IS IT EASIER TO *OPTIMIZE* THAN TO *ESTIMATE* IN THE PRESENCE OF INPUT MODEL RISK?

I-Sim 2017 Research Workshop
Towards an Ecosystem of Simulation Models & Data

Barry L Nelson & Eunhye Song
Department of Industrial Engineering & Management Sciences

Northwestern | McCORMICK SCHOOL OF ENGINEERING

SIMULATION OPTIMIZATION IN THE PRESENCE OF INPUT MODEL RISK

Amazon Supply Chain Optimization Technology Group

Barry L Nelson & Eunhye Song
Department of Industrial Engineering & Management Sciences
Northwestern | McCORMICK SCHOOL OF ENGINEERING

Three high-level ideas in this talk

1. Validation vs. Model Risk
   - "Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful?" --- G.E.P. Box
2. Inference about the **Simulation** vs. inference about the "**Real World**"
   - Optimization to the resolution of the model.
3. Simulation = Inputs + Logic
   - Is this perspective out of date?

I am skeptical about quantitative validation

- Model validation is "the substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application."
  - Bob Sargent, WSC '16
- There has been great work on systematic approaches, but such validation is…
  - Difficult (certainly time consuming)
  - Rarely done unless mandated (pushed to late in the project)
  - Does not tell us what to do if it "fails" (cancel the project?)
  - Does not account for your remaining exposure if it succeeds.
- We are good at hedging against risks if we can characterize "how wrong?" → **Change the focus to model risk.**

A success story: Input model risk

- We drive simulations with a set $\boldsymbol{F} = \{F_1, F_2, \ldots, F_L\}$ of input models for service times, machine failures, customer characteristics, etc.

- The simulation output depends on the models we choose

$$Y(\boldsymbol{F}) = \mu(\boldsymbol{F}) + \varepsilon(\boldsymbol{F})$$

The output could be an average, an indicator variable, a variance, a sample quantile, etc.

- We use fitted distributions $\widehat{\boldsymbol{F}}$ to approximate the true “real world” so there is clearly risk:

$$Y(\widehat{\boldsymbol{F}}) = \mu(\widehat{\boldsymbol{F}}) + \varepsilon(\widehat{\boldsymbol{F}})$$

Two schools of thought

We know $\widehat{\mathrm{F}}$ is wrong, so what do we do?

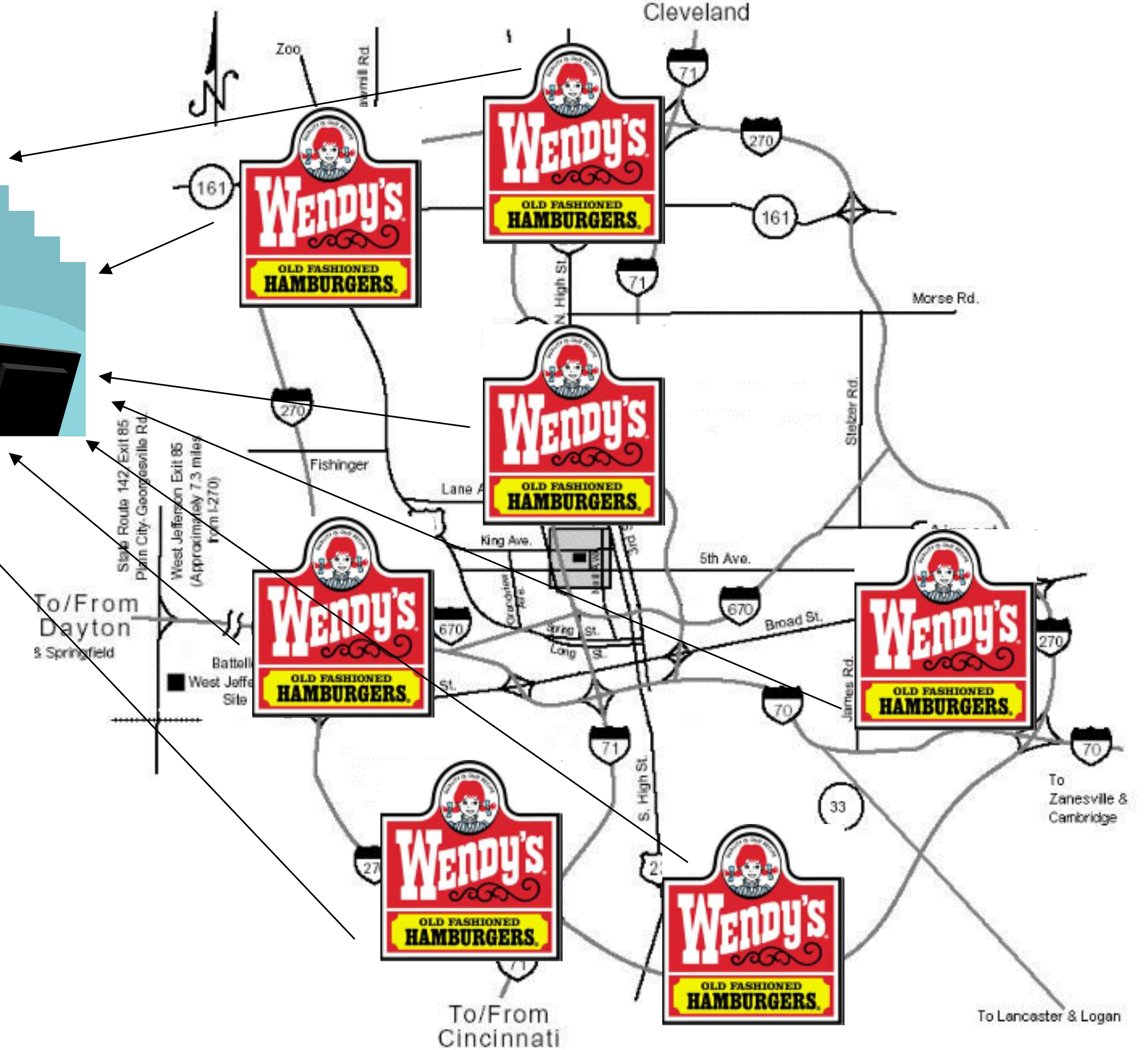
Barton, Nelson, Schruben, Song et al.: Try to propogate the uncertainty in $\widehat{\mathbf{F}}$ to the output $Y(\widehat{\mathbf{F}})$ and quantify it.

Ex: Estimate $\mathrm{Var}[Y(\widehat{\mathbf{F}})]$ including uncertainty about $\widehat{\mathbf{F}}$.

Hong, Lam et al.: Choose a defensive $\widehat{\mathbf{F}}$ that protects you against the risks that matter to you.

Ex: $\widehat{\mathbf{F}}$ that gives the worst-case expected response time within some uncertainty set consistent with the real-world data.

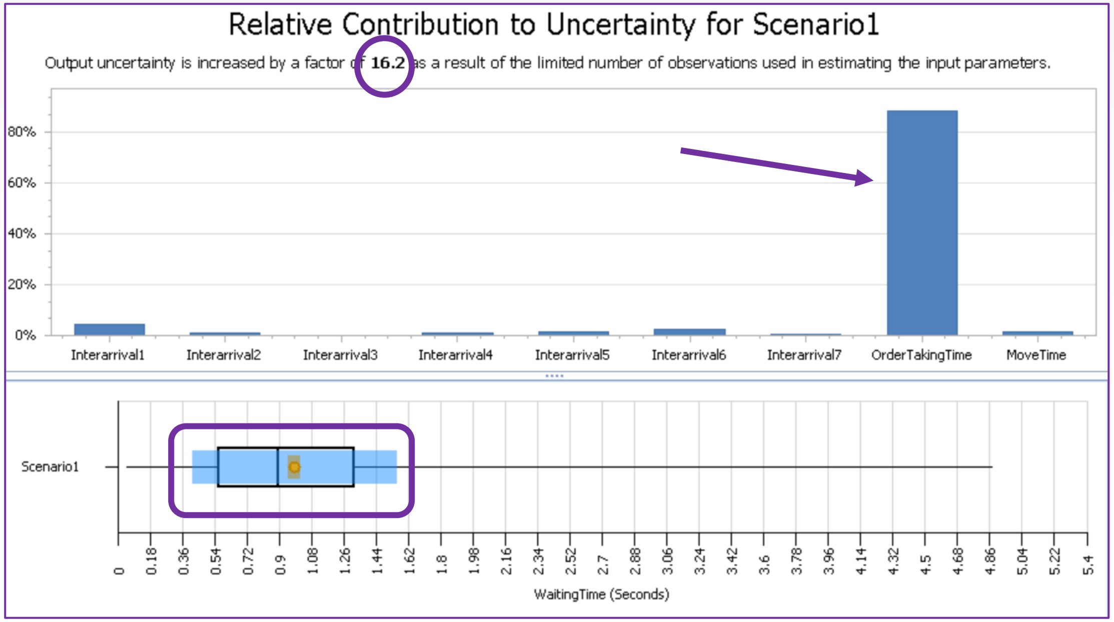
Illustration



Can we attain average response < 5 seconds with fewer agents?

Inference about the "real world"

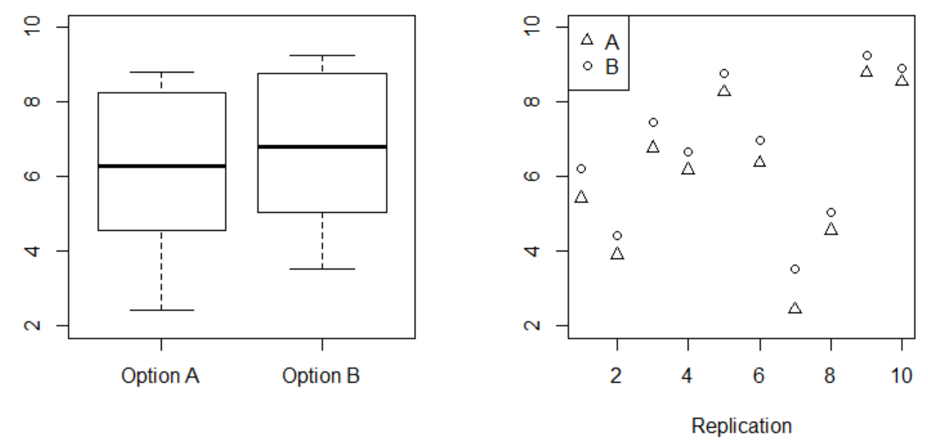
- Simio "Sample Size Error" feature based on work from S&N [*IIE Transactions* **47** (2015), 1-17]:



- **Real-world model risk is 16.2 times larger than simulation error! This seems discouraging.**

Changes matter in model risk

- Common random numbers is based on natural variability affecting all options similarly.
- This idea is more general than just the random numbers:
  - Common input models
  - Common logical relationships

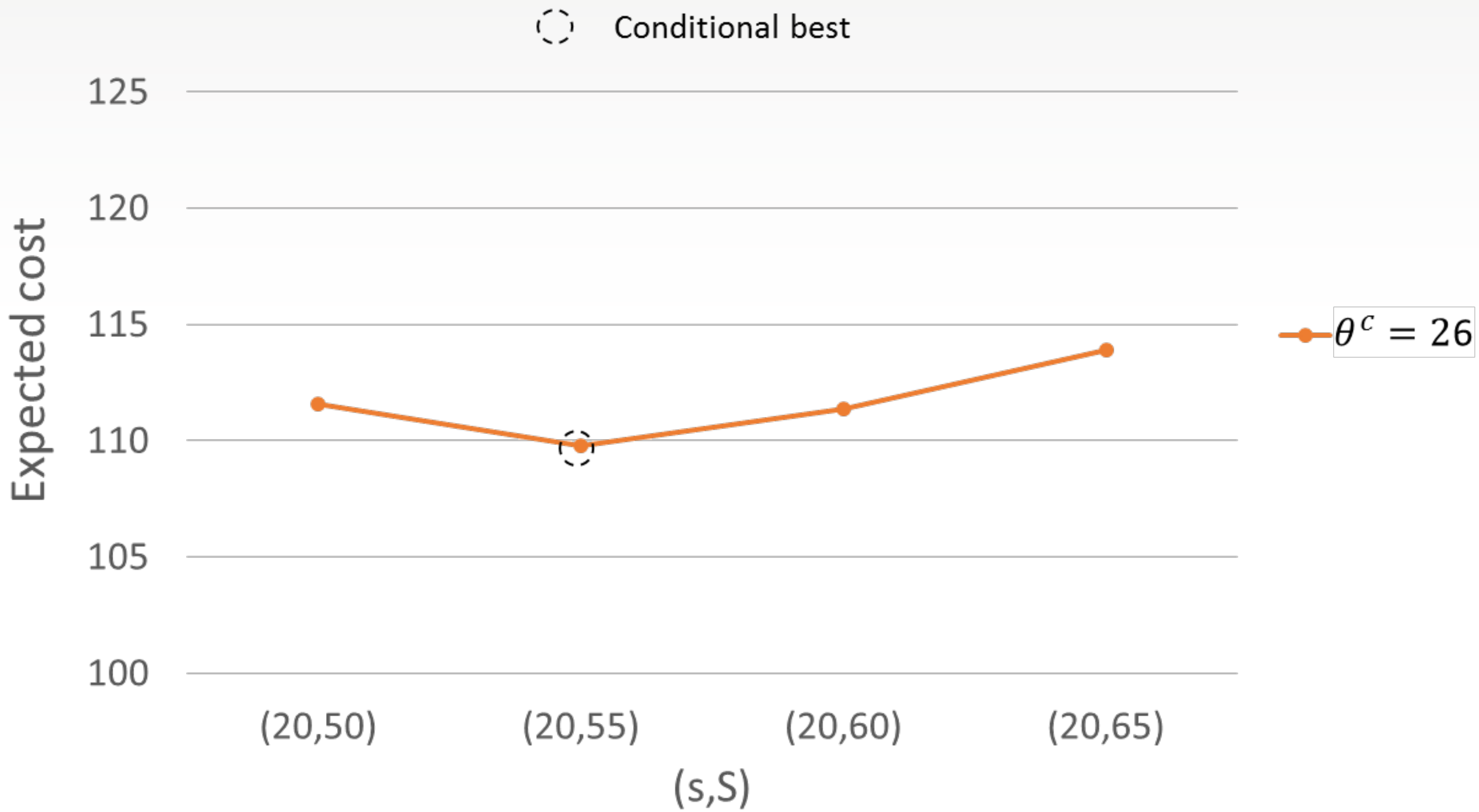


- When **optimizing**, model risk arises when systems are affected **differently** by uncertainty in random numbers, input distributions or system logic.

This difference is what we have to quantify, and it is easier to do so when the risks are common.

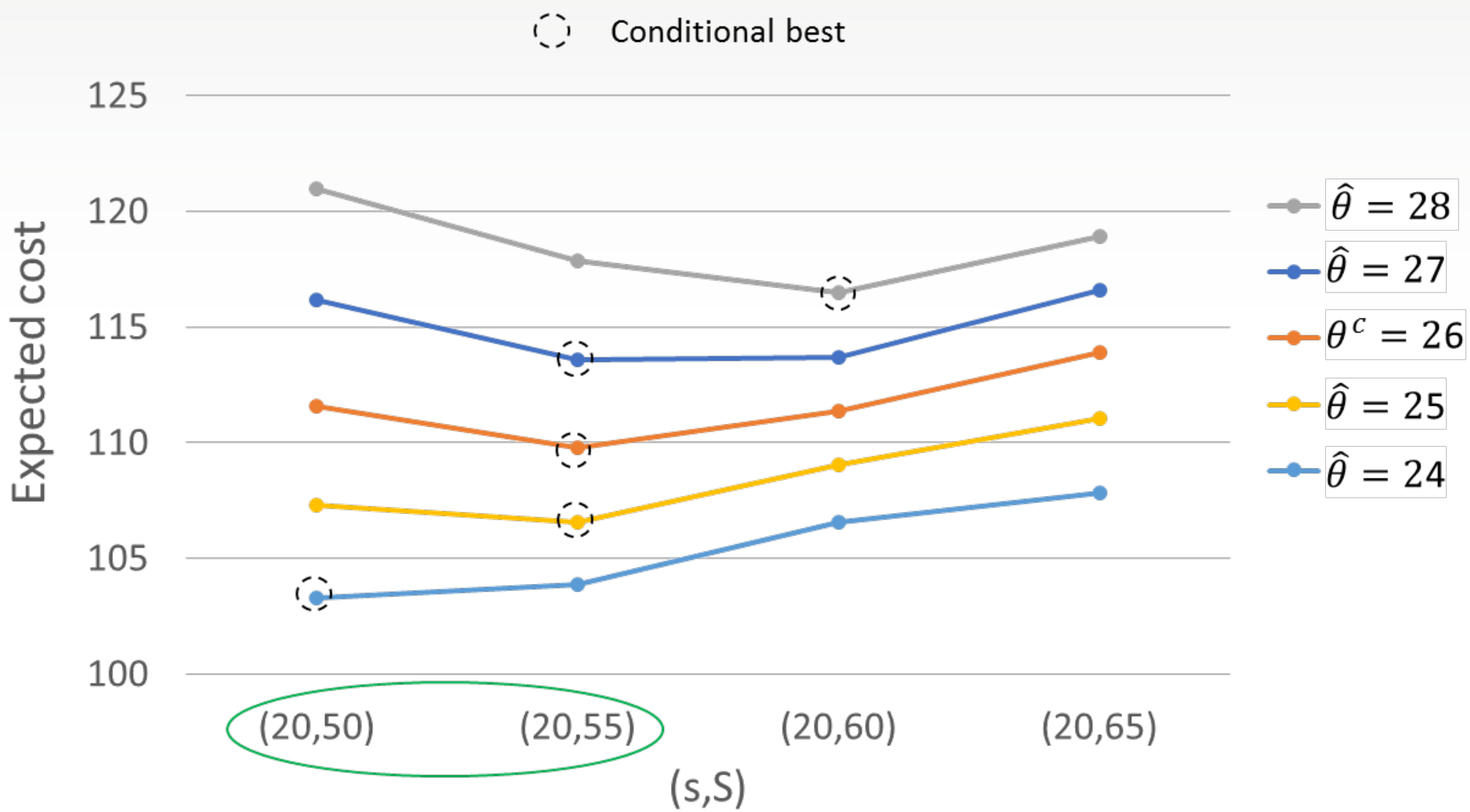
Motivating example: (*s*, *S*) inventory

- Four candidate policies (s, S): $(20, 50), (20, 55), (20, 60), (20, 65)\}$
- Unknown true demand $\text{Poisson}(\theta^c = 26)$



Impact of input uncertainty

- How *far* $\hat{\theta}$ is from θ^c and how *differently* the systems are affected by θ matters. This is the **common input data** (CID) effect.



Optimizing to the resolution of the model

- **To make statements about the "real-world" optimal, we have to acknowledge the limits of the simulation model (IMHO).**
  - Averaging over the sampling distribution or posterior of $\widehat{F}$ is **not** the same thing.
  - We may not be able to identify the optimal, even if it is unique.
- **Change the emphasis from *selection* to *comparisons*.**
  - Identify the systems that **cannot be separated** from the best, given the resolution of the input models.
  - We might then apply robust/defensive selection to this subset: much less conservative.
- **To make the comparisons as sharp as possible we want to exploit the common input data (CID) effect.**

Now the technical program

- Our model for the CID and CRN effect is

$$Y_i(\hat{\theta}) - Y_j(\hat{\theta}) = \mu_i(\hat{\theta}) - \mu_j(\hat{\theta}) + \varepsilon_i(\hat{\theta}) - \varepsilon_j(\hat{\theta})$$

$$= \mu_i(\theta^c) - \mu_j(\theta^c) + \boxed{b_i(\hat{\theta}, \theta^c) - b_j(\hat{\theta}, \theta^c)} + \varepsilon_i(\hat{\theta}) - \varepsilon_j(\hat{\theta})$$

$$\approx \mu_i(\theta^c) - \mu_j(\theta^c) + \boxed{(\beta_i - \beta_j)^T (\hat{\theta} - \theta^c)} + \varepsilon_i(\hat{\theta}) - \varepsilon_j(\hat{\theta})$$

- We want to form $1 - \alpha$ simultaneous MCB confidence intervals

$$\mu_i(\theta^c) - \max_{j \neq i} \mu_j(\theta^c) \in [L_i, U_i], \forall i$$

- To do this we need to capture the joint distribution of $(\hat{\beta}_i - \hat{\beta}_j)^T (\hat{\theta} - \theta^c)$ and $\varepsilon_i(\hat{\theta}) - \varepsilon_j(\hat{\theta})$ across all $i \neq j$.

Input-Output Uncertainty (IOU) Comparisons

- $\left(\hat{\beta}_i - \hat{\beta}_j\right)^T \left(\hat{\theta} - \theta^c\right)$ is the hard part.

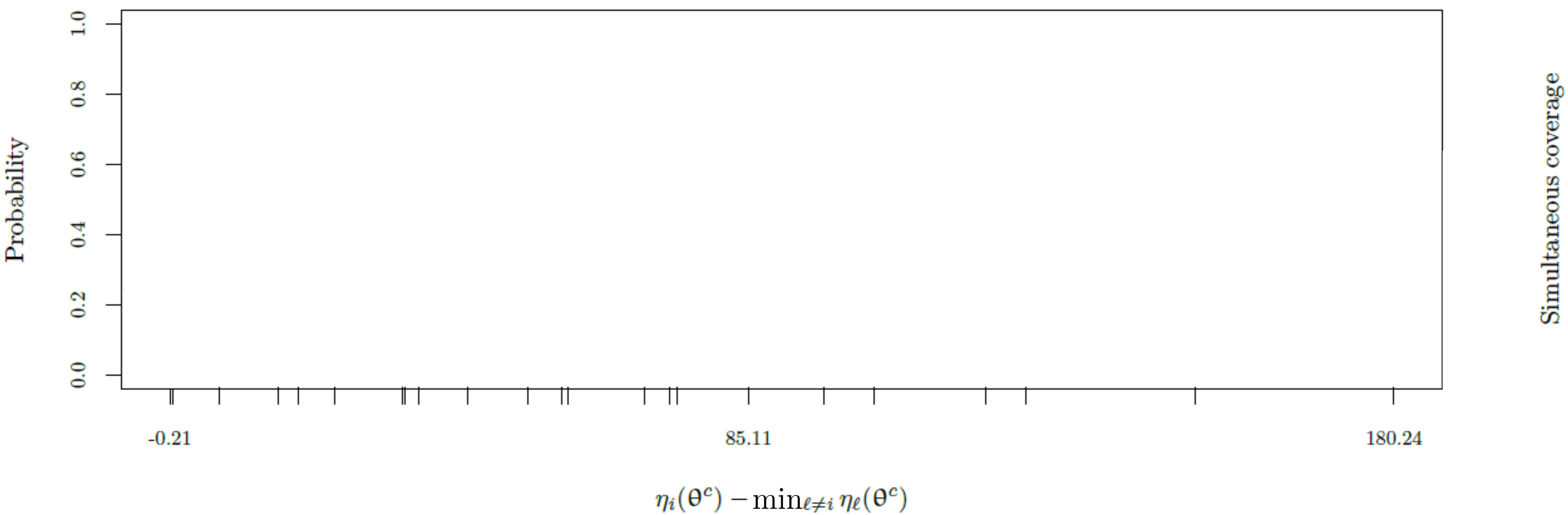
- **Plug-in IOU-C**
  - Insert your favorite gradient estimator $\hat{\beta}$ and use the asymptotic normal distribution of $\left(\hat{\theta} - \theta^c\right)$ from MLE.
- **All-in IOU-C**
  - Use asymptotically normal regression estimator $\hat{\beta}$ and use the asymptotic normal distribution of $\left(\hat{\theta} - \theta^c\right)$ from MLE and solve

$$
\begin{aligned}
&\min \ \left(\beta_i - \beta_j\right)^T \left(\hat{\theta} - \theta^c\right) \\
&\text{subject to } \mathcal{B}_i \in \mathrm{CR}_1 \text{ and } \left(\hat{\theta} - \theta^c\right) \in \mathrm{CR}_2
\end{aligned}
$$

- Both can be shown to be asymptotically $1 - \alpha$ when m, n, B get large in just the right way.

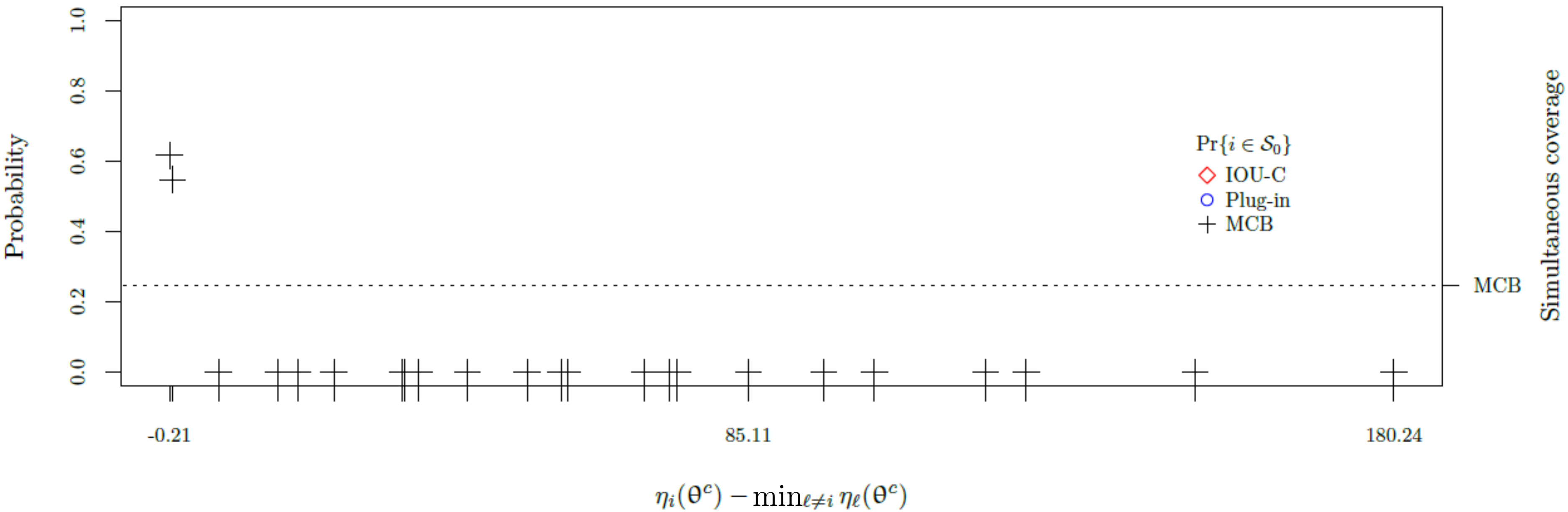
Empirical performance

(s, S) inventory problem with 23 solutions, three input processes (demand, lead time, and yield), where $m = 100$ for each, and $1 - \alpha = 0.9$



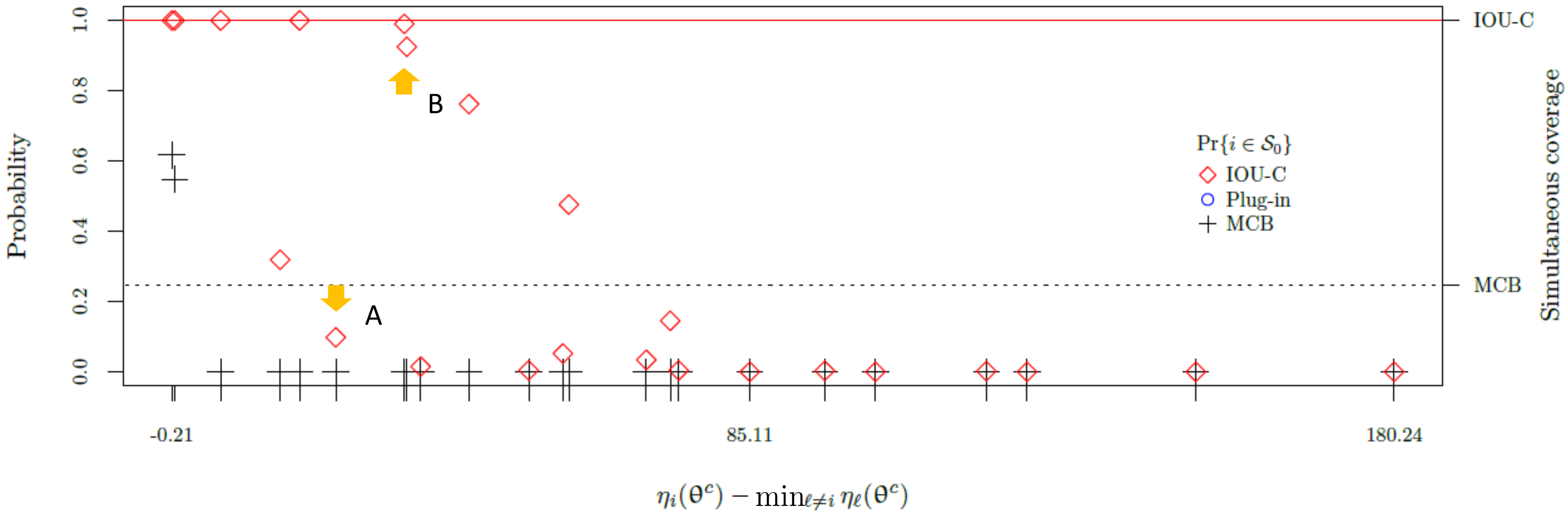
Empirical performance

(s, S) inventory problem with 23 solutions, three input processes (demand, lead time, and yield), where $m = 100$ for each, and $1 - \alpha = 0.9$



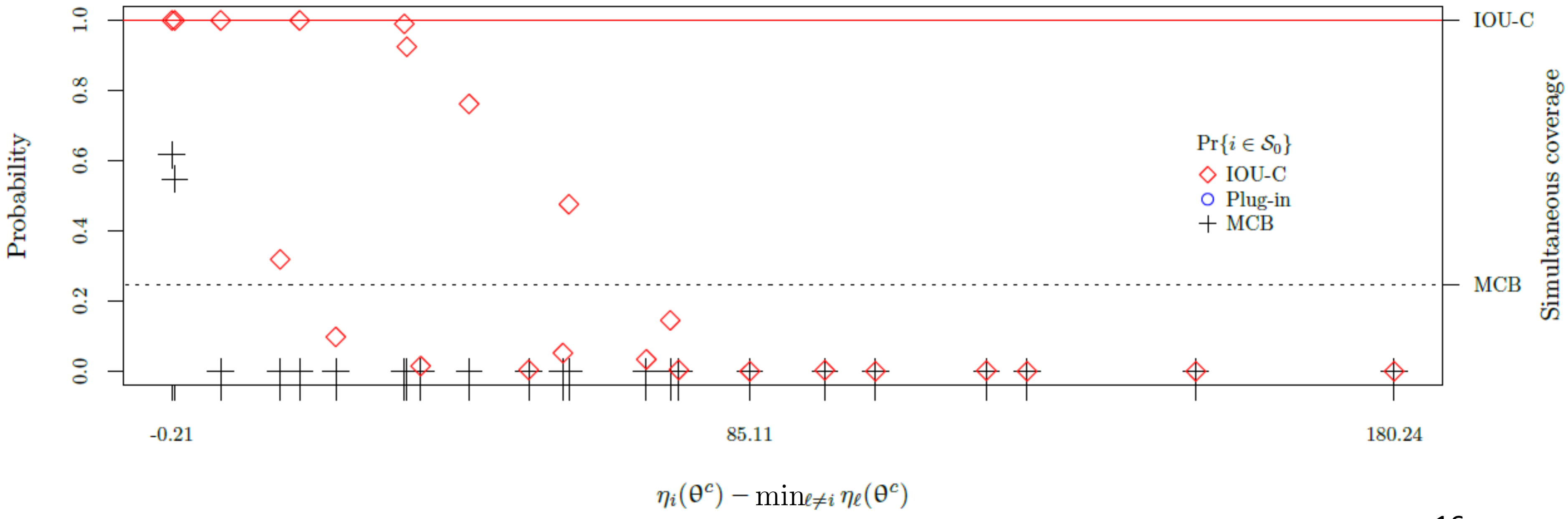
Empirical performance

(s, S) inventory problem with 23 solutions, three input processes (demand, lead time, and yield), where $m = 100$ for each, and $1 - \alpha = 0.9$



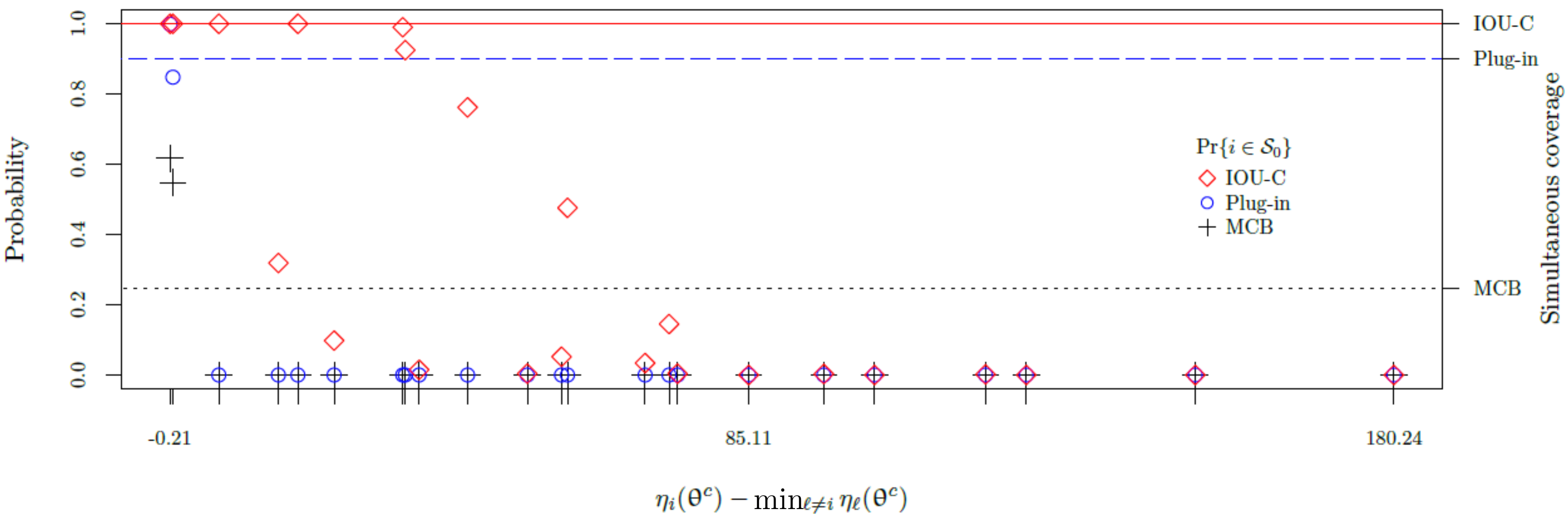
Empirical performance

(s, S) inventory problem with 23 solutions, three input processes (demand, lead time, and yield), where $m = 100$ for each, and $1 - \alpha = 0.9$



Empirical performance

(s, S) inventory problem with 23 solutions, three input processes (demand, lead time, and yield), where $m = 100$ for each, and $1 - \alpha = 0.9$



Simulation = Inputs + Logic

- I have said this for years.
  - The inputs are the statistical part, which means we can estimate error.
  - The logic is the "art" part that is either true-enough or not, right level or wrong level.
    - Tocher (1963) *The Art of Simulation*
  - I have acted like there is a clear distinction.
- **Can we extend the definition of "input" so we can attach statistical uncertainty to the logical model?**

Logic revisited in the age of video & ML

- **Suppose we have a video of the current system.**
- The “logic” could be “machine learned” from dissecting it.
  - Fixed objects, dynamic objects, what follows what and with what regularity.
  - Analysis of hours of video would reveal inconsistencies, worker differences, rare events, etc. that I would never observe.
  - Therefore the logic becomes more like a statistical model.
- **Questions**:
  1. What do we do with this data?
  2. Have we reduced simulation to data analytics?

Regression analogy: $Y = l(x) + \varepsilon$

Think of Y as a system output, $\mathbf{x}$ as a vector of controllable decision variables, and ε as system noise.

Traditional Simulation: We "artfully" model the logic as $l(\mathbf{x}) = \mathbf{x}\beta^{\text{known}}$ and treat ε is an input model. Thus we can only quantify input uncertainty.

Parametric logic: We model the form of the logic as $\mathbf{x}\beta$ but we estimate $\widehat{\beta}$ from observation. Now we can quantify some of the logical model risk.

Nonparametric logic: We observe $(Y, \mathbf{x})$, but all we know are what are the x's. Now (maybe) the overall model risk can be quantified.

Learning simulations

- Statistical models can be learned from data.
  - We should be pushing as much of the simulation model as possible to being an "input."
- But we are interested in more than the *observable* input-output relationship.
  - Embedded in the data is a control **x** that I want to change.
- Maybe the "art" part comes in deciding what a *change in* **x** will do to the I-O relationship.
  - This might lead to a very different type of simulation model building and analysis: Modeling the impact of <u>changes</u>.

Three high-level ideas in this talk

1. Validation vs. Model Risk
   - Go, No-Go is not as useful as capturing the uncertainty.
2. Inference about the **Simulation** vs. inference about the "**Real World**"
   - Clearly we want the latter; to what resolution do we have it?
3. Simulation = Inputs + Logic
   - What we want to use our modeling skill for is representing the impact of *changes*.