Stochastic Modeling using Virtual Training Sets

2016 AIChE Spring Meeting
Big Data Analytics and Fundamental Modeling

James C Cross III
April 12, 2016

Overview

For a stochastic system, a "reference" forecast offers a view of an "expected" outcome, but does not provide any insight on the distribution of alternative outcomes.

OBJECTIVE: **Create a forecast for outcome probabilities**

- Outcomes are solutions to optimization problems
- The optimization problems depend on stochastic parameters
- Solving the optimization problem is computationally expensive

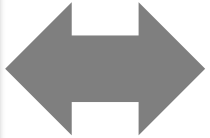
... What can be done?

Methods for estimating the likelihood of alternative outcomes are central to data analytics, to improve operational decision-making under uncertainty.

Solution Strategies

For computationally expensive optimizations, there may be an advantage to trading off individual solution accuracy for improved runtime performance and ensemble resolution.

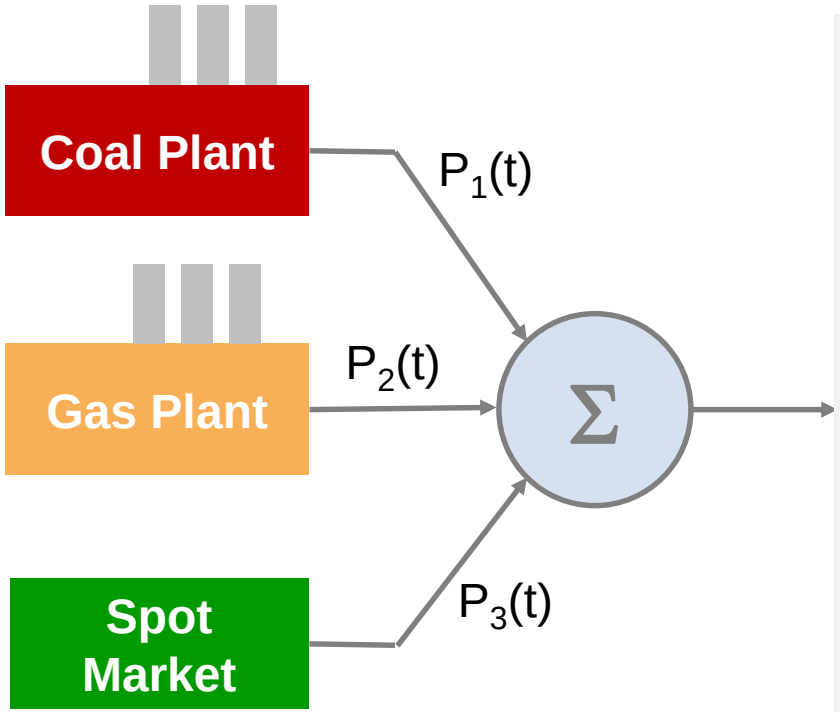
Traditional	Alternative
(1) Generate ensemble of scenarios (2) Run **optimization** for each scenario (3) Assemble the distribution	(1) Build training set from prior solutions (2) Create model (3) Generate ensemble of scenarios (4) Run **model** for each scenario (5) Assemble the distribution
• Optimal, but slow solutions • Low resolution distributions	• Approximate, but fast solutions • High resolution distributions

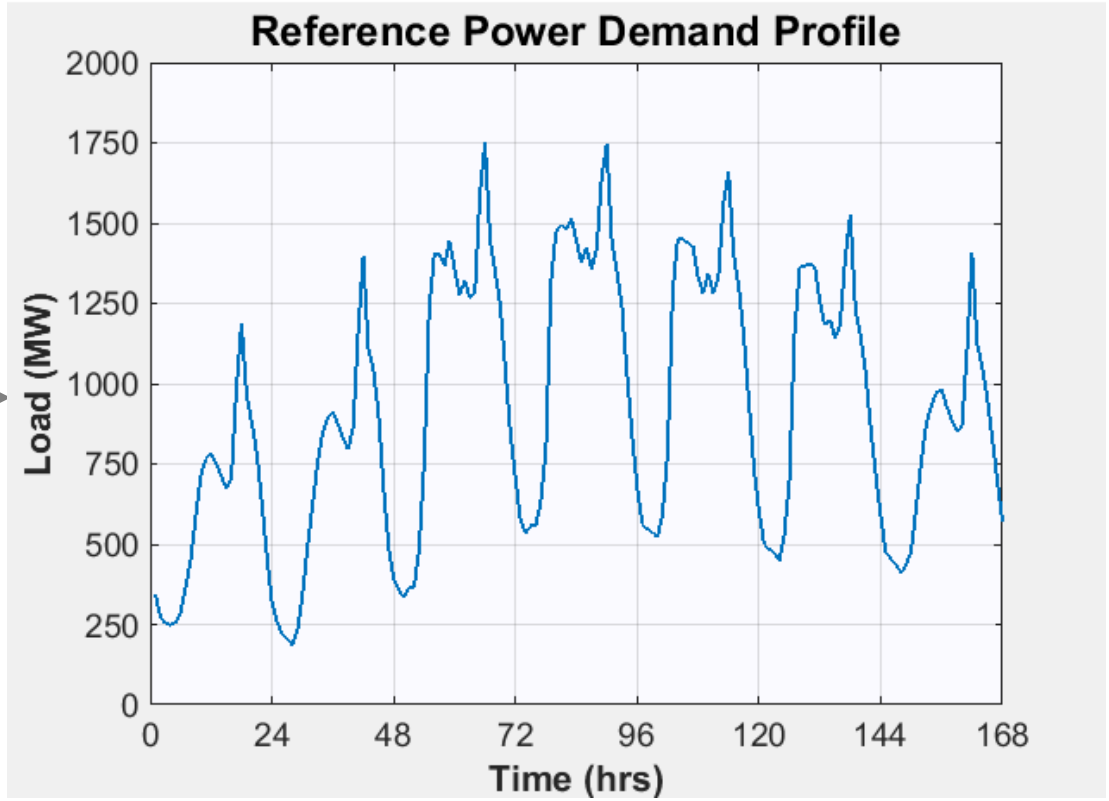


Can a model "learn" the solutions of a complex optimization problem?

Application: Power Plant Dispatch

The Unit Commitment Problem: What operating schedule, for a pool of power plants, delivers the total power demand at the lowest cost?





Question posed in this study: What is the probability distribution of total cost and plant operating capacities corresponding to variations in fuel prices and power demand?

Optimization Problem

This example is a Mixed Integer Linear Programming (MILP) optimization problem ...

Minimize total cost

- Meet total power demand
- Abide operational constraints

$$\min_x f^T x \text{ subject to } \begin{cases} x(\text{intcon}) \text{ are integers} \\ A \cdot x \le b \\ Aeq \cdot x = beq \\ lb \le x \le ub. \end{cases}$$

Solved using ***intlinprog*** function in *MATLAB Optimization Toolbox*

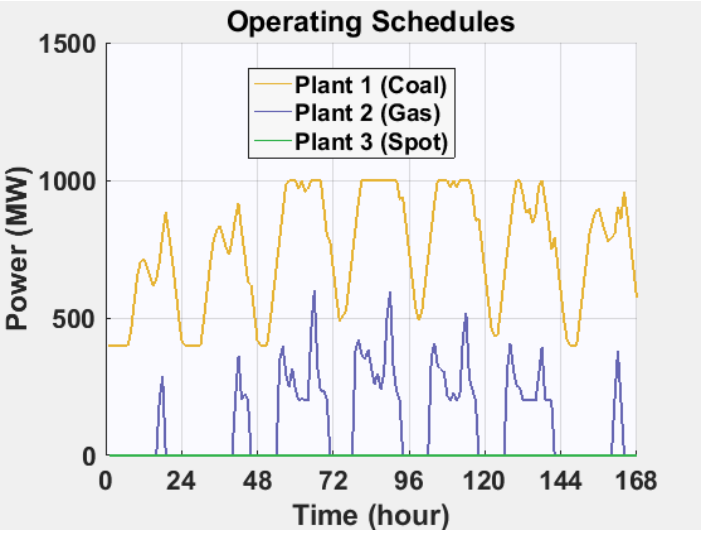
REFERENCE Plant Parameter Values										
Plant ID	Fuel Type	Fuel Cost	Operating Cost	Startup Cost	Min Power	Max Power	Max Ramp+	Max Ramp-	Min Up	Min Down
		$/MWh	$/h	$/start	MW	MW	MW/h	MW/h	h	h
1	Coal	32	1000	500	400	1000	100	100	3	2
2	Gas	36	2000	100	200	1000	400	400	1	1
3	Spot	200	200	100000	100	2000	2000	2000	1	0

… and is complex on account of the plant operating ranges and time-based constraints.

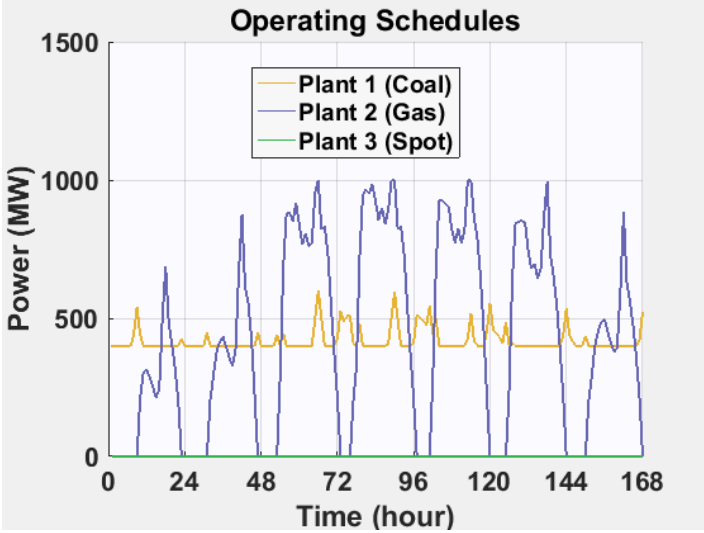
Optimization Solutions

The impact of changes in parameter values on the solutions (the schedules leading to the lowest total cost) is readily apparent.

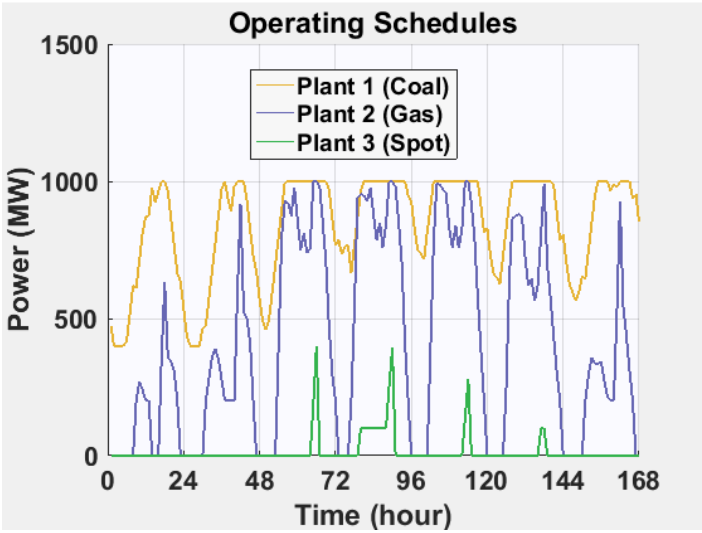
- Coal cost: 0.8 x ref
- Gas cost: 1.2 x ref
- Max(Load) < Capacity



- Coal cost: 1.2 x ref
- Gas cost: 0.8 x ref
- Max(Load) < Capacity



- Coal cost: ref
- Gas cost: ref
- Max(Load) > Capacity



What set of cases should be used to build a training set?

Build Training Set

The training set can be developed using pre-existing "data", or via a "design of experiments" approach involving deliberate optimizations over a specified ensemble.

Inputs

- Weekly load profile (daily variability)
- Coal & gas prices (weekly variability)
- Plant parameters

Training ensemble

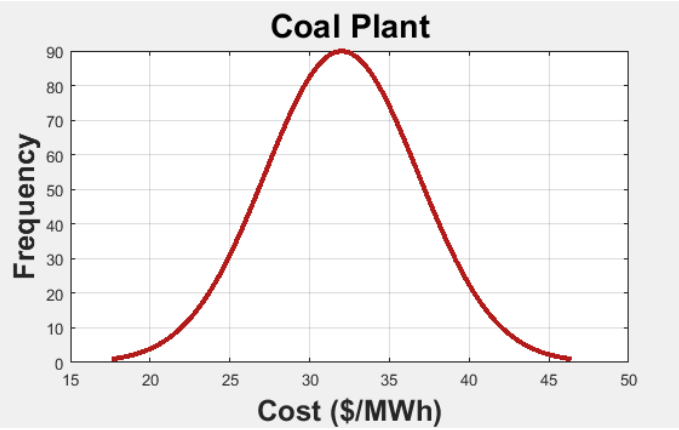
- Normal distributions
- Quantile-based levels (2, 3, 4, & 5)
- Fuel prices: $\sigma = 15\%$
- Load level: $\sigma = 20\%$

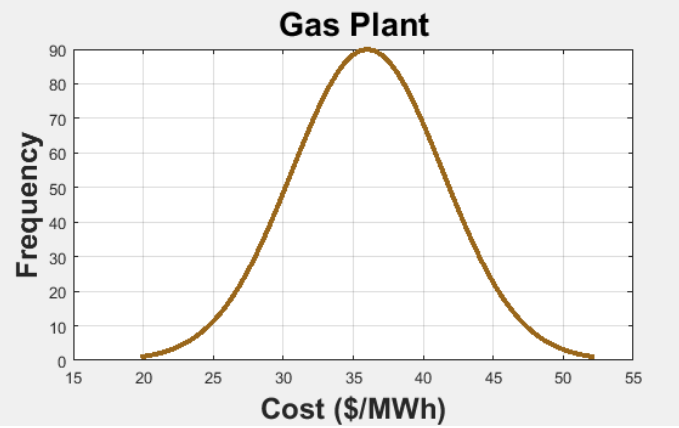
Predictors

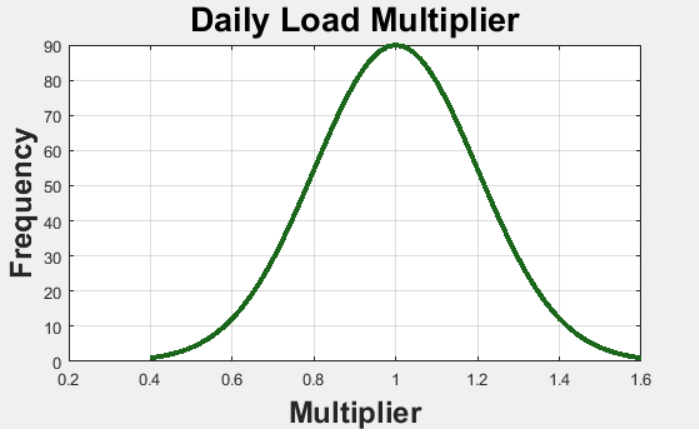
- Load (t-6 to t+6)
- Fuel prices
- Hour of day
- Day of week
- 17-element vector

Outputs

- Coal plant $P_1(t)$
- Gas plant $P_2(t)$
- Spot market $P_3(t)$







Outputs are generated by running the optimization model on progressive scenarios drawn from the training ensemble.

Training Set Assembly

The training set is assembled by collecting the predictors and outputs, and randomly sorting.

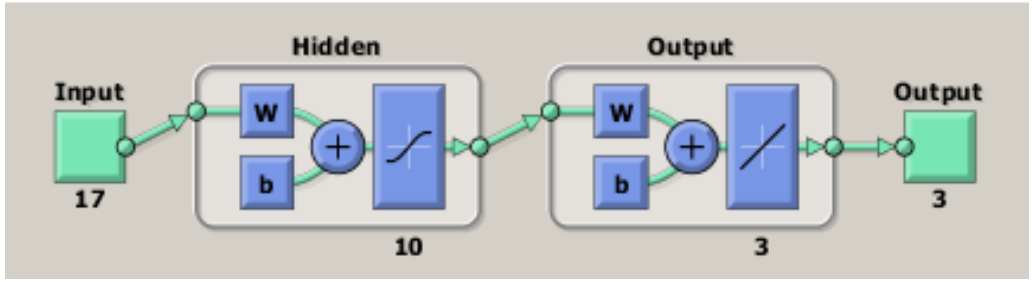
Virtual Training Set (3 levels)																		
Predictors																Outputs		
Load (-6)	Load (-5)	Load (-4)	Load (-3)	Load (-2)	Load (-1)	Load (0)	Load (+1)	Load (+2)	Load (+3)	Load (+4)	Load (+5)	Load (+6)	Coal ($/MWh)	Gas ($/MWh)	Hour	Day	P1 (coal) (MW)	P2 (gas) (MW)	P3(spot) (MW)
474	611	751	870	926	964	973	923	879	844	861	1089	1393	32.1	39.6	12	7	973	0	0
1670	1883	1881	1864	1850	1729	1664	1739	1662	1726	2023	2153	1817	35.3	35.8	13	5	1000	664	0
1650	1555	1461	1511	1451	1465	1824	2004	1654	1544	1422	1196	970	28.8	39.6	17	3	1000	724	100
1321	1548	1648	1390	1294	1143	936	732	600	496	481	470	446	28.8	39.6	22	5	736	200	0
617	749	841	884	906	867	824	790	854	1155	1384	1101	1033	28.8	39.6	14	2	824	0	0
1876	1784	1844	1760	1837	2134	2262	1871	1751	1606	1331	1086	877	32.1	35.8	18	4	1000	1000	262
486	390	355	332	360	361	477	744	1227	1390	1392	1357	1430	32.1	35.8	5	3	477	0	0
1650	1555	1461	1511	1451	1465	1824	2004	1654	1544	1422	1196	970	28.8	32.1	17	3	1000	724	100
1494	1223	957	784	649	628	614	583	687	907	1491	1764	1773	35.3	35.8	3	6	614	0	0
530	552	554	628	786	1307	1462	1483	1466	1502	1436	1365	1411	35.3	35.8	8	4	807	655	0
1350	1176	878	636	509	463	434	471	472	624	972	1604	1816	32.1	39.6	2	3	434	0	0
1876	1784	1844	1760	1837	2134	2262	1871	1751	1606	1331	1086	877	35.3	35.8	18	4	1000	1000	262
1997	1652	1546	1418	1175	959	774	642	626	616	596	680	902	35.3	39.6	24	4	774	0	0
878	636	509	463	434	471	472	624	972	1604	1816	1819	1773	28.8	32.1	4	3	524	0	0
970	807	658	612	637	639	724	907	1508	1687	1710	1691	1733	32.1	35.8	5	4	724	0	0
959	774	642	626	616	596	680	902	1474	1663	1661	1646	1633	35.3	39.6	5	5	680	0	0
711	864	971	1020	1046	1001	950	911	985	1332	1597	1270	1192	28.8	32.1	14	2	950	0	0
...																			

Once the “virtual training set” has been assembled, a model can be created.

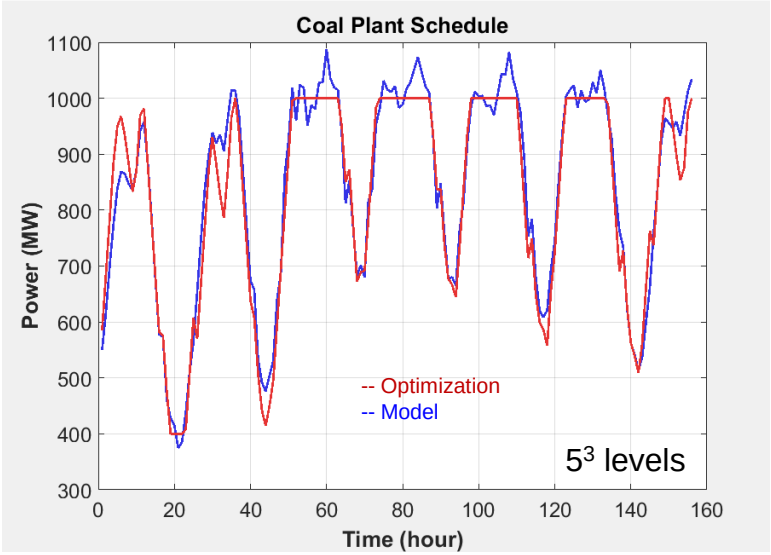
Create Model

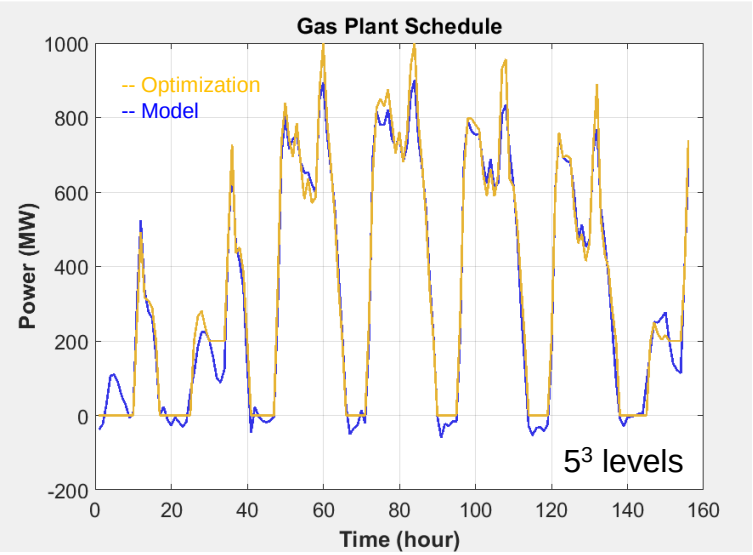
There are a number of candidate machine learning frameworks for modeling the optimization solutions – in the present case, a neural network model was selected.

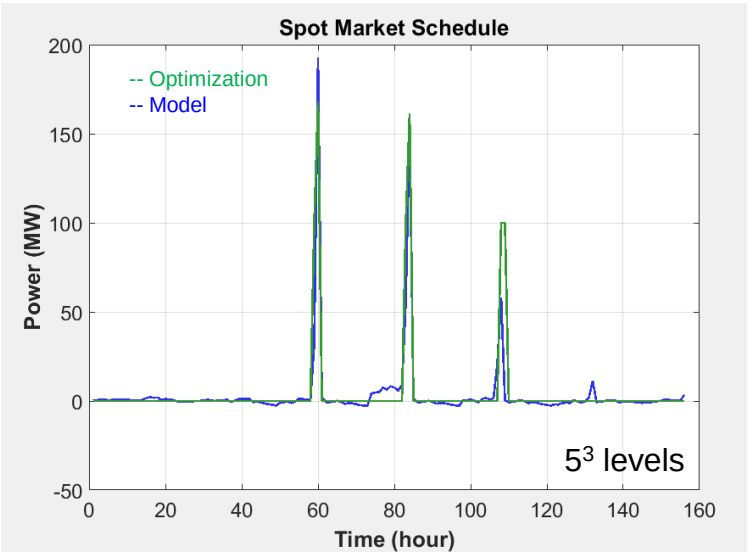
- Specify neural network structure (1HL, 10 nodes)
- Train the model
- Validate the model



Model created using ***train*** function in *MATLAB Statistics & Machine Learning Toolbox*





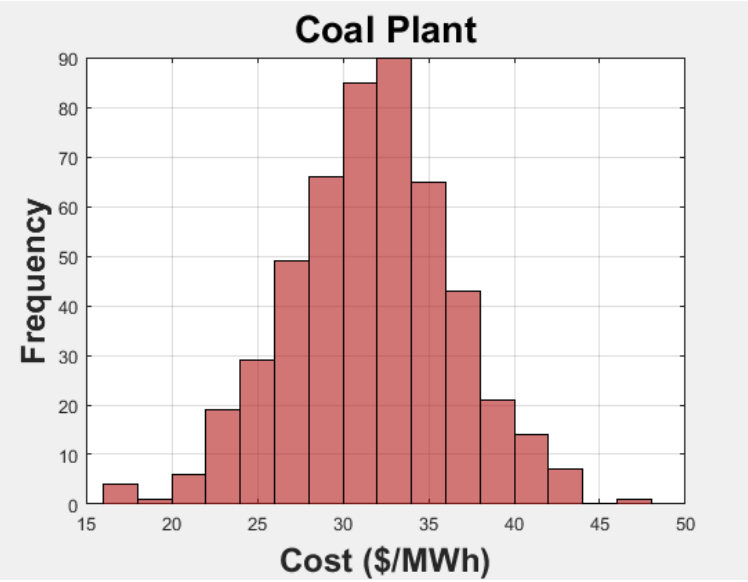


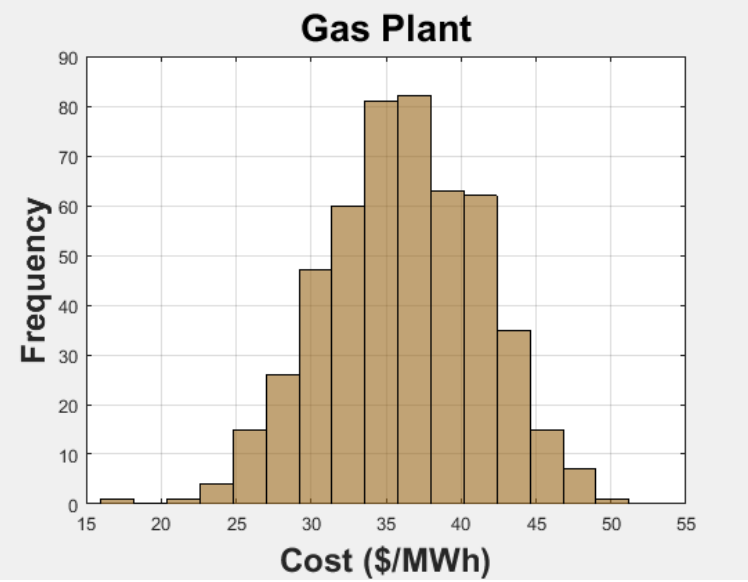
The inaccuracies of the model for a single scenario are apparent…but it's possible that some of these errors will "cancel out" in the assembly of the probability distribution.

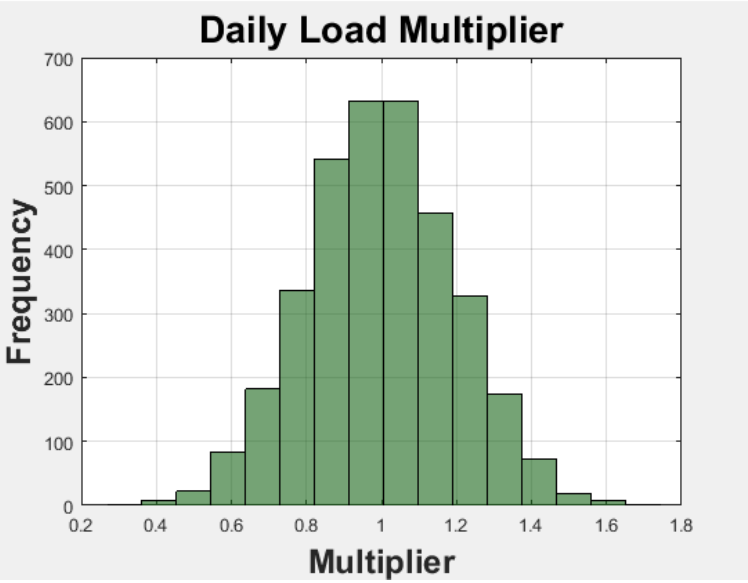
Generate Scenarios

For simplicity, independent normal distributions were assumed for fuel price and aggregate electrical demand fluctuations (as used in training).

Ensemble: 500 scenarios







μ = \$32/MWh
σ = \$4.8/MWh

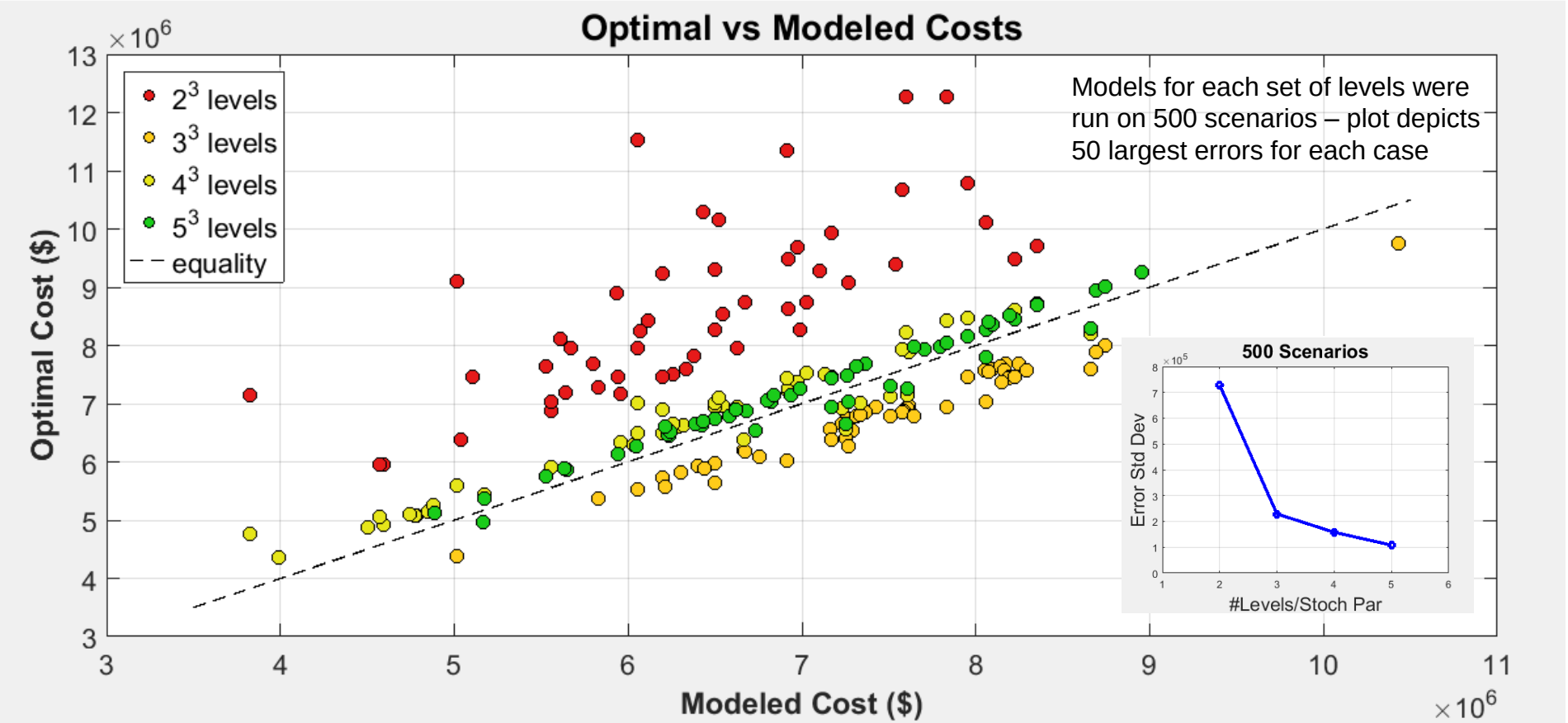
μ = \$36/MWh
σ = \$5.4/MWh

μ = 1.0
σ = 0.2

An individual week-long scenario is defined by nine normal random variates: one for each of the stochastic parameters (2 fuel prices + 7 daily load multipliers).

Run Model

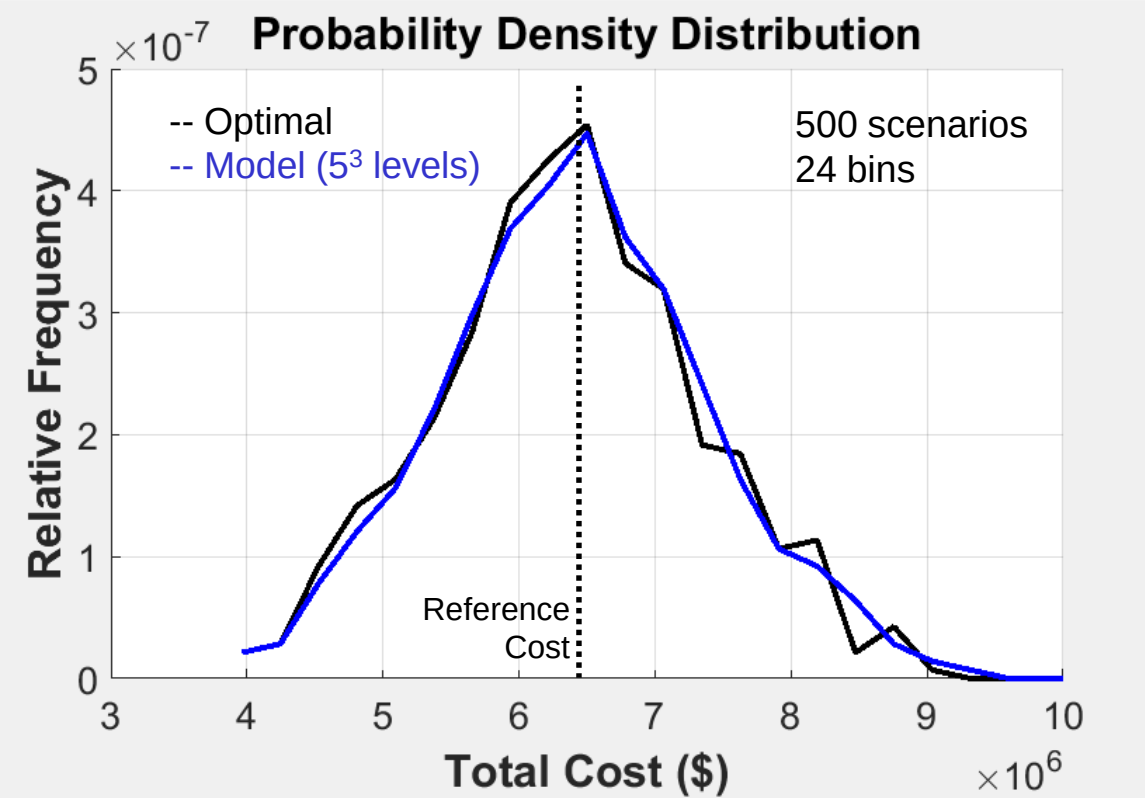
A plot of the optimal versus the modeled cost indicates the effectiveness of the proposed approach, and shows the impact of training set size on the results.

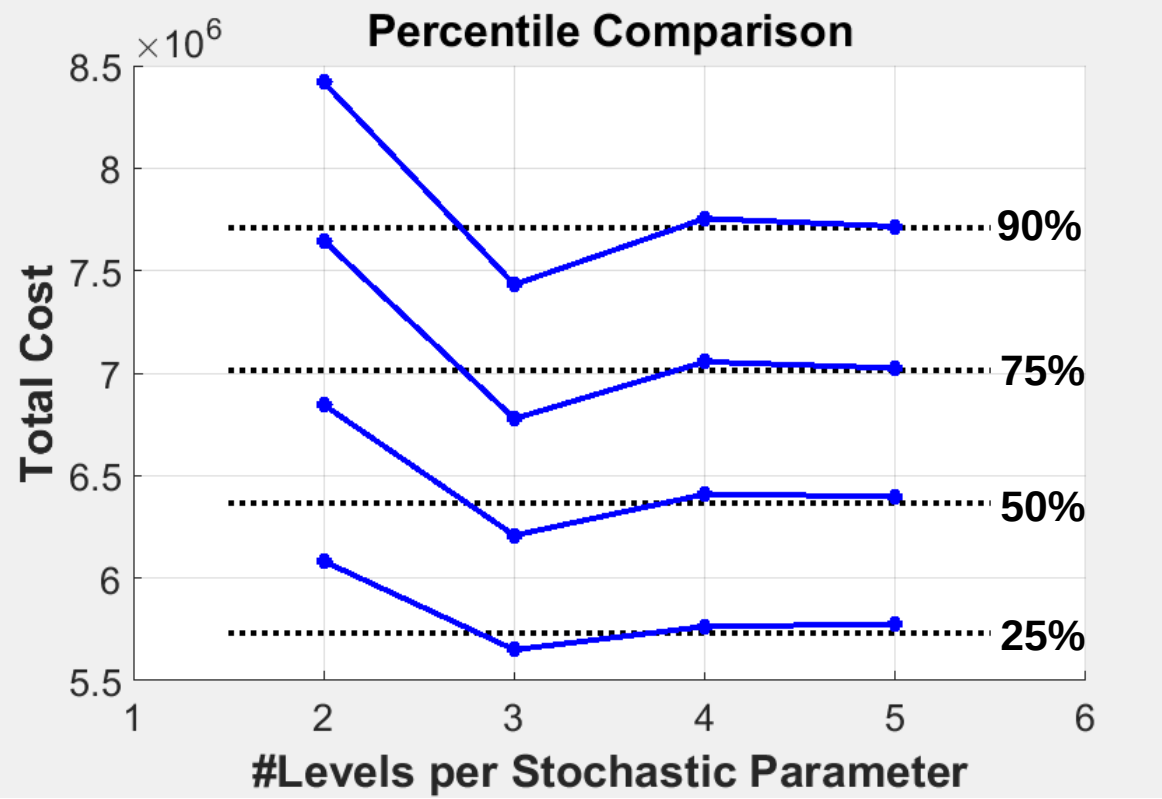


As expected, the agreement between the two approaches increases with the quantity of training data…however, extra data may not always be available.

Assemble the Distribution

The probability density distribution of total cost is assembled by creating a histogram of the scenario ensemble results, and then normalizing.

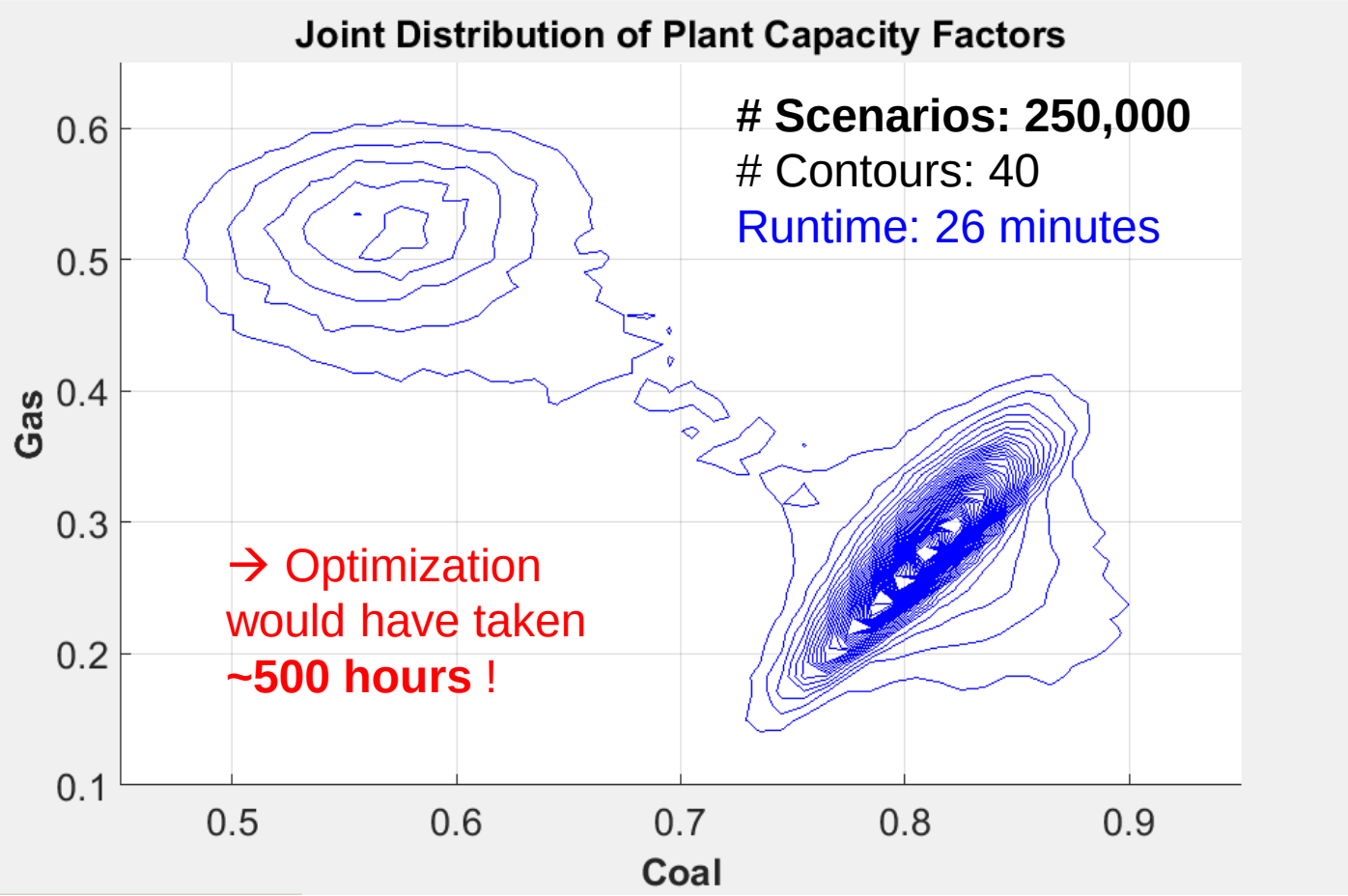


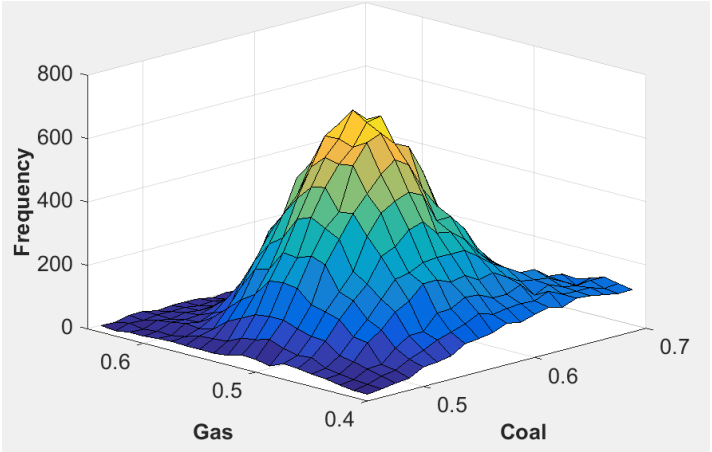


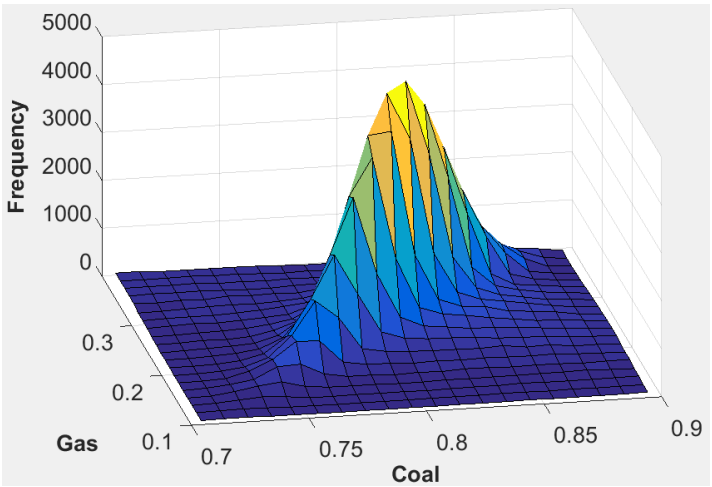
Percentile values are used to compare the two distributions, and can be used to guide the selection of levels according to any a priori accuracy requirements.

Exercise the Model

The model can now be used to interrogate questions involving a large number of scenarios…



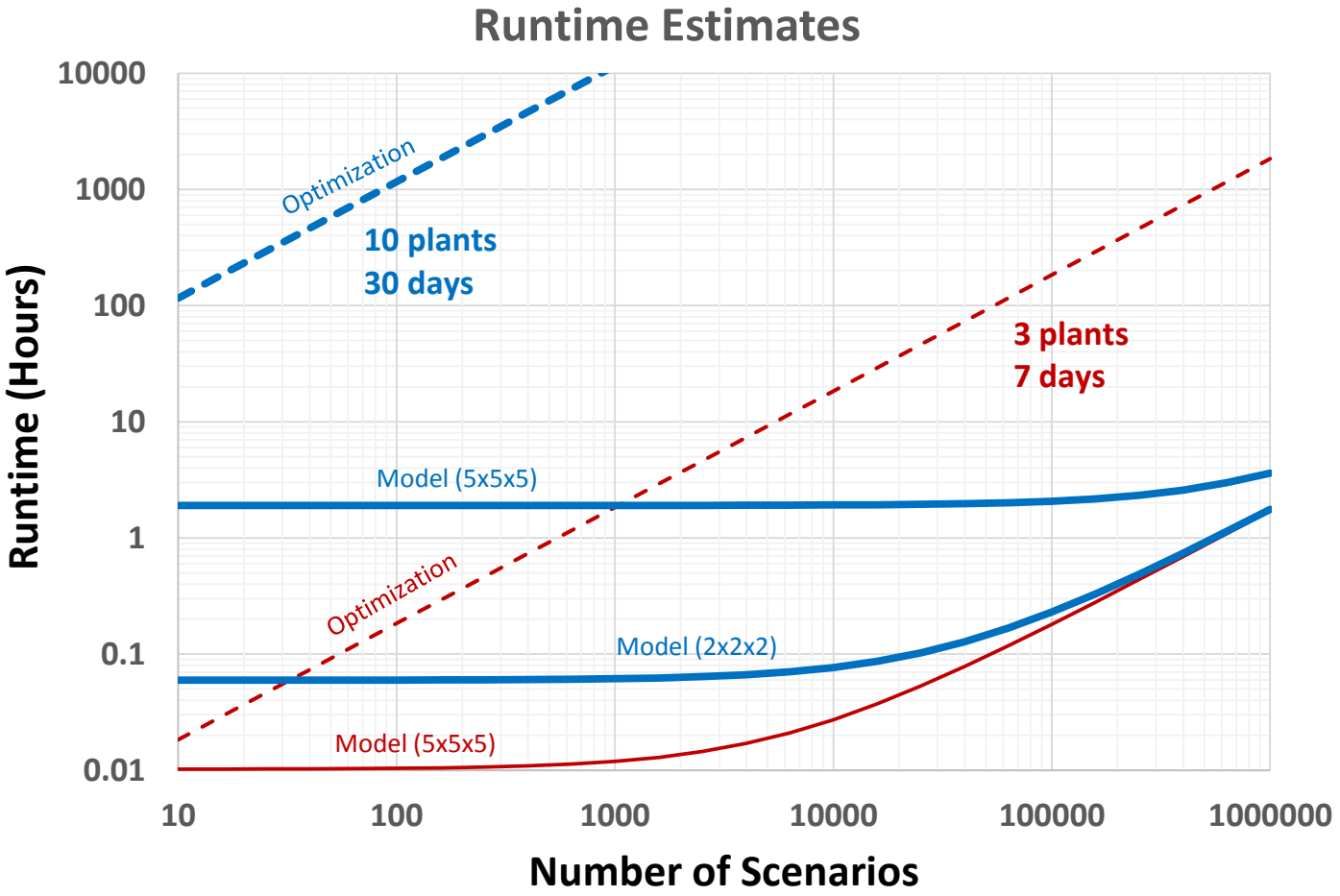




…and develop a resolved probability density distribution, within a reasonable time.

Runtime Analysis

A parametric study was conducted to estimate computational runtime scaling relationships.



For complex optimizations having long independent scenario runtimes, the modeling approach offers a practical means of estimating the probability distribution of outcomes.

Summary

A density estimation method for complex optimization problems was presented which can improve operational decision-making under uncertainty.

- Trades off individual scenario optimality for ensemble resolution
- Broad applicability:
  - Plants: power, chemicals, manufacturing
  - Resources: oil & gas, water, agriculture
  - Transportation networks & logistics
  - Finance, insurance, and business operations
- Virtual and/or actual data can be used
- Bonus: model can be used to create an intelligent initial guess for a global optimizer

Future Work

- Accuracy tradeoffs at the individual scenario level vs the ensemble level

Acknowledgment

- Szilard Nemeth – original creator of the MATLAB Unit Commitment demo

Interested to learn more, discuss, engage, collaborate?
www.mathworks.com/services/consulting/
james.cross@mathworks.com

Thank you for your attention.