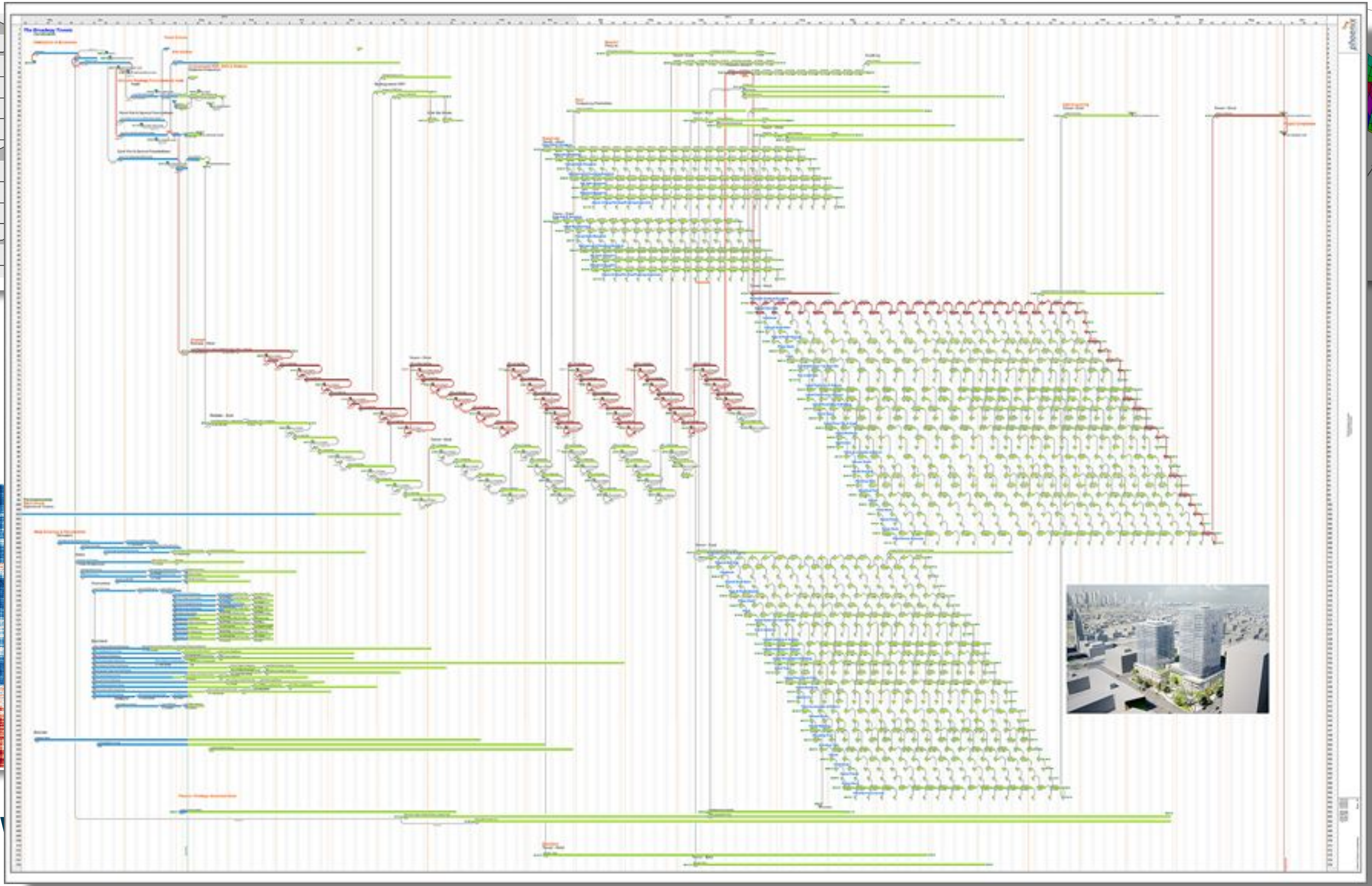


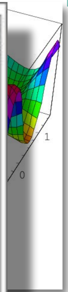
Task Duration Distributions

Data Components





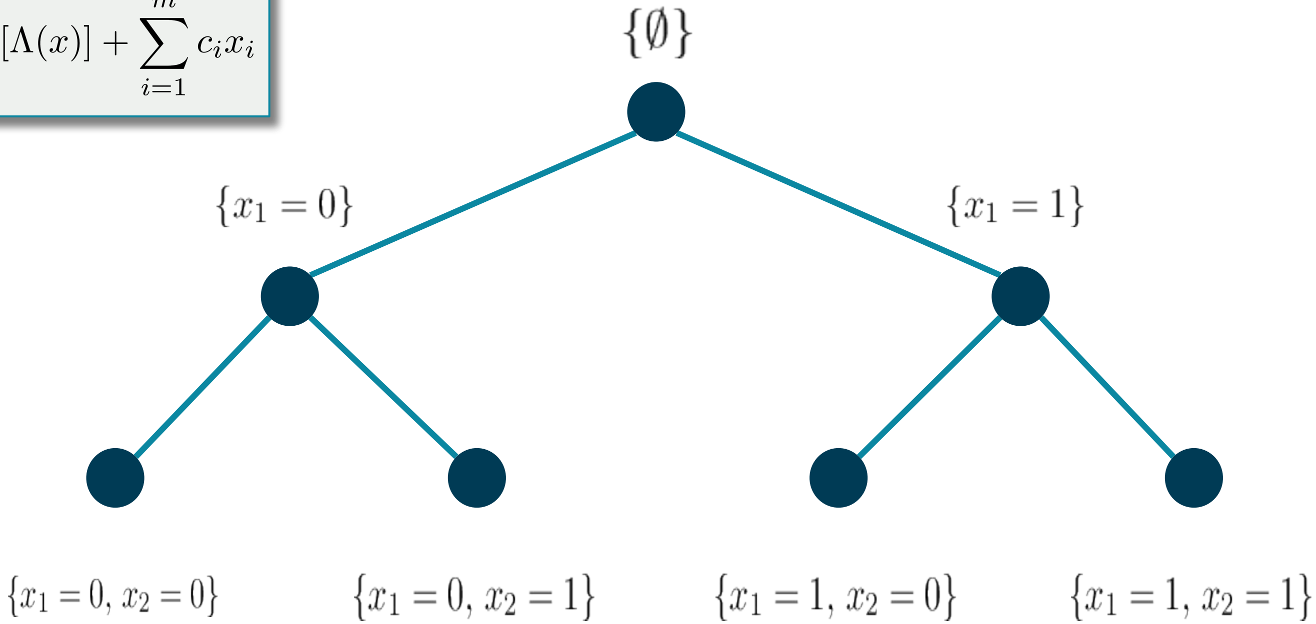




Co

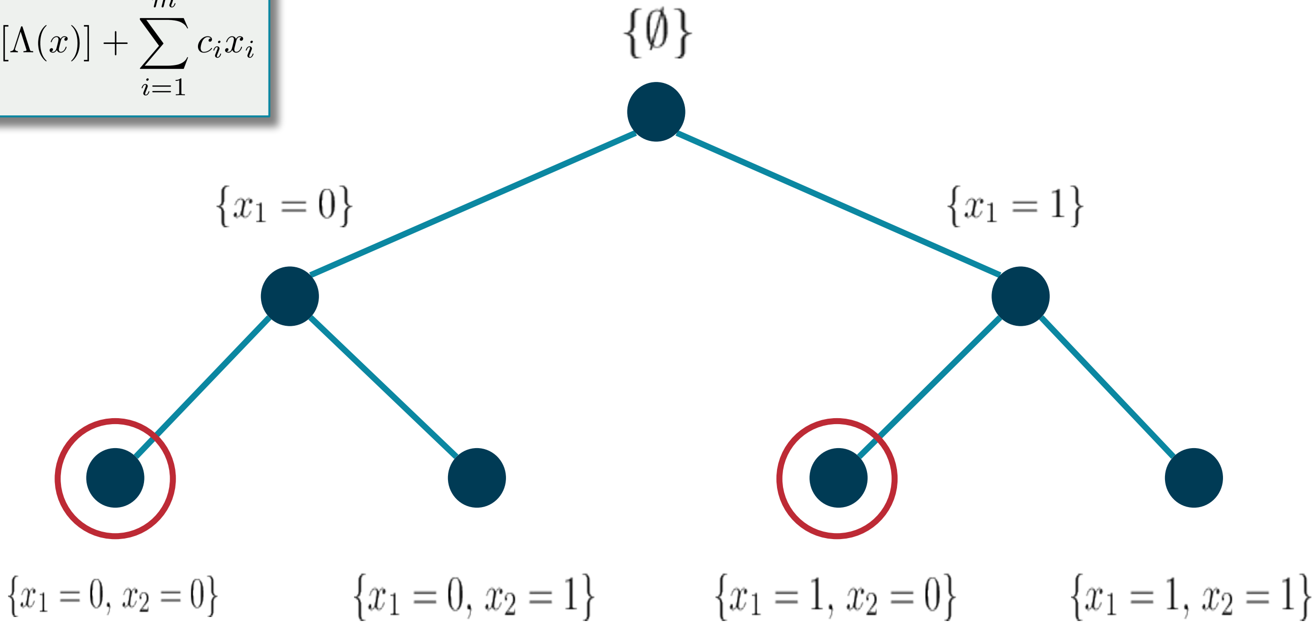
ML Approach to Crashing

$$\min_{x_i \in \{0,1\}} \mathbb{E}_\Omega[\Lambda(x)] + \sum_{i=1}^{m} c_i x_i$$



ML Approach to Crashing

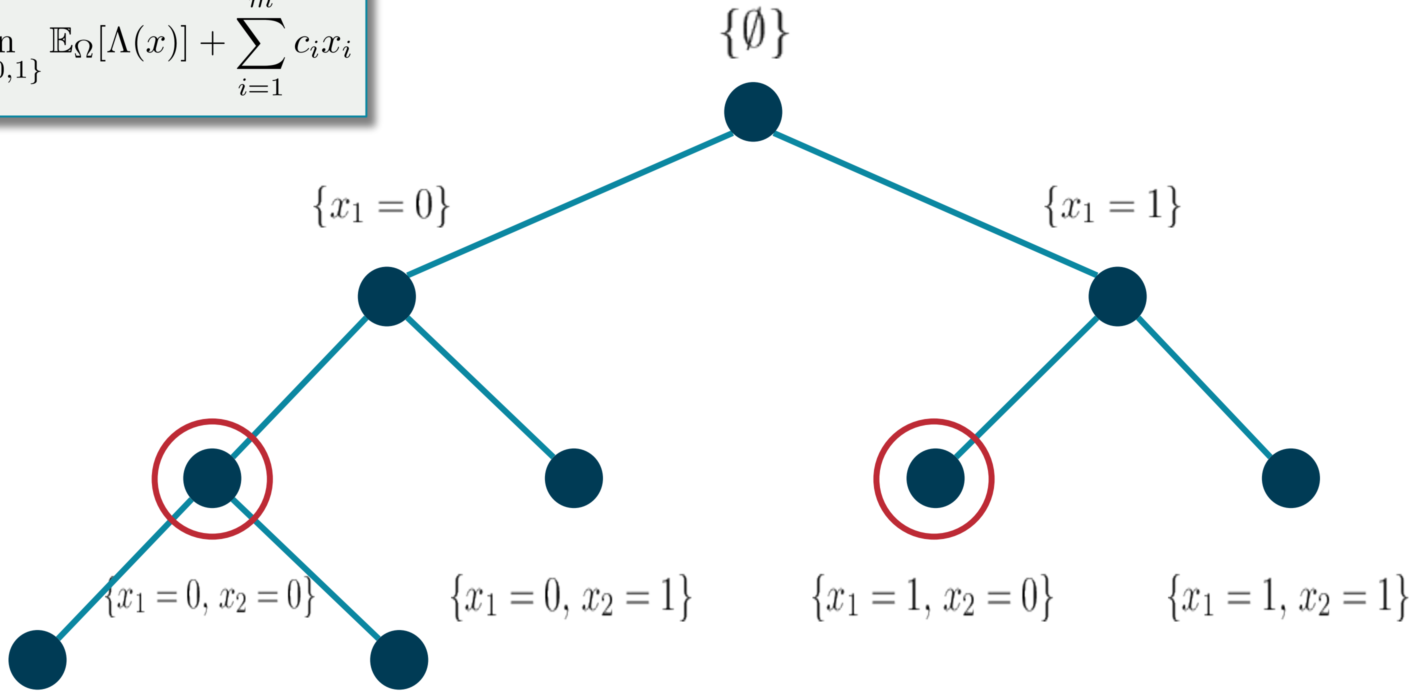
$$\min_{x_i \in \{0,1\}} \mathbb{E}_\Omega[\Lambda(x)] + \sum_{i=1}^{m} c_i x_i$$



Where to branch next?

ML Approach to Crashing

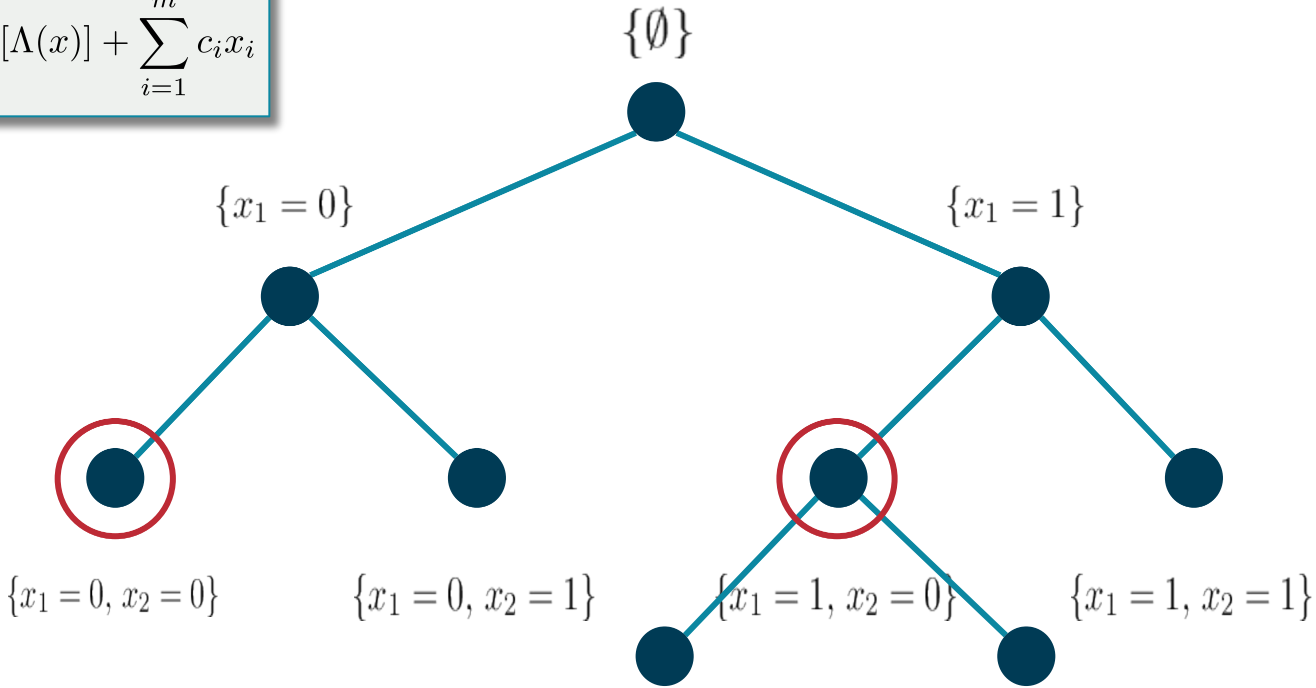
$$\min_{x_i \in \{0,1\}} \mathbb{E}_\Omega[\Lambda(x)] + \sum_{i=1}^{m} c_i x_i$$



Where to branch next?

ML Approach to Crashing

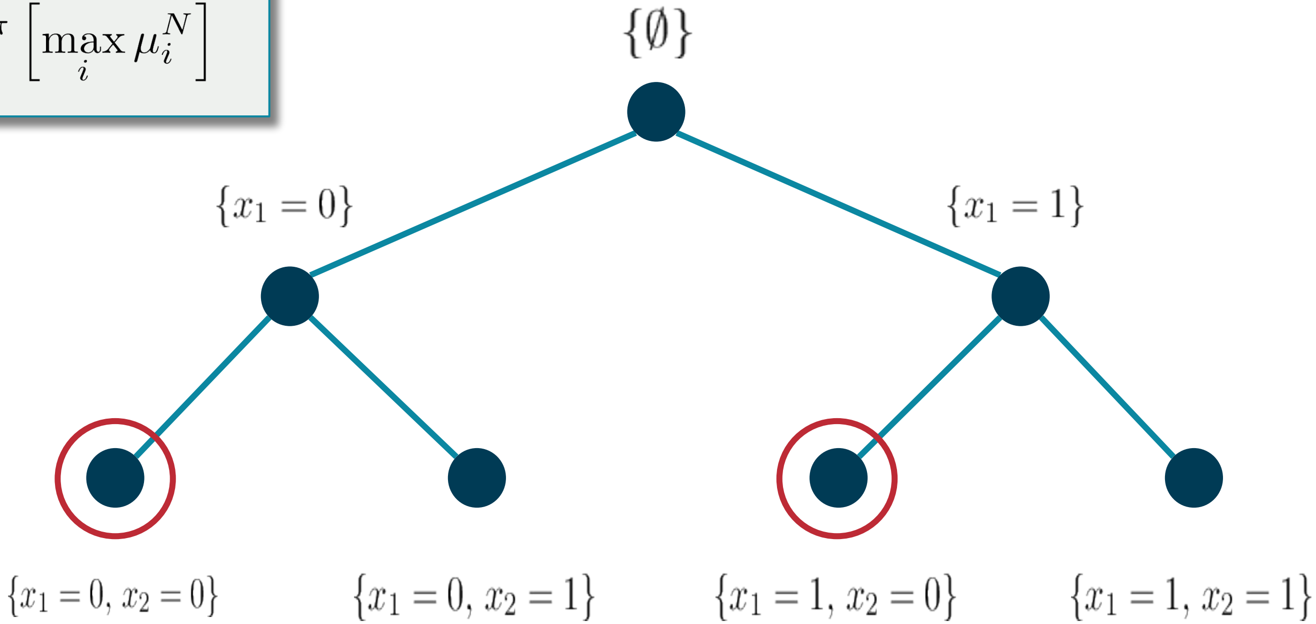
$$\min_{x_i \in \{0,1\}} \mathbb{E}_\Omega[\Lambda(x)] + \sum_{i=1}^{m} c_i x_i$$



Where to branch next?

ML Approach to Crashing

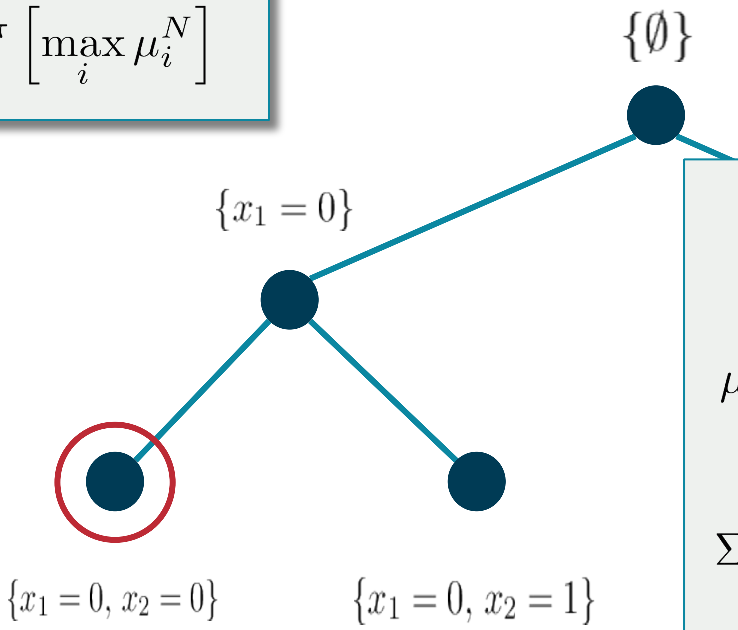
$$\sup_{\pi \in \Pi} \mathbb{E}^{\pi}\left[\max_i \mu_i^N\right]$$



$$\arg\max_x \mathbb{E}_n\left[\max_i \mu_i^{n+1} \mid S^n = s, x^n = x\right] - \max_i \mu_i^n$$

ML Approach to Crashing

$$\sup_{\pi \in \Pi} \mathbb{E}^{\pi}\left[\max_i \mu_i^N\right]$$



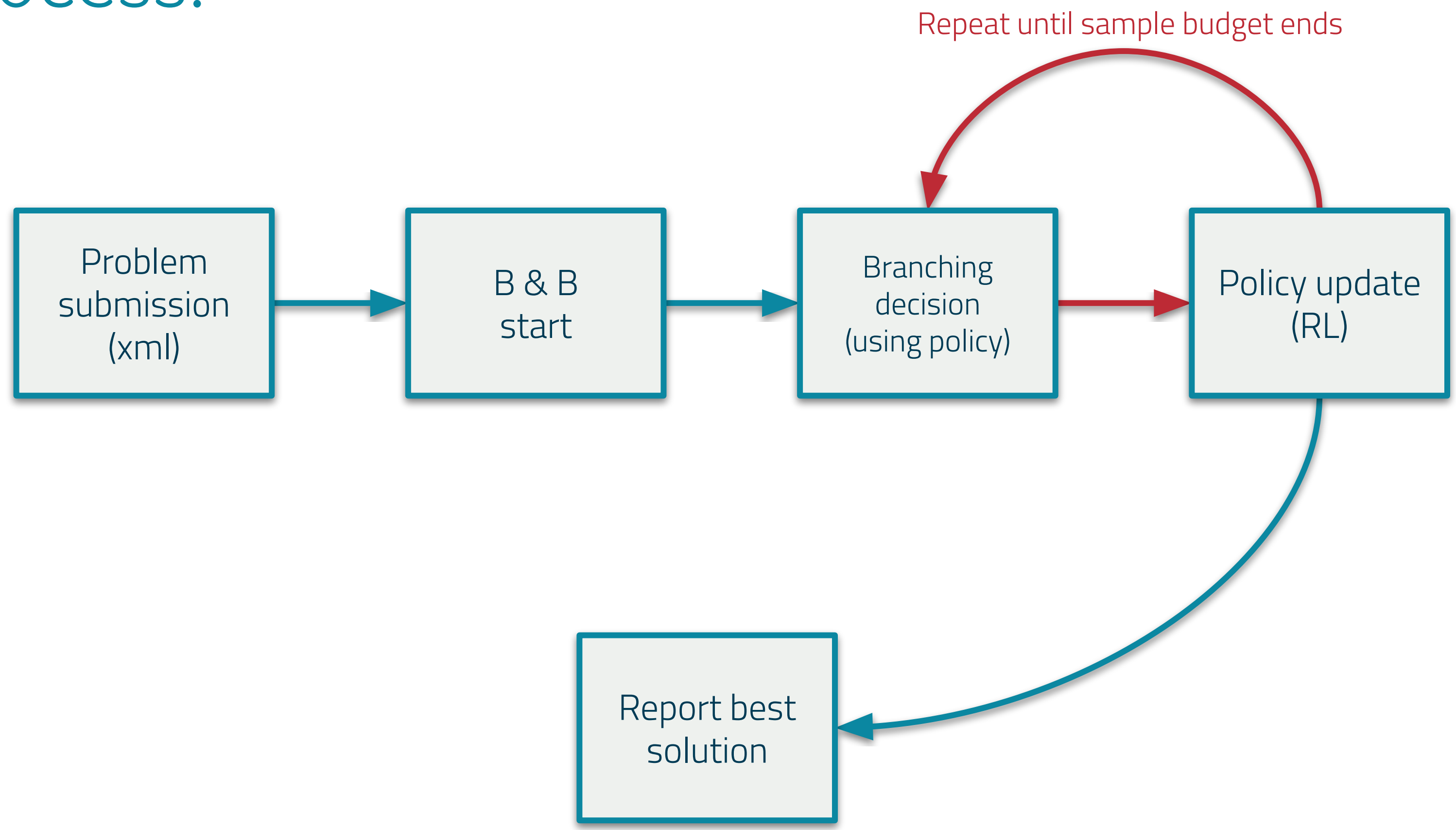
Bayesian Update

$$\mu^{n+1} = \mu^n + \frac{\hat{y}^{n+1} - \mu_x^n}{\lambda_x + \Sigma_{xx}^n} \Sigma^n e_x$$

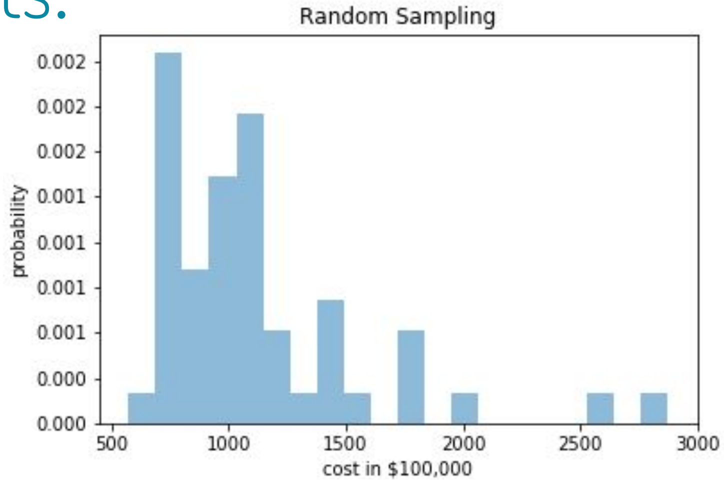
$$\Sigma^{n+1} = \Sigma^n - \frac{\Sigma^n e_x e_x' \Sigma^n}{\lambda_x + \Sigma_{xx}^n}$$

$$\arg\max_x \mathbb{E}_n\left[\max_i \mu_i^{n+1} \mid S^n = s, x^n = x\right] - \max_i \mu_i^n$$

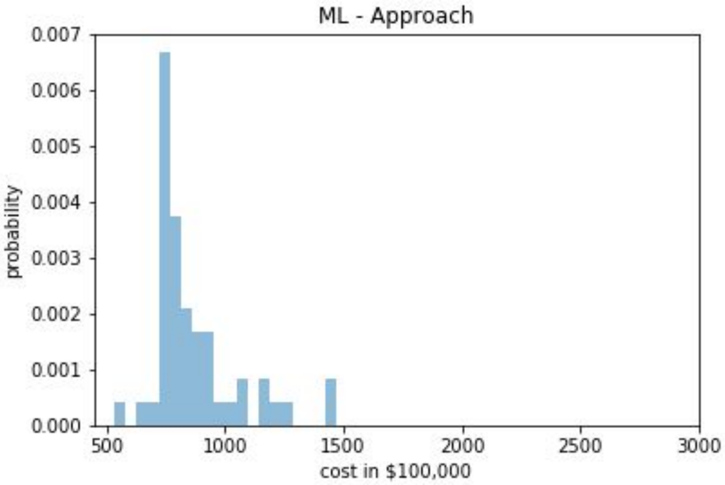
Process:


Repeat until sample budget ends
Problem submission (xml)
B & B start
Branching decision (using policy)
Policy update (RL)
Report best solution

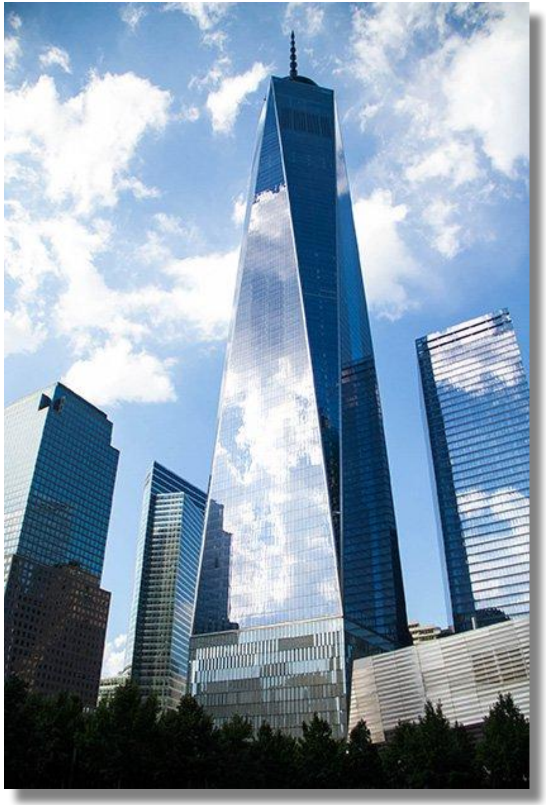
Results:



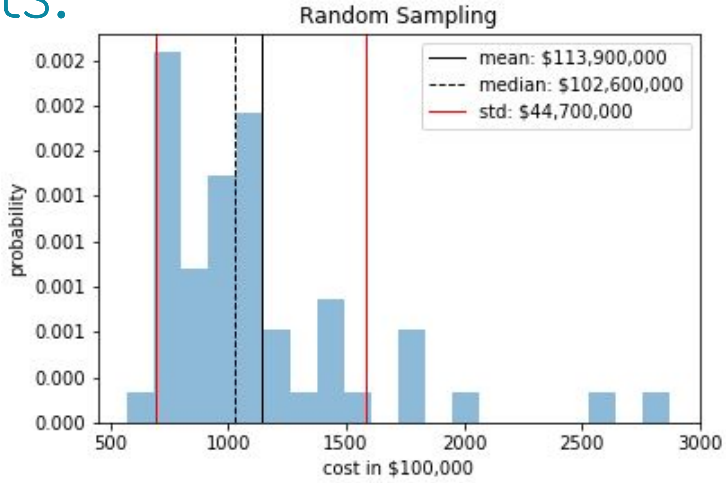
Random Sampling
probability
0.002
0.002
0.002
0.001
0.001
0.001
0.001
0.000
0.000
500
1000
1500
2000
2500
3000
cost in $100,000



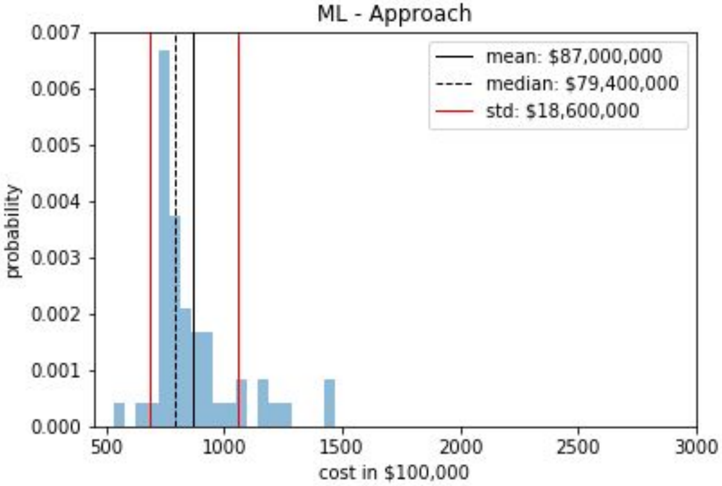
ML - Approach
probability
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
500
1000
1500
2000
2500
3000
cost in $100,000



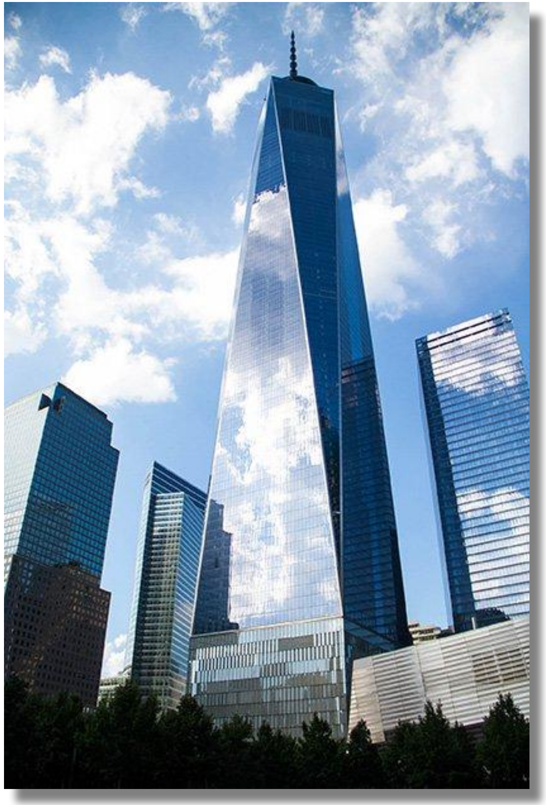
Results:



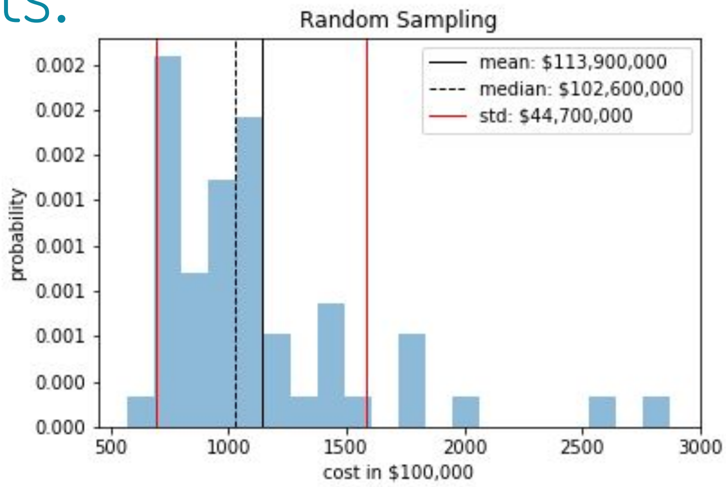
Random Sampling
mean: $113,900,000
median: $102,600,000
std: $44,700,000
probability
0.002
0.002
0.002
0.001
0.001
0.001
0.001
0.000
0.000
500
1000
1500
2000
2500
3000
cost in $100,000

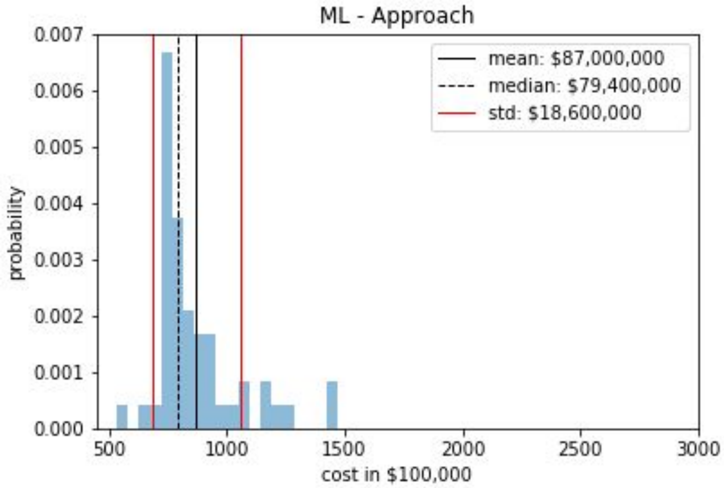


ML - Approach
mean: $87,000,000
median: $79,400,000
std: $18,600,000
probability
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
500
1000
1500
2000
2500
3000
cost in $100,000



Results:





Savings: $24M

Reduced Risk: 59%

Ricardo A. Collado

Rutgers

Princeton University

Stevens Institute of Technology — The Innovation University®

Center for Quantum Science & Engineering

Stevens Institute of Technology

Scenario decomposition of risk-averse multistage stochastic programming problems
RA Collado, D Papp, A Ruszczyński
Annals of Operations Research 200 (1), 147-170

Network interdiction–models, applications, unexplored directions
RA Collado, D Papp
Rutcor Res Rep, RRR4, Rutgers University, New Brunswick, NJ

Quantile optimization for heavy-tailed distribution using asymmetric signum functions
JH Kim, WB Powell, RA Collado
Princeton University

Network interdiction–models, applications
RA Collado, D Papp
unexplored directions. Technical report, RUTCOR Technical Report RRR 4-2012 ...

Risk-averse stochastic path detection
R Collado, S Meisel, L Priekule
European Journal of Operational Research 260 (1), 195-211

Time series forecasting with a learning algorithm: an approximate dynamic programming approach
R Collado, GG Creamer
22nd International Conference on Computational Statistics (COMPSTAT)

Scenario decomposition of risk-averse stochastic optimization problems
R Collado
Rutgers University-Graduate School-New Brunswick

Risk-averse dynamic arbitrage in illiquid markets
S Moazeni, R Collado, A Zhang
Journal of Risk, Forthcoming

Resource Allocation for Contingency Planning: An Inexact Bundle Method for Stochastic Optimization
R Collado, S Moazeni
Available at SSRN 3059005

THRESHOLD RISK MEASURES PART 1: FINITE HORIZON
RA COLLADO, WB POWELL

"A schedule defends from chaos and whim. A net for catching days.

Annie Dillard - American Author

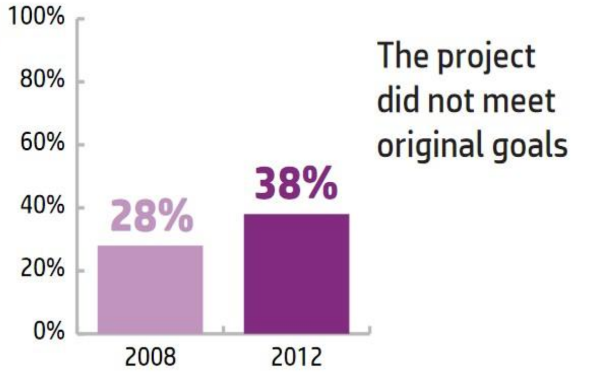
Complexities of Project Management

Companies risk $135 million for every $1 billion spent on a project.[4]

The formula:

% of projects **not meeting goals** (37.7%) ✖ % of projects' **budget lost** (35.9%) = % of every dollar **at risk** (13.5%)

More than one-third of projects did not meet their business intent.[1]



COMPUTERWORLD UNITED STATES INSIDER

FEATURE

IT's biggest project failures -- and what we can learn from them

Homeland Security's virtual fence

The U.S. Department of Homeland Security is bolstering the U.S. Border Patrol with a network of radar, satellites, sensors and communication links -- what's commonly referred to as a "virtual fence." In September 2006, a contract for this Secure Border Initiative Network (SBInet, not to be confused with Skynet) was awarded to Boeing, which was given $20 million to construct a 28-mile pilot section along the Arizona-Mexico border.

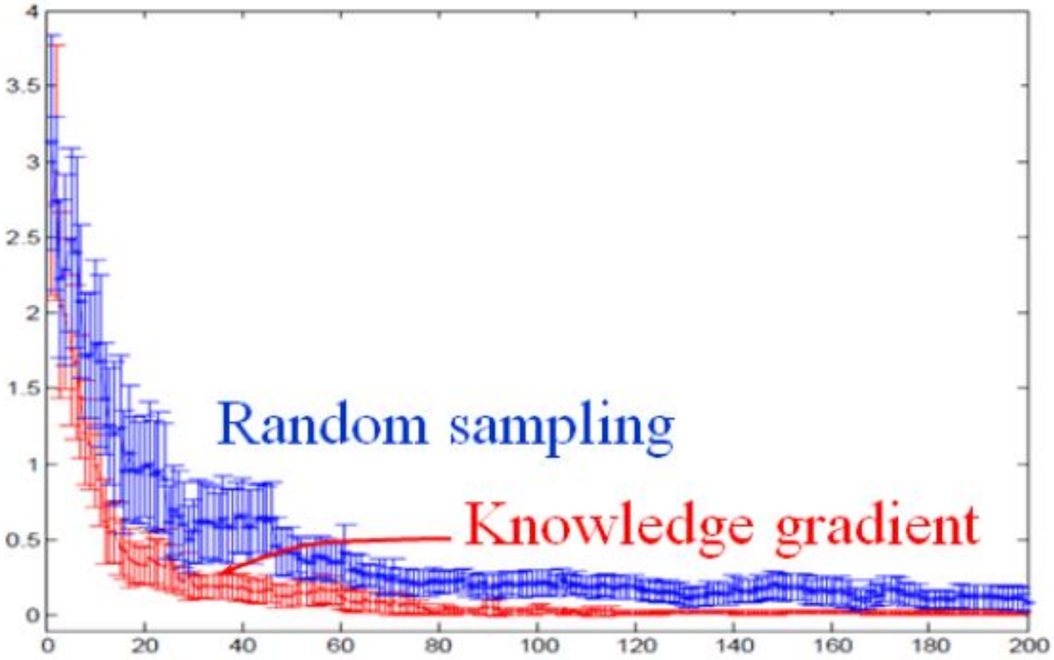
But early this year, Congress learned that the pilot project was being delayed because users had been excluded from the process and the complexity of the project had been underestimated. (Sound familiar?) In February 2008, the Government Accountability Office reported that the radar meant to detect aliens coming across the border could be set off by rain and other weather, and the cameras mean to zoom in on subjects sent back images of uselessly low resolution for objects beyond 3.1 miles. Also, the pilot's communications system interfered with local residents' WiFi networks -- not good PR.

FoxMeyer ERP program

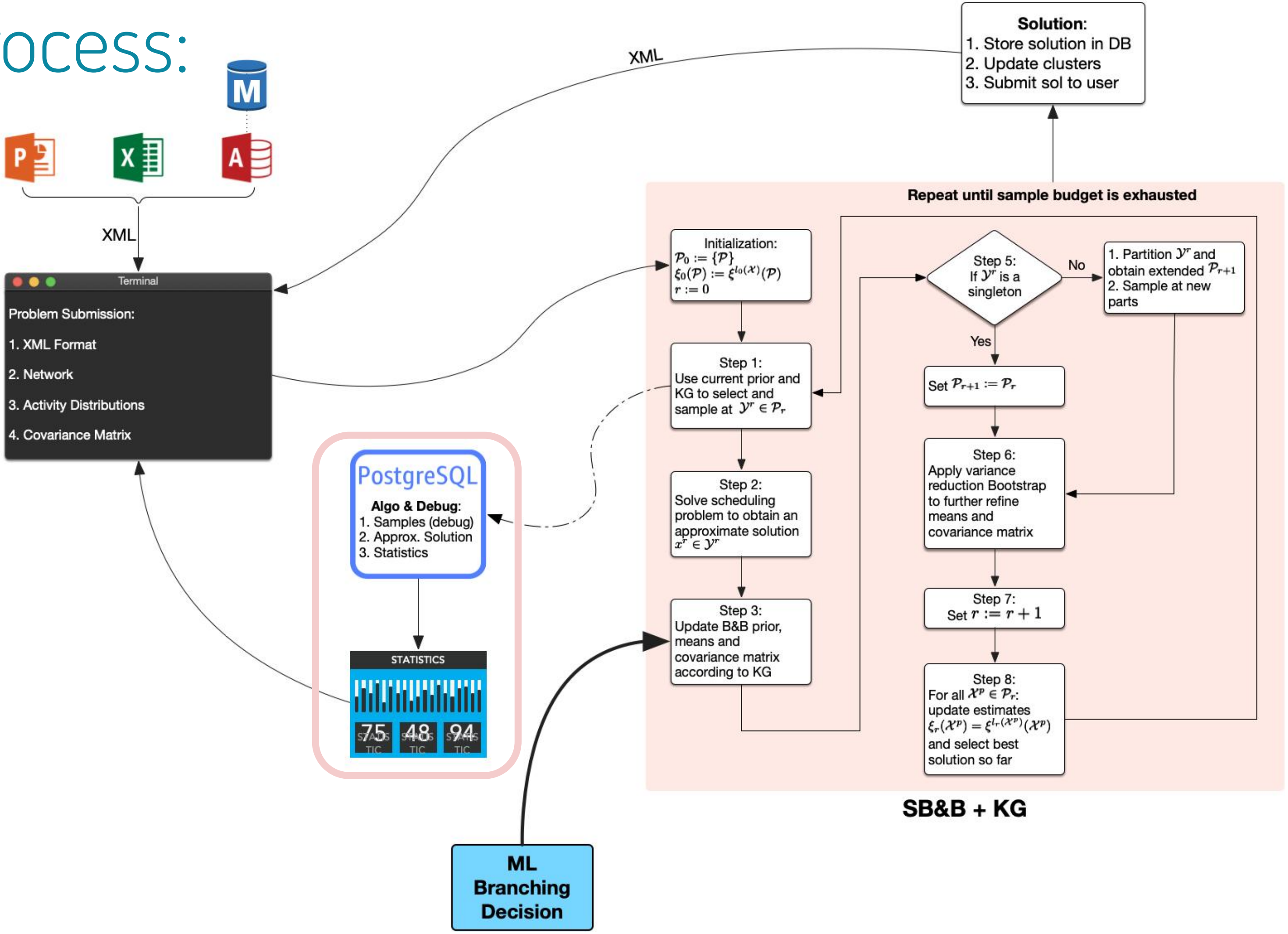
In 1993, FoxMeyer Drugs was the fourth largest distributor of pharmaceuticals in the U.S., worth $5 billion. In an attempt to increase efficiency, FoxMeyer purchased an SAP system and a warehouse automation system and hired Andersen Consulting to integrate and implement the two in what was supposed to be a $35 million project. By 1996, the company was bankrupt; it was eventually sold to a competitor for a mere $80 million.

The reasons for the failure are familiar. First, FoxMeyer set up an unrealistically aggressive time line -- the entire system was supposed to be implemented in 18 months. Second, the warehouse employees whose jobs were affected -- more accurately, threatened -- by the automated system were not supportive of the project, to say the least. After three existing warehouses were closed, the first warehouse to be automated was plagued by sabotage, with inventory damaged by workers and orders going unfilled.

Expected Performance of KG

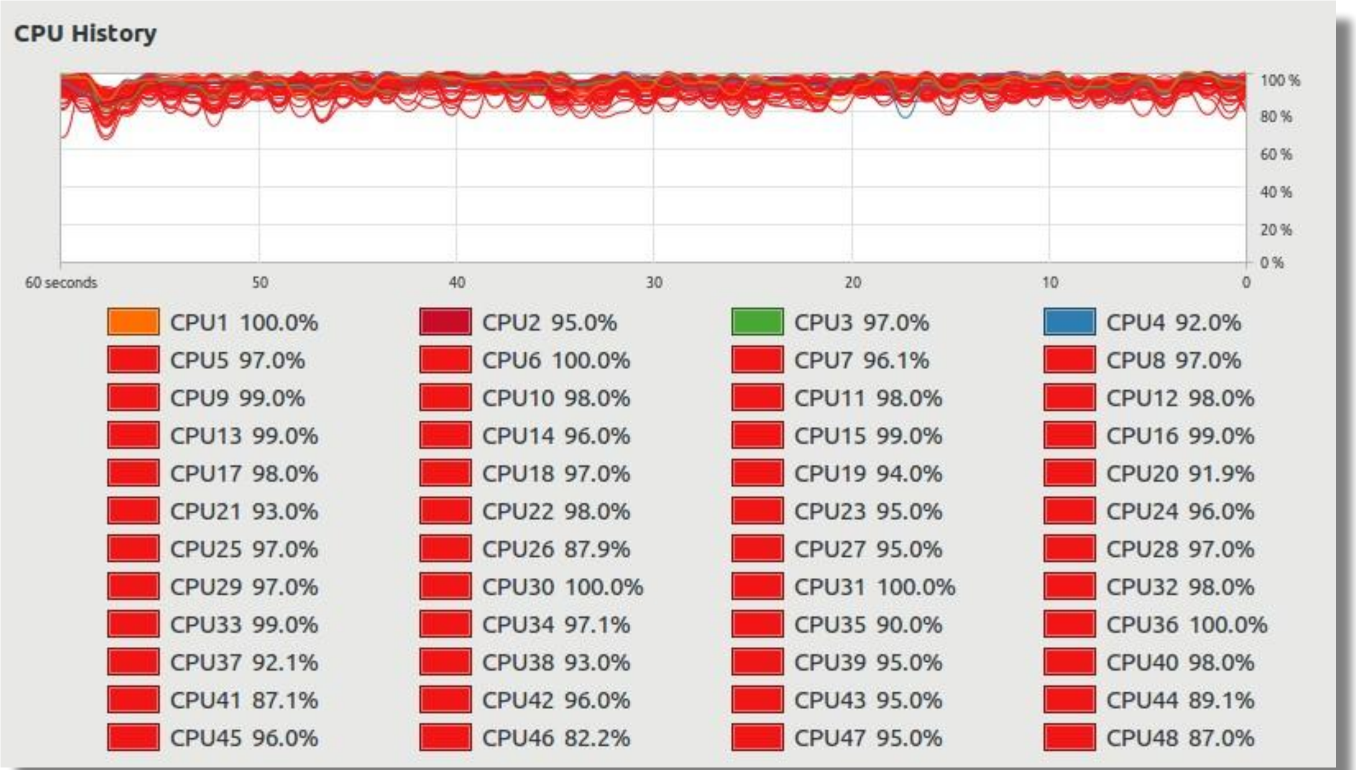


Process:



XML
Solution:
1. Store solution in DB
2. Update clusters
3. Submit sol to user
XML
Terminal
Problem Submission:
1. XML Format
2. Network
3. Activity Distributions
4. Covariance Matrix
PostgreSQL
Algo & Debug:
1. Samples (debug)
2. Approx. Solution
3. Statistics
STATISTICS
75
48
94
Repeat until sample budget is exhausted
Initialization:
$\mathcal{P}_0 := \{\mathcal{P}\}$
$\xi_0(\mathcal{P}) := \xi^{l_0(\mathcal{X})}(\mathcal{P})$
$r := 0$
Step 1:
Use current prior and KG to select and sample at $\mathcal{Y}^r \in \mathcal{P}_r$
Step 2:
Solve scheduling problem to obtain an approximate solution $x^r \in \mathcal{Y}^r$
Step 3:
Update B&B prior, means and covariance matrix according to KG
Step 5:
If $\mathcal{Y}^r$ is a singleton
No
Yes
1. Partition $\mathcal{Y}^r$ and obtain extended $\mathcal{P}_{r+1}$
2. Sample at new parts
Set $\mathcal{P}_{r+1} := \mathcal{P}_r$
Step 6:
Apply variance reduction Bootstrap to further refine means and covariance matrix
Step 7:
Set $r := r + 1$
Step 8:
For all $\mathcal{X}^p \in \mathcal{P}_r$:
update estimates
$\xi_r(\mathcal{X}^p) = \xi^{l_r(\mathcal{X}^p)}(\mathcal{X}^p)$
and select best solution so far
SB&B + KG
ML Branching Decision

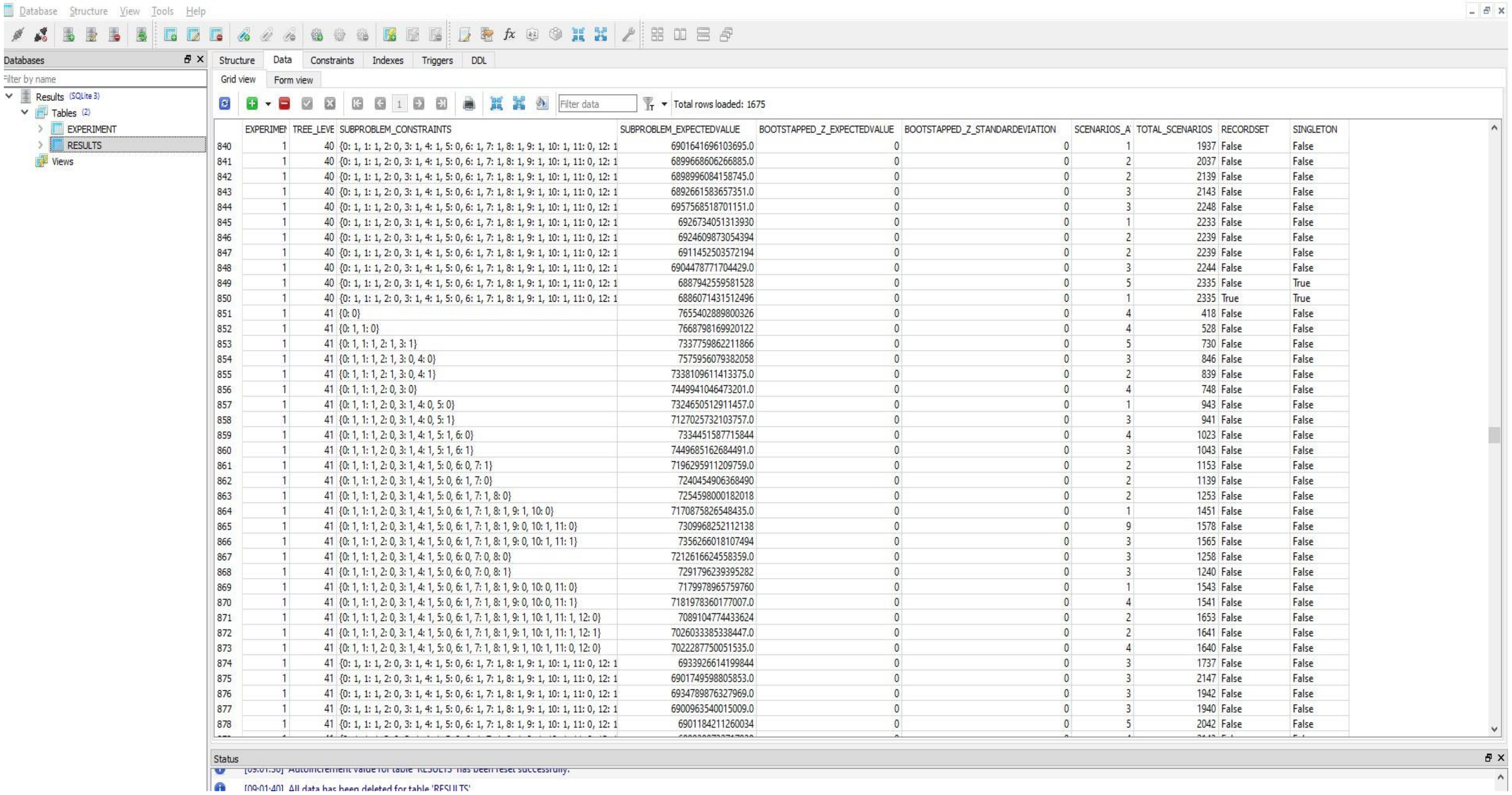
Parallel Execution



Project Costs

Building	City	Floors	Height (meters)	year	Cost ($Billion)	Cost per Floor ($ million)
One World Trade Center	New York City	104	541	2014	3.9	37.5
Shanghai Tower	Shanghai	121	632	2015	2.4	19.8
The Shard	London	73	306	2013	1.9	26.0
Taipei 101	Taipei	101	509	2004	1.8	17.4
Burj Khalifa	Dubai	163	828	2010	1.5	9.2

```
Using license file C:\Users\Ricardo\Licenses\Gurobi\gurobi.lic
Academic license - for non-commercial use only

ObjVal = 113.0
OptSol = [0, 0, 17, 31, 31, 43, 38, 45, 45, 61, 75, 38, 75, 75, 59, 69, 85, 104, 104, 113]

ObjVal = 362.0
OptSol = [0, 0, 22, 51, 51, 73, 108, 130, 130, 156, 250, 108, 250, 250, 174, 189, 275, 344, 344, 362]
Using license file C:\Users\Ricardo\Licenses\Gurobi\gurobi.lic
Total Time taken to execute Random is 41.0012246



ObjVal = 141.0
OptSol = [0, 0, 14, 22, 22, 42, 22, 0, 53, 64, 75, 95, 95, 95, 107, 116, 131, 116, 95, 141]

ObjVal = 359.0
OptSol = [0, 0, 74, 87, 87, 125, 87, 0, 141, 172, 203, 258, 258, 258, 290, 304, 344, 304, 258, 359]
Using license file C:\Users\Ricardo\Licenses\Gurobi\gurobi.lic
Total Time taken to execute Uniform is 299.1538673



ObjVal = 107.0
OptSol = [0, 0, 15, 32, 0, 40, 15, 49, 32, 42, 56, 64, 56, 75, 85, 97, 85, 85, 92, 107]

ObjVal = 303.0
OptSol = [0, 0, 30, 77, 0, 115, 30, 184, 77, 112, 141, 209, 184, 190, 256, 278, 256, 210, 273, 303]
Using license file C:\Users\Ricardo\Licenses\Gurobi\gurobi.lic
Total Time taken to execute Distance is 74.357485



ObjVal = 114.0
OptSol = [0, 0, 0, 0, 18, 28, 46, 46, 28, 57, 57, 74, 83, 66, 96, 66, 96, 18, 96, 114]

ObjVal = 456.0
OptSol = [0, 0, 0, 0, 93, 113, 206, 206, 113, 227, 227, 304, 343, 286, 366, 286, 366, 93, 366, 456]
Using license file C:\Users\Ricardo\Licenses\Gurobi\gurobi.lic
Total Time taken to execute Pareto_Inverse is 82.00499100000002
```



	EXPERIMEN	TREE_LEVE	SUBPROBLEM_CONSTRAINTS	SUBPROBLEM_EXPECTEDVALUE	BOOTSTAPPED_Z_EXPECTEDVALUE	BOOTSTAPPED_Z_STANDARDDEVIATION	SCENARIOS_A	TOTAL_SCENARIOS	RECORDSET	SINGLETON
840	1	40	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6901641696103695.0	0	0	1	1937	False	False
841	1	40	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6899668606266885.0	0	0	2	2037	False	False
842	1	40	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6898996084158745.0	0	0	2	2139	False	False
843	1	40	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6892661583657351.0	0	0	3	2143	False	False
844	1	40	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6957568518701151.0	0	0	3	2248	False	False
845	1	40	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6926734051313930	0	0	1	2233	False	False
846	1	40	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6924609873054394	0	0	2	2239	False	False
847	1	40	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6911452503572194	0	0	2	2239	False	False
848	1	40	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6904478771704429.0	0	0	3	2244	False	False
849	1	40	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6887942559581528	0	0	5	2335	False	True
850	1	40	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6886071431512496	0	0	1	2335	True	True
851	1	41	{0: 0}	7655402889800326	0	0	4	418	False	False
852	1	41	{0: 1, 1: 0}	7668798169920122	0	0	4	528	False	False
853	1	41	{0: 1, 1: 1, 2: 1, 3: 1}	7337759862211866	0	0	5	730	False	False
854	1	41	{0: 1, 1: 1, 2: 1, 3: 0, 4: 0}	7575956079382058	0	0	3	846	False	False
855	1	41	{0: 1, 1: 1, 2: 1, 3: 0, 4: 1}	7338109611413375.0	0	0	2	839	False	False
856	1	41	{0: 1, 1: 1, 2: 0, 3: 0}	7449941046473201.0	0	0	4	748	False	False
857	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 0, 5: 0}	7324650512911457.0	0	0	1	943	False	False
858	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 0, 5: 1}	7127025732103757.0	0	0	3	941	False	False
859	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 1, 6: 0}	7334451587715844	0	0	4	1023	False	False
860	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 1, 6: 1}	7449685162684491.0	0	0	3	1043	False	False
861	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 0, 7: 1}	7196295911209759.0	0	0	2	1153	False	False
862	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 0}	7240454906368490	0	0	2	1139	False	False
863	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 0}	7254598000182018	0	0	2	1253	False	False
864	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 0}	7170875826548435.0	0	0	1	1451	False	False
865	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 0, 10: 1, 11: 0}	7309968252112138	0	0	9	1578	False	False
866	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 0, 10: 1, 11: 1}	7356266018107494	0	0	3	1565	False	False
867	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 0, 7: 0, 8: 0}	7212616624558359.0	0	0	3	1258	False	False
868	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 0, 7: 0, 8: 1}	7291796239395282	0	0	3	1240	False	False
869	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 0, 10: 0, 11: 0}	7179978965759760	0	0	1	1543	False	False
870	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 0, 10: 0, 11: 1}	7181978360177007.0	0	0	4	1541	False	False
871	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 1, 12: 0}	7089104774433624	0	0	2	1653	False	False
872	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 1, 12: 1}	7026033385338447.0	0	0	2	1641	False	False
873	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 0}	7022287750051535.0	0	0	4	1640	False	False
874	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6933926614199844	0	0	3	1737	False	False
875	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6901749598805853.0	0	0	3	2147	False	False
876	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6934789876327969.0	0	0	3	1942	False	False
877	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6900963540015009.0	0	0	3	1940	False	False
878	1	41	{0: 1, 1: 1, 2: 0, 3: 1, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 0, 12: 1	6901184211260034	0	0	5	2042	False	False

Extensions & Pivots

Extensions

- Extra policy: random policy (used in practice)
- Extra Policy: based on evolutionary optimization (GA)
- Input from standard OM software (MS Project)

Pivots

- Limit problem generation to a narrow class to avoid complex clustering and extra computation time in training
- Limit to uncorrelated KG (simpler and faster to train) but slower to converge