LMI
GOVERNMENT CONSULTING

THE OPPORTUNITY TO MAKE A DIFFERENCE HAS NEVER BEEN GREATER

Cost Estimation as a Linear Programming Problem

2009 ISPA/SCEA Annual Conference
St. Louis, Missouri

Kevin Cincotta
Andrew Busick

Acknowledgments

- The author wishes to recognize and thank the following individuals and institutions for their contributions:
  - The General Services Administration (GSA), for sponsorship of research
  - Mr. Jeremy Eckhause, LMI, for original coding necessary to perform optimizations

Agenda

- **Background**
- Setting up the Problem
- Solving the Problem
- Adding Uncertainty
- Conclusions/Next Steps

Background

- In a typical hardware cost estimating problem, we know *how many* of each commodity we would like to buy, but we don't know *how much each one will cost*
  - Even if we know the cost of the first item, we may not know the cost of all other items
    - Learning curves
    - Economies/Diseconomies of scale, etc.
  - Even if we know the cost of all items, we may not know the total system cost
    - Integration costs
    - Ambiguous loading factors for indirect/infrastructure costs

None of these are problems for us!

Background:
A New Type of Problem

- The General Services Administration (GSA) purchases commercial-off-the-shelf (COTS) items from vendors, where:
  - There is no learning
  - There are no volume discounts
  - All prices are set by vendors' GSA schedules
  - There is no uncertainty as to how much each item will cost
  - We know ahead of time how many of each item we need
- But this doesn't mean we are home free!
  - We may have contractual obligations to give a specific percentage of sales to "small" vendors
  - Further, we wish to maximize value to GSA, and to the taxpayer (i.e., do not overpay)

How many of each item should we order from each vendor, and what will it cost us?

Agenda

- Background
- **Setting up the Problem**
- Solving the Problem
- Adding Uncertainty
- Conclusions/Next Steps

Setting up the Problem:
Not Your Usual Dog and Pony Show

- This is not a cost estimating problem in the traditional sense
  - All unit costs (and *aggregate* quantities) are known, but total cost is unknown
  - Traditional CERs would not help
    - No applicable regressors, no analogous “systems”
  - Cost as an Independent Variable (CAIV) or “Design-to-Cost” would not help
    - No performance metric against which to trade cost; cost and performance are initially assumed to be equal
    - A problem with n vendors would require visualizing a n-dimensional tradespace (not easy at the enterprise level)
  - We don’t simply need to know how much something will cost—we need to know what to do (how much to order, by vendor)
    - Traditional “Analysis of Alternatives” approaches do not help, because there is no finite list of well-defined courses of action

Setting up the Problem: Cost as a Linear/Integer Program

- **Linear programming** is a technique for optimization of a linear objective function, subject to linear equality and inequality constraints [1]
  - When all of the unknown variables are integers, the problem is called an integer programming (IP) problem
- **IP problems** are computationally more complex than their noninteger analogues
  - Discrete (not continuous) set of possible values for each decision variable
  - Cannot use traditional calculus to find noninteger answers, then round
  - Some IP problems are **infeasible** (or are feasible, but with no provable global optimum solution)

---

[1] *Wikipedia*

IP Setup for the GSA Problem: Definitions and Objective Function

- Assume that there are v vendors, each of which offer each of n items at prices p_{ij}
- Let q_{ij} = the quantity we order from vendor i of item j. **This implies (v * n) decision variables.**
- Let d_j = total demand for item j (known)
- Let a_i = required allocation (as a percentage of total sales) to be given to vendor i (also known)
- **Objective function**: total cost = $\Sigma\, p_{ij} * q_{ij}$, i.e. the sum of price * quantity across all items and vendors
- **Constraints**:
  - ☐ $\Sigma\, q_{ij} = d_j$ for all j (must exactly meet demand for each item)
  - q_{ij} are whole numbers for all i, j
  - $q_{ij} = 0$ whenever p_{ij} is undefined (can’t order item if unavailable)
  - $p_{ij} * q_{ij} \geq a_{i*}\, \Sigma\, p_{ij} * q_{ij}$ for every i (every vendor must receive at least their allocation, as a percentage of total sales)

Example: Given These Unit Prices and Demands, How Much Should We Order? What Will it Cost?

NSN\Vendor	A	B	C	D	Total Demand
1	$ 2.35	$ 2.10	$ 1.63	$ 1.60	4513
2			$ 3.64		7345
3	$ 3.26	$ 3.01	$ 2.54	$ 2.55	9653
4	$ 1.24		$ 1.02	$ 1.03	8088
5	$ 4.41	$ 4.16	$ 3.69	$ 3.68	5246
6	$ 3.27	$ 3.52	$ 3.05	$ 3.08	3106
7	$ 3.73		$ 3.01	$ 3.04	1219
8	$ 4.36		$ 4.14	$ 4.10	4442
9			$ 3.16		1364
10			$ 32.01	$ 32.00	1678
11	$ 10.46	$ 10.21	$ 9.74	$ 9.77	2267
12	$ 4.56	$ 4.81	$ 4.34	$ 4.37	3773
13	$ 3.62	$ 3.37	$ 2.90	$ 2.87	8338
14	$ 5.94	$ 6.19	$ 5.72	$ 5.71	3950
15	$ 3.95	$ 3.70	$ 3.23	$ 3.19	7682
16	$ 13.14	$ 13.39	$ 12.92	$ 12.95	2944
17			$ 19.30	$ 19.30	1645
18	$ 1.94		$ 1.72	$ 1.68	1667
19	$ 6.59	$ 5.24	$ 4.77	$ 4.74	2069
20	$ 3.03	$ 3.28	$ 2.81	$ 2.77	2819

Agenda

- Background
- Setting up the Problem
- **Solving the Problem**
- Adding Uncertainty
- Conclusions/Next Steps

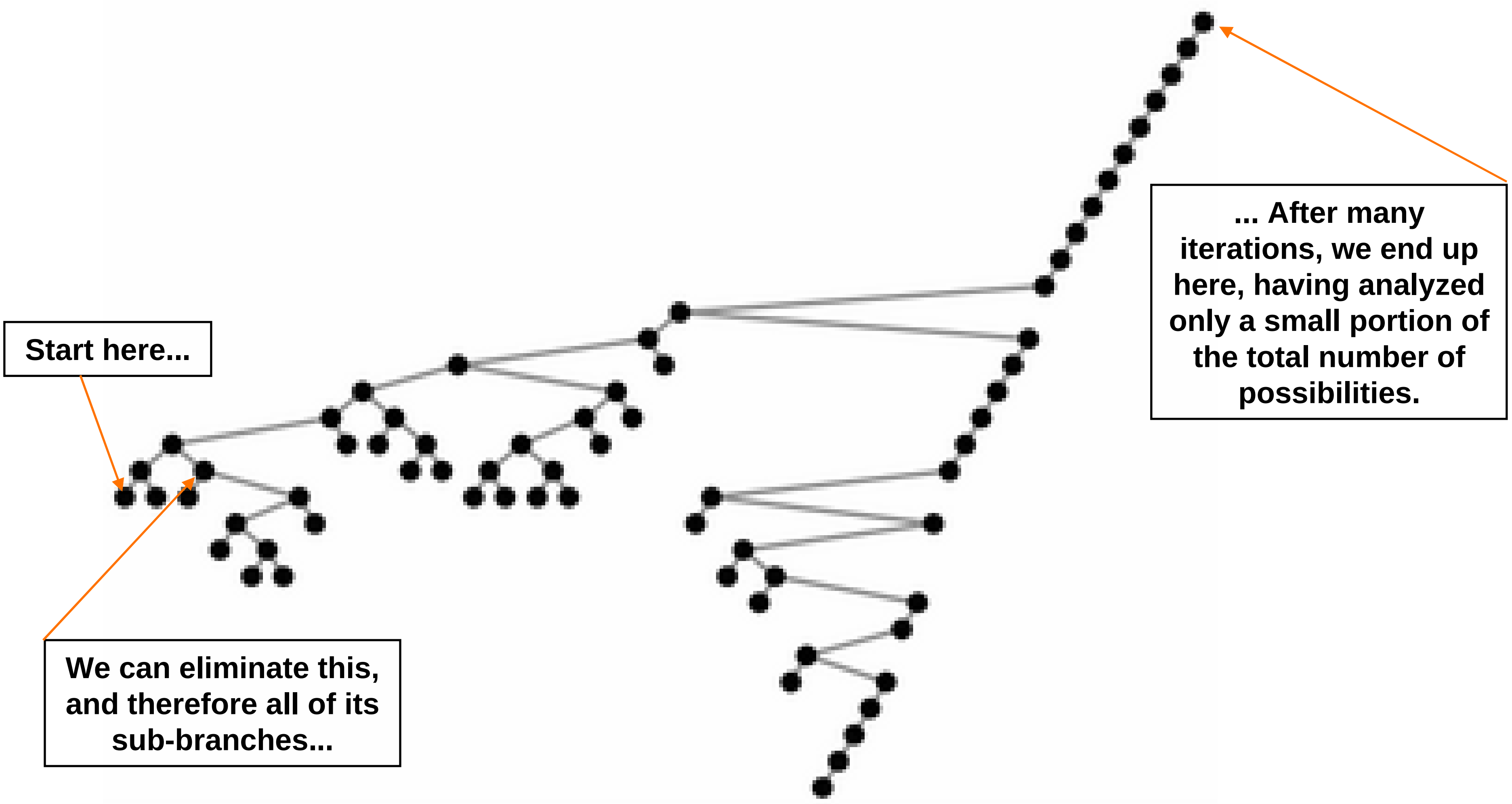
Solving the Problem, Step 1: Select the Right Tool

- The problems with Excel Solver and pencil-and-paper approaches are well-documented
- Need an operations research (OR)-type tool capable of solving constrained optimization problems using an IP framework that:
  - Allows unlimited (or very large) number of constraints and decision variables
  - Reports (optimized) values of all decision variables, *and* objective function (to support cost estimating)
  - Doesn't take all night to run
- We used Lindo Systems' LINGO 11.0
  - "A comprehensive tool designed to make building and solving linear, nonlinear, and integer optimization models faster, easier, and more efficient."[2]

---

[2] www.lindo.com

Solving the Problem, Step 2: “Branch and Bound” Technique

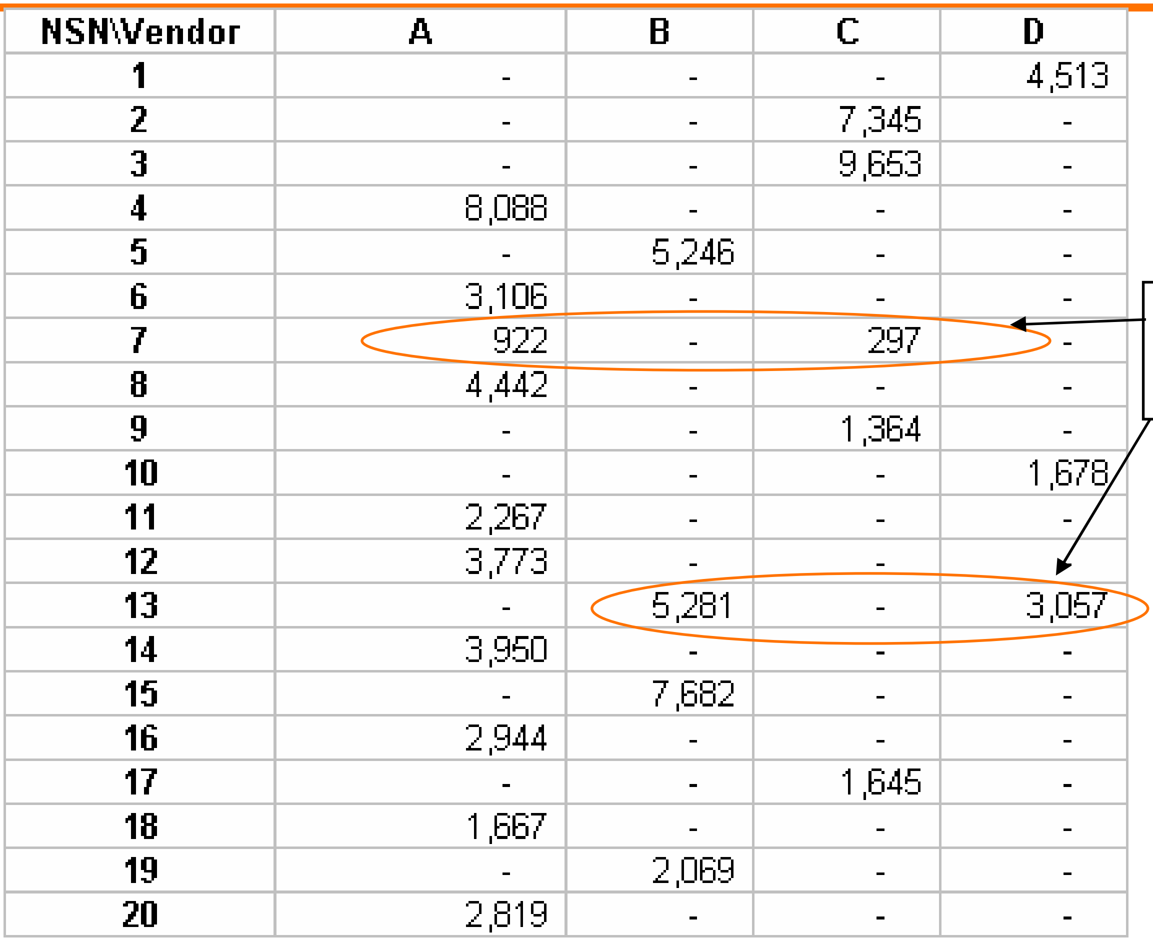


Start here...

We can eliminate this, and therefore all of its sub-branches...

... After many iterations, we end up here, having analyzed only a small portion of the total number of possibilities.

Solving the Problem, Step 3: Analyze Results



NSN\Vendor	A	B	C	D
1	-	-	-	4,513
2	-	-	7,345	-
3	-	-	9,653	-
4	8,088	-	-	-
5	-	5,246	-	-
6	3,106	-	-	-
7	922	-	297	-
8	4,442	-	-	-
9	-	-	1,364	-
10	-	-	-	1,678
11	2,267	-	-	-
12	3,773	-	-	-
13	-	5,281	-	3,057
14	3,950	-	-	-
15	-	7,682	-	-
16	2,944	-	-	-
17	-	-	1,645	-
18	1,667	-	-	-
19	-	2,069	-	-
20	2,819	-	-	-

Most items ordered from only 1 vendor; these are exceptions

Total Cost: $394,577.62

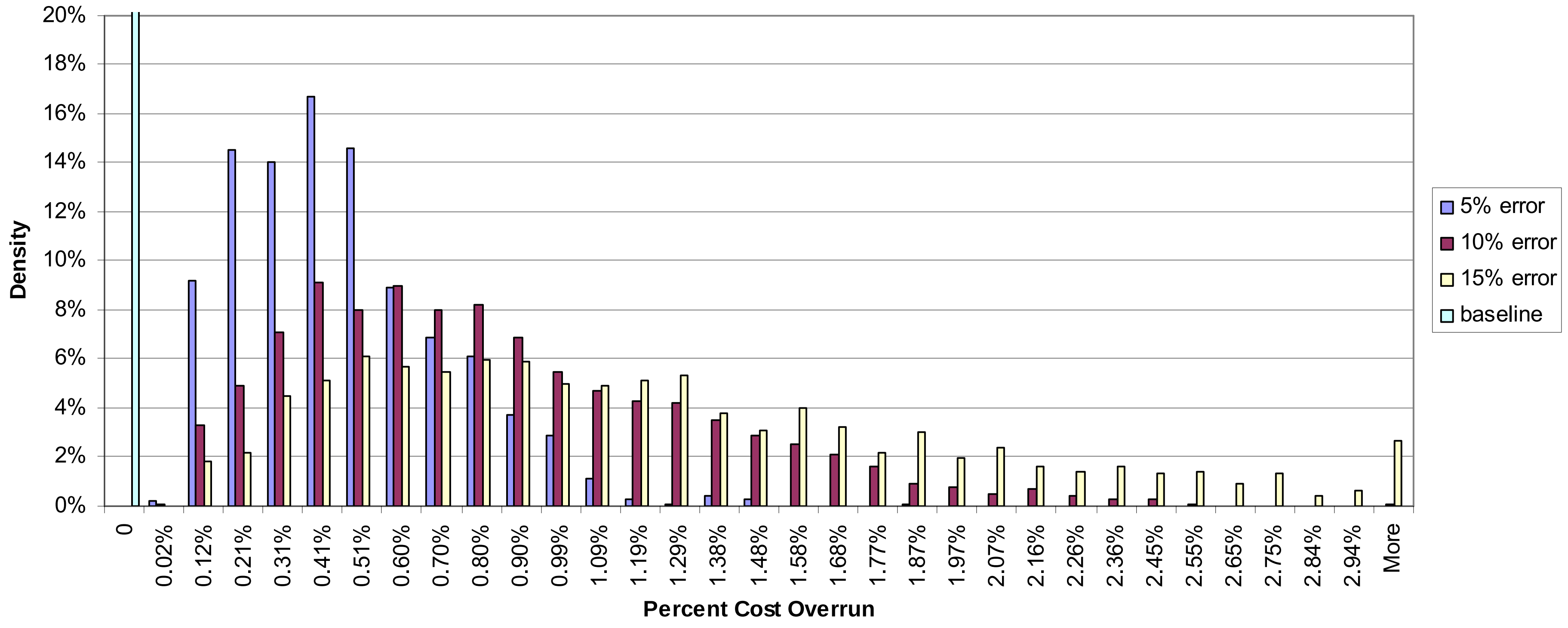
Agenda

- Background
- Setting up the Problem
- Solving the Problem
- **Adding Uncertainty**
- Conclusions/Next Steps

Adding Uncertainty

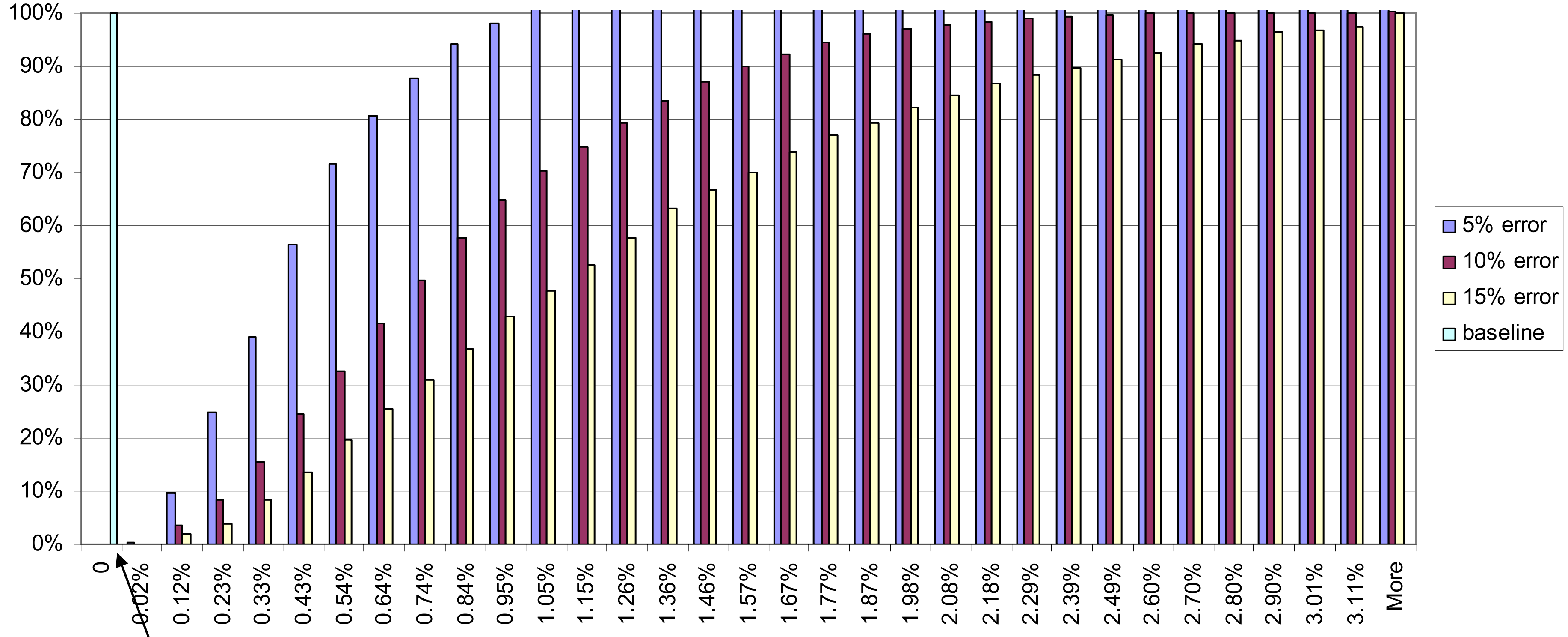
- We disturbed each order amount by an additive error term that is N(0,σ) where σ is a percentage of the (optimized) order amount
  - Only applied when more than one vendor supplies an item
  - Constrained total demand to be constant, for “apples to apples” comparison
  - Small vendor always given the positive error when possible
    - Incurring slightly greater cost is better than violating contractual obligations
- Example:
  - Optimized order quantities are 100 (Vendor A) and 80 (Vendor B),
  - $\sigma = 5\%$
  - A random variable from N(0, 5%) is drawn; we obtain 5%
  - Then, we would order 105 from Vendor A and 75 from Vendor B

Probability Density Function (pdf)



LMI GOVERNMENT CONSULTING

Cumulative Distribution Function (cdf)



Point estimate lies at 0th percentile!

Descriptive Statistics for Percent Cost Overrun

Error	mean	modal class mark	median	standard deviation	80th percentile
0% (baseline)	0.000%	0.000%	0.000%	0.000%	0.000%
5%	0.444%	0.484%	0.404%	0.272%	0.661%
10%	0.842%	0.484%	0.744%	0.519%	1.276%
15%	1.248%	0.587%	1.092%	0.795%	1.891%

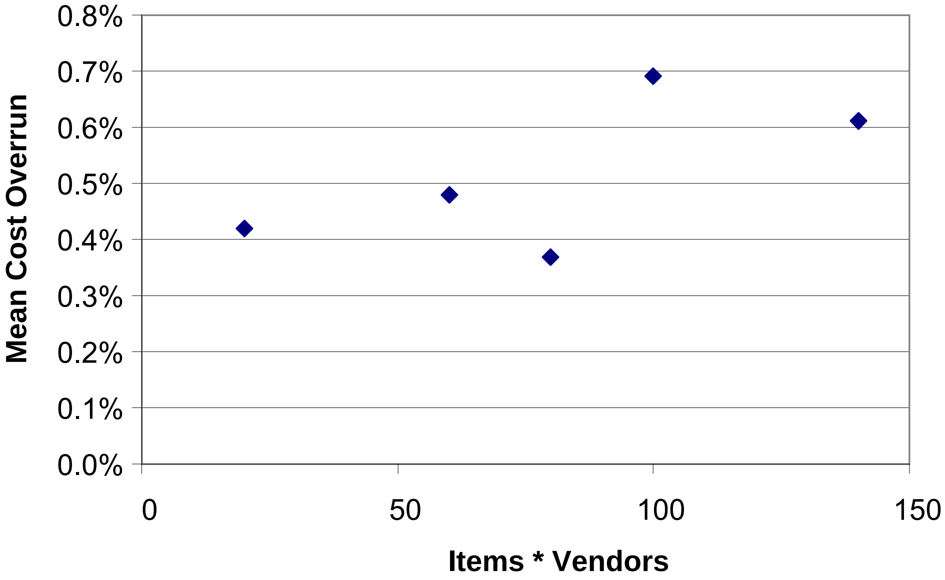
Cost overruns increase, and become more volatile, as ordering error increases

Impact of Relaxing/Tightening Constraints

Required Allocations		% increase in total cost
Vendor A	Vendor B	
15%	35%	-1.7%
16%	36%	-1.4%
17%	37%	-1.1%
18%	38%	-0.7%
19%	39%	-0.4%
20%	40%	0% (baseline)
21%	41%	0.4%
22%	42%	0.8%
23%	43%	1.2%
24%	44%	1.6%
25%	45%	2.1%

Increasing the Number of Vendors and Items

- Suppose we expand the scope:
  - 4 new scenarios
  - Increased number of vendors and items
- Hold ordering error constant (σ=5%)
- Run each scenario 1,000 times



Error	Items	Vendors	mean	median	standard deviation	80th percentile
5%	20	4	0.420%	0.382%	0.257%	0.625%
5%	60	7	0.479%	0.420%	0.297%	0.707%
5%	80	9	0.368%	0.353%	0.149%	0.482%
5%	100	10	0.692%	0.661%	0.277%	0.924%
5%	140	13	0.612%	0.593%	0.230%	0.809%

Agenda

- Background
- Setting up the Problem
- Solving the Problem
- Adding Uncertainty
- **Conclusions/Next Steps**

Conclusions

- Some cost estimating problems require solutions outside the general "toolkit" and require looking into other disciplines (e.g., Operations Research)
- A linear programming-based plan for formulating and executing your budget can help
  - Easy to program and run, if you have the right software
  - Greatly exceeds the capabilities of desktop tools (e.g. Excel Solver)
  - Gives an "ideal plan" for how much to spend, and where to spend it
- But using its results literally, without adjustment, puts you at the 0th percentile of cost
- Adjust LP-generated estimates with real world/common sense knowledge/experience, to achieve a better cost estimate

Next Steps

- Get data from GSA with actual ordering experience
  - Numbers of vendors, items, allocations, etc.
  - Use MLE or other method to estimate σ
  - Use distribution of possible values of σ to inform risk-adjusted cost estimates
  - Further analyze relationship between cost overrun and numbers of vendors, items
- Assess LP vs. other approaches (e.g., real-time simulation or automated rule)
- Solicit feedback from the cost community...

THE OPPORTUNITY TO MAKE A DIFFERENCE HAS NEVER BEEN GREATER

ACQUISITION • FACILITIES & ASSET MANAGEMENT • FINANCIAL MANAGEMENT •
INFORMATION TECHNOLOGY • LOGISTICS • ORGANIZATIONS & HUMAN CAPITAL

LMI
GOVERNMENT CONSULTING