NEW OPPORTUNITIES FOR FAKE DATA

Barry L Nelson
Walter P Murphy Professor Emeritus
Industrial Engineering & Management Sciences

Camm Lecture 2026, Clemson University

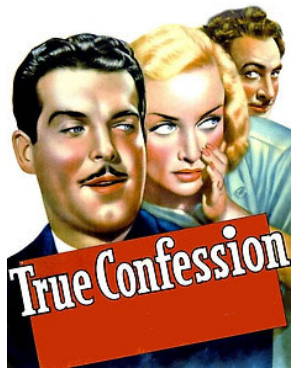
Northwestern | McCORMICK SCHOOL OF ENGINEERING

A true confession

- I have been generating fake data for over forty years, have taught hundreds of students to do the same...

  ...and have no remorse!

- I work on methodology for the design and analysis of systems that
  - are not entirely predictable;
  - are too expensive or critical to experiment on directly;
  - defy mathematical analysis; and
  - usually do yet exist (and thus there are no data).



A cute, but real, example…

- Jai Alai—handball on steroids—is a game on which there is pari-mutuel betting.
- 8 players compete, first to 7 points wins.
- Players play in order, two at a time, and hold court if they win.



A cute, but real, example…

- Jai Alai—handball on steroids—is a game on which there is pari-mutuel betting.
- 8 players compete, first to 7 points wins.
- Players play in order, two at a time, and hold court if they win.
- Clearly the first few players are at a scoring advantage, so...
- Spectacular 7 Scoring: 1 point/win for the first 7 games, 2 points/win thereafter.



Does starting position matter?

A stochastic simulation answer:

1. Line up my 8 (fake) players.
2. Assume that all players are equally good, so Probability{win point} $= 1/2$, like a coin flip.
3. Play pretend games as long as their patience lasts.
4. Record the number of wins by position.



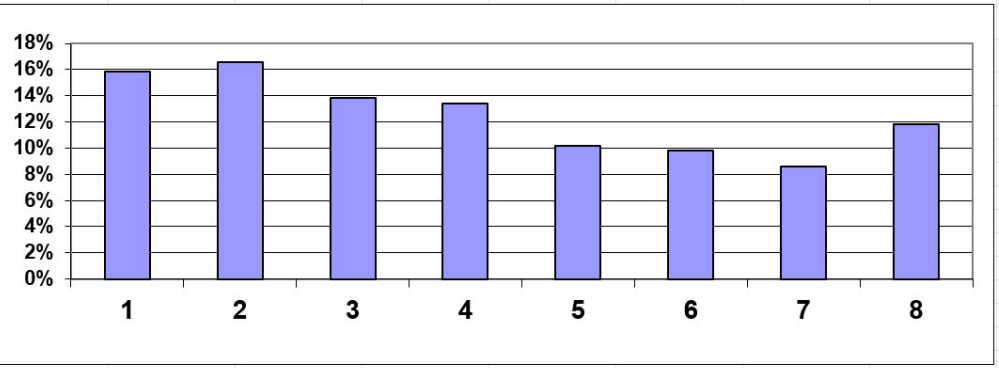
Does starting position matter?

A stochastic **computer** simulation answer:

1. Write an algorithm in a computer language that represents the rules of Jai Alai.
2. Let pseudorandom numbers stand in for the coin.
3. Run the algorithm for many games and record the number of wins by position.



	A	B	C	D	E	F	G	H	I	J	K
1	1583	1657	1382	1338	1014	982	862	1182	Wins		
2	16%	17%	14%	13%	10%	10%	9%	12%	Percentage		
3	0.7%	0.7%	0.7%	0.7%	0.6%	0.6%	0.6%	0.6%	± Error (95% CI)		
4	Player 1	Player 2	Player 3	Player 4	Player 5	Player 6	Player 7	Player 8			



Number of Games

10000

Number Played

10000

Play Jai Alai

Reset

Jai Alai Simulation
"Spectacular 7 Scoring"



Takeaways:

We can simulate (fake) relevant data that *do not exist*.

Stochastic simulation has *estimation error*.

The goal can be *optimization* rather than estimation.

Simulation takes *time*.

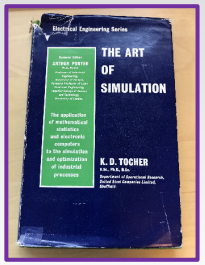
Modern simulation in Simio

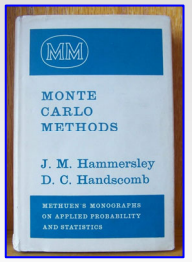


Simulation: ancient (~1963)

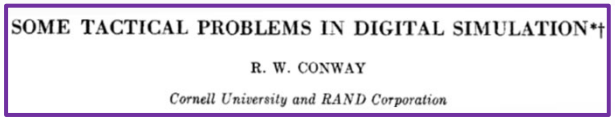
Tocher (1963) *The Art of Simulation*, English University Press

Hammersley & Handscomb (1964), *Monte Carlo Methods*, Methuen





Conway (1963) Some tactical problems in digital simulation, *Management Science* **10**, 47–61


SOME TACTICAL PROBLEMS IN DIGITAL SIMULATION*†

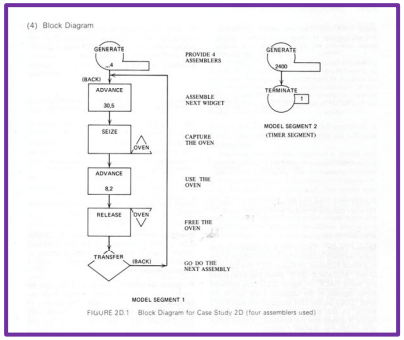
R. W. CONWAY

Cornell University and RAND Corporation

- Guided research for ~ 50 years!

Gordon's Programmable Simulation System (GPSS)

- Process-oriented simulation modeling
- Top-to-bottom eventually beaten by left-to-right :-)



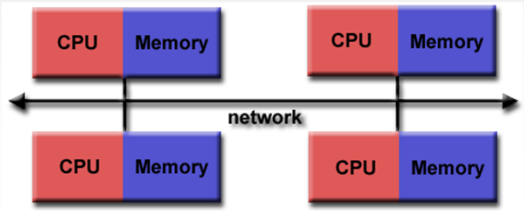
Simulation: ancient (~1963) but still useful!

A few Nelson applications:

- Bet-the-company fuel injector line for Delphi.
- Productivity of automated vs. manual manufacturing for Sloan Valve.
- Cost-effective magazine distribution supply chain for Chas Levy.
- Computer hardware migration for LexisNexis.
- Risk analysis of wind and solar energy projects for Sunairio.
- Feasibility of exotic "future mobility" options for GM.

Tactical problems not anticipated by Conway (1963)

1. Cheap parallel computing should help us do simulation optimization, but how can we use it in a statistically valid *and* computationally efficient way?

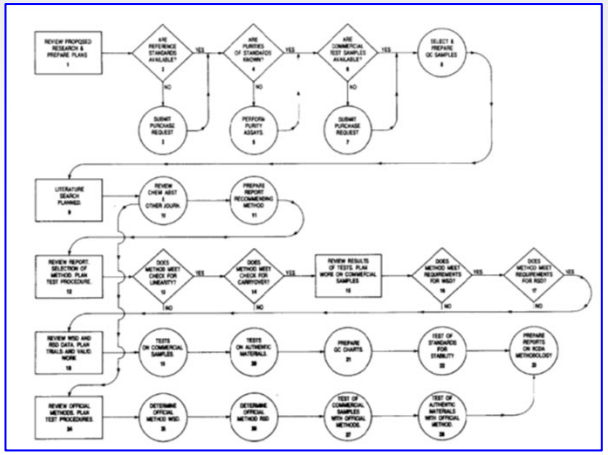


2. Motivated in part by the emergence of "digital twins," how can simulation optimization be deployed for real-time, data-driven decision making?



Illustration #1

- An FDA Research Planning Network has 28 (stochastic) activities, of which we can speed up 7 by allocating resources.
  - Implies **1,184,040** feasible allocations.
- We can simulate activity times, and thus can estimate the mean time to complete the project for any allocation.
- Which activities should we speed-up to minimize the mean time to complete the project?



Simulation optimization is *infuriating*!

The 3 errors in stochastic simulation optimization:

1. You may *never simulate* the optimal solution!
2. You may not even recognize the best solution you *did* simulate!
3. Your estimate of the performance of the solution you pick is *optimistic*!



Simulation optimization is *infuriating*!

The 3 errors in stochastic simulation optimization:

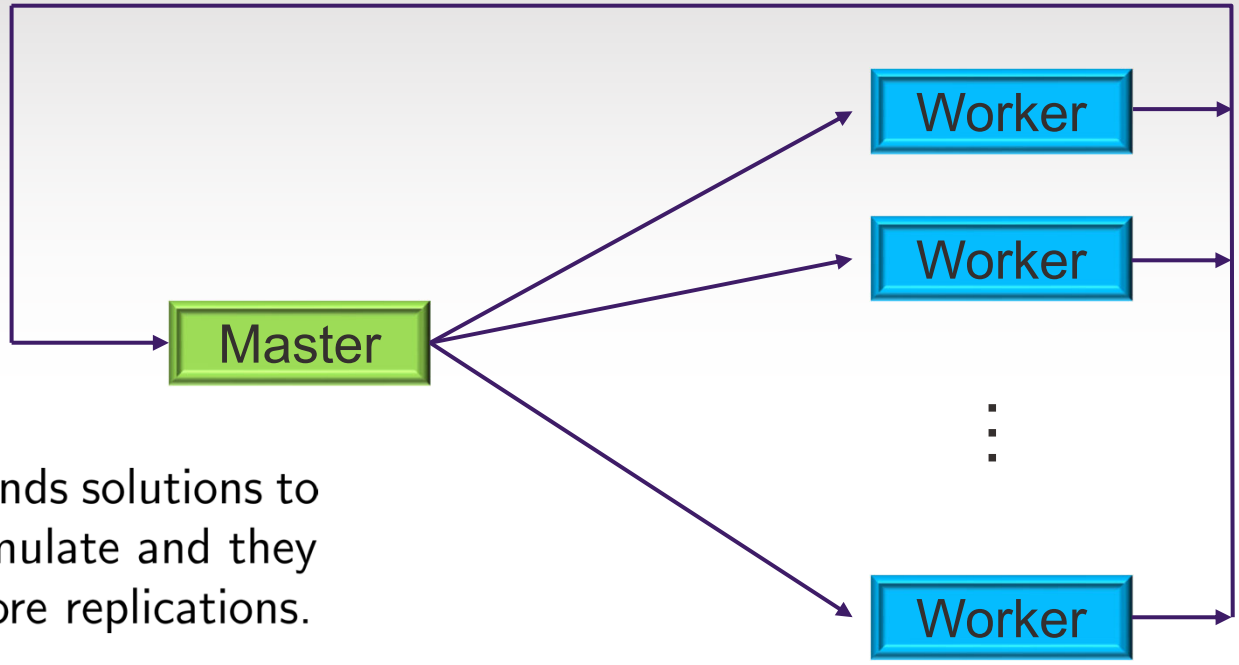
1. You may *never simulate* the optimal solution!
2. You may not even recognize the best solution you *did* simulate!
3. Your estimate of the performance of the solution you pick is *optimistic*!

However, we can control all 3 errors when we *simulate every feasible solution*.

Cheap parallel computing increases the "simulate every feasible solution" limit.



Master-worker processor set up



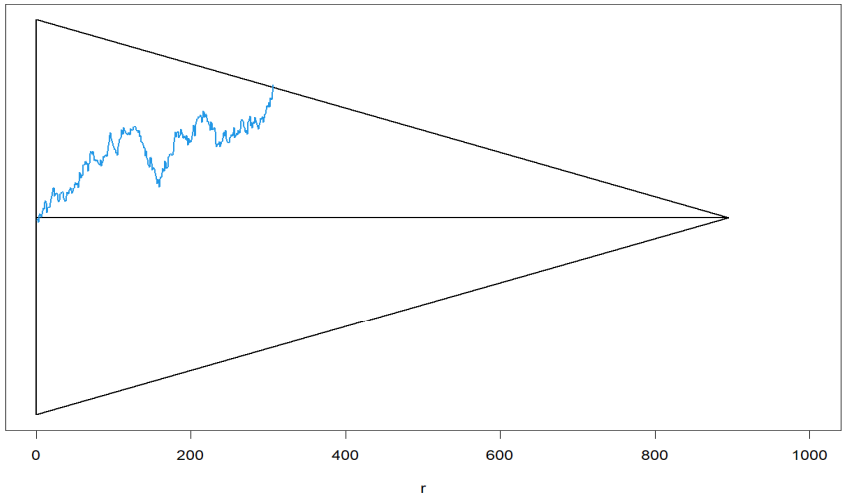
The Master sends solutions to Workers to simulate and they return 1 or more replications.

Just a tiny bit of notation

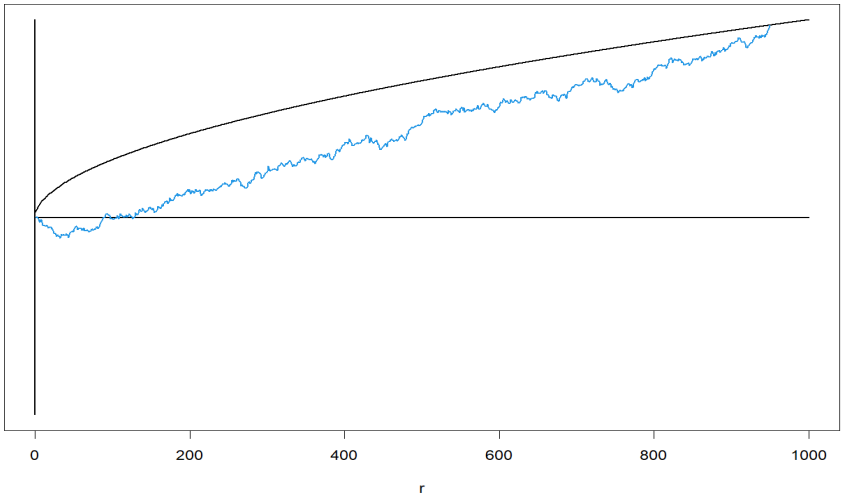
- Y_{ij} is jth i.i.d. simulation replication of feasible solution $i = 1, 2, \ldots, k$.
- $\mu_i = \mathrm{E}(Y_i)$ the expected (mean) value of solution i.
- Smaller is better, so the best is $i^\star = \mathrm{argmin}_i \, \mu_i$ with value $\mu^\star$.
- There are $p \geq 1$ parallel "processors" available.

Just a tiny bit of notation

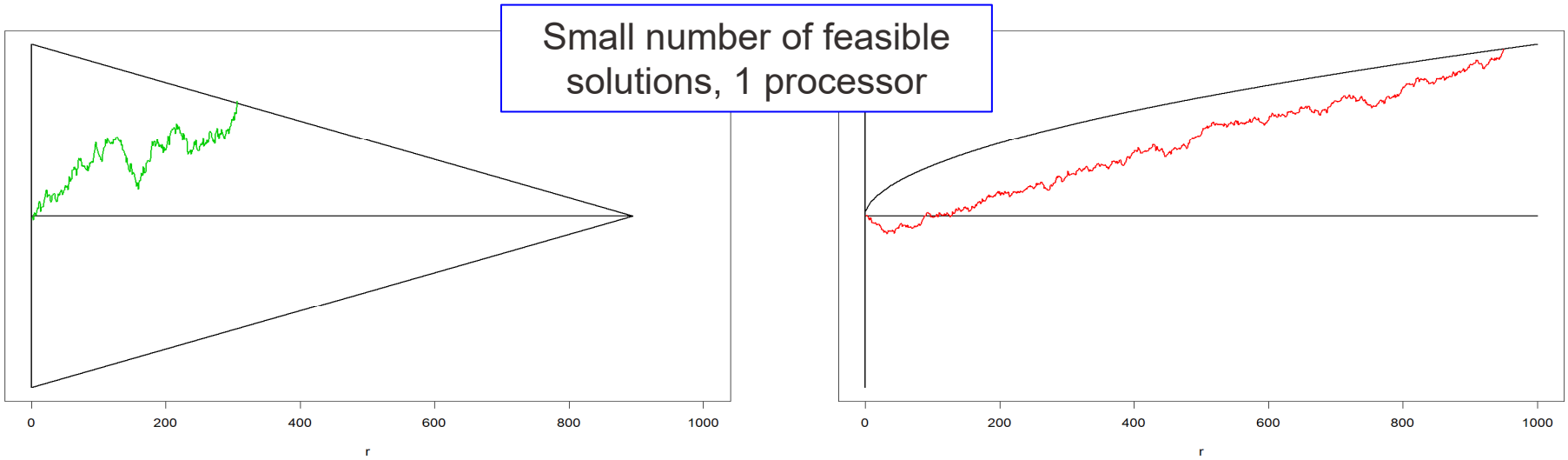
- Y_{ij} is jth i.i.d. simulation replication of feasible solution $i = 1, 2, \ldots, k$.
- $\mu_i = \mathrm{E}(Y_i)$ the expected (mean) value of solution i.
- Smaller is better, so the best is $i^\star = \mathrm{argmin}_i \, \mu_i$ with value $\mu^\star$.
- There are $p \geq 1$ parallel "processors" available.
- Our goal is to find the best with statistical confidence.
- So sit back and relax while I tell the tale of two statistics, one GOOD and one HORRIBLE...
- ...or is it the other way around?



Statistic A: $\sum_{j=1}^{r}(Y_{ij} - Y_{\ell j}),\ i \neq \ell$



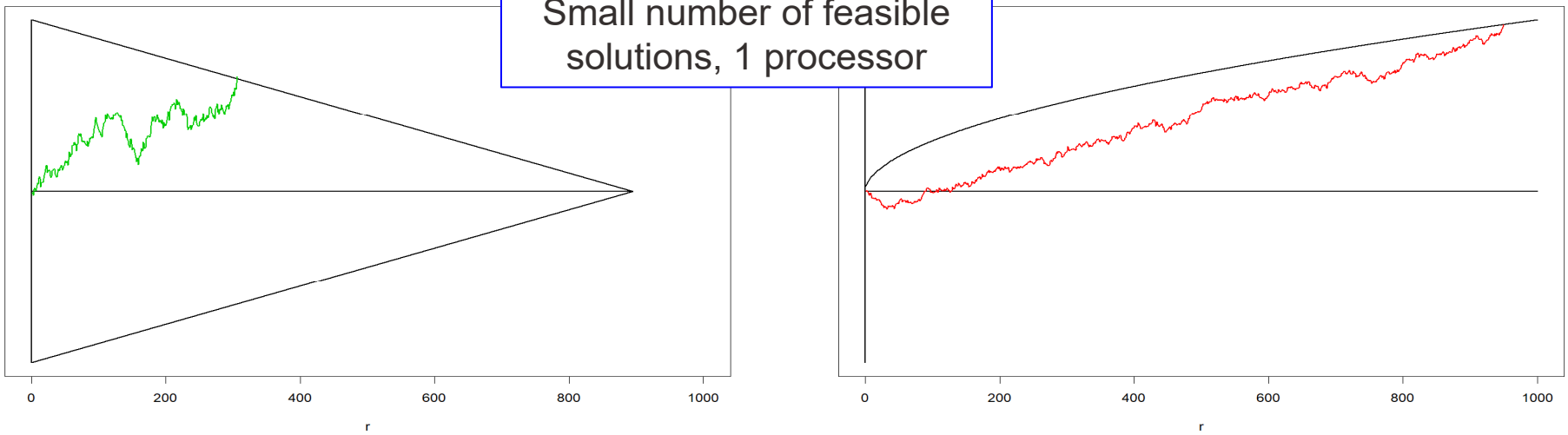
Statistic B: $\sum_{j=1}^{r}(Y_{ij} - \bar{\bar{Y}})$



Good: $\sum_{j=1}^{r}(Y_{ij} - Y_{\ell j}),\ i \neq \ell$

- Superior solutions quickly eliminate inferior ones.
- Probability of correct selection (PCS) guarantee.

Small number of feasible solutions, 1 processor



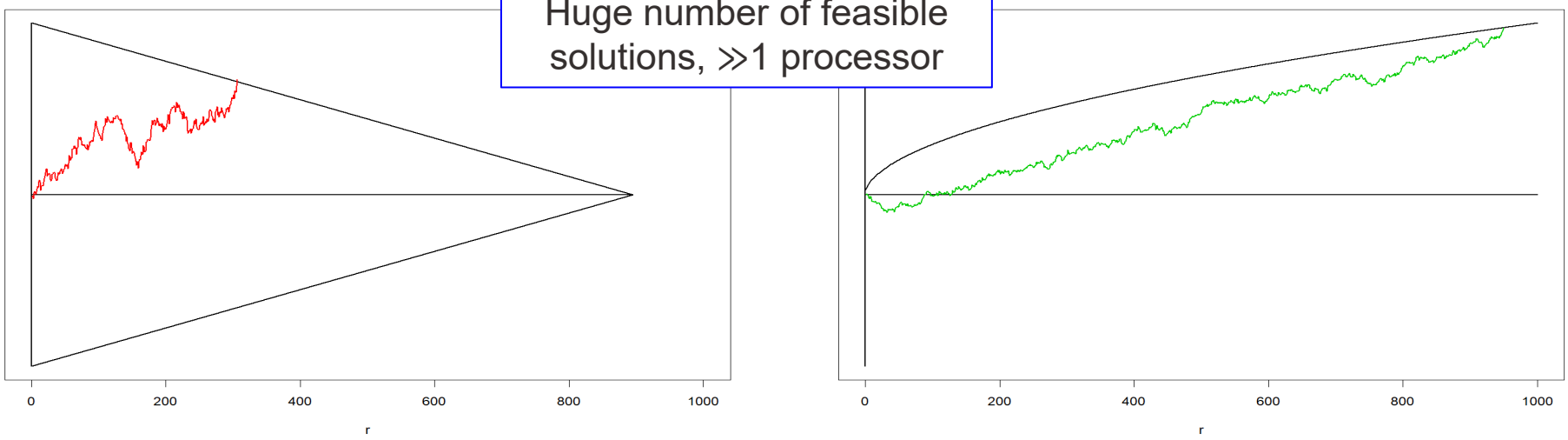
Good: $\sum_{j=1}^{r}(Y_{ij} - Y_{\ell j}),\ i \neq \ell$

- Superior solutions quickly eliminate inferior ones.
- Probability of correct selection (PCS) guarantee.

Horrible: $\sum_{j=1}^{r}(Y_{ij} - \bar{\bar{Y}})$

- In worst case requires 400% more simulation!
- Expected false eliminate rate (EFER) guarantee.

Huge number of feasible solutions, $\gg$1 processor



Horrible: $\sum_{j=1}^{r}(Y_{ij} - Y_{\ell j}),\ i \neq \ell$

- $\binom{k}{2}$ explosion (FDA: 700 billion!).
- Highly coupled operations.
- PCS becomes absurdly demanding.

Good: $\sum_{j=1}^{r}(Y_{ij} - \bar{\bar{Y}})$

- Fast updates.
- No need to wait for coupling.
- EFER guarantee is scale-free.

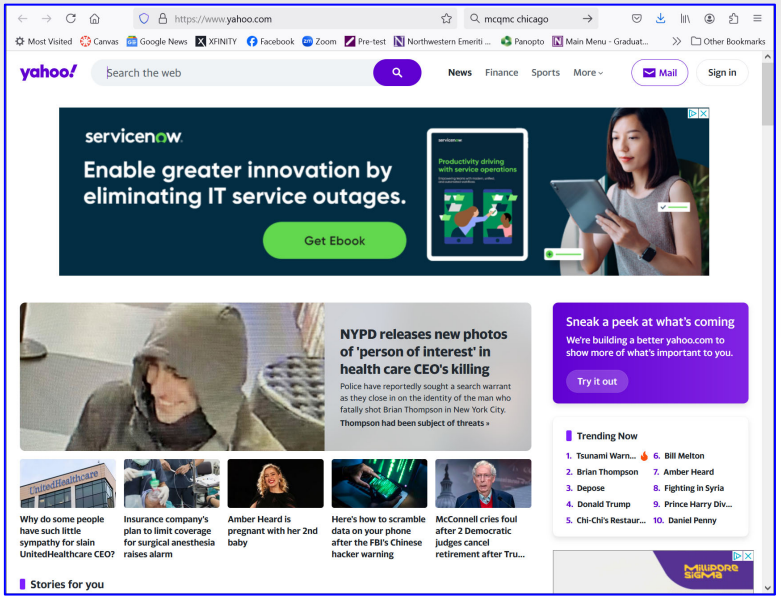
Extra for experts

- Parallel is great, but "efficiency" is now more than "number of observations." And statistical *validity* is even more critical (and difficult)!
- Yes, in extensive tests we beat competitors yada, yada, yada....
- The million system problem takes < 20 minutes using just $p = 21$ processors in an MPI4Py environment.
- Statistic A is the basis for the procedure KN in Simio.
- Statistic B is the basis for bi-PASS the newest tool in Simio (but we make no $ 😢).
- Pei, Nelson & Hunter (2024). Parallel adaptive survivor selection. *Operations Research* **72**, 336–354.



Illustration #2

- When you load the Yahoo! web page they want you to click on *something*.
- They know *a lot* about you ("covariates").
- Given your covariates they can simulate assortments to estimate the $\Pr\{\text{Click}\}$ for each.
- But how can they display the best assortment for you in milliseconds?



Data-driven decision making & simulation

- Making online decisions conditional on contemporaneous information.
  - Deferring the decision reduces uncertainty which refines the decision.
- BIG PUSH #1: Digital twins!
  - Examples: Alcoa Baie-Comeau facility in Quebec; airspace in the UK.
  - Time enough for at least some simulation.
- BIG PUSH #2: Decision is needed too quickly for real-time simulation.
  - Examples: Personalized medicine; assortment optimization.
  - "Offline simulation for online application."

Just a tiny bit more notation

- $\mathbf{X}$ is a vector of covariates to which we want to react.
- $\mu_i(\mathbf{x}) = \mathrm{E}(Y_i|\mathbf{X} = \mathbf{x})$, the conditional expected value of decision i.
- If you tell me the covariate value $\mathbf{x}$, I can simulate $Y_{ij}|\mathbf{X} = \mathbf{x},\ j = 1, 2, \ldots$.

Just a tiny bit more notation

- $\mathbf{X}$ is a vector of covariates to which we want to react.
- $\mu_i(\mathbf{x}) = \mathrm{E}(Y_i|\mathbf{X} = \mathbf{x})$, the conditional expected value of decision i.
- If you tell me the covariate value $\mathbf{x}$, I can simulate $Y_{ij}|\mathbf{X} = \mathbf{x},\ j = 1, 2, \ldots$.
  - So *if* I had time, which I don't, I could use simulation optimization to select the best decision among $i = 1, 2, \ldots, k$ with high confidence.
  - Still, it would be nice to leverage the fact that I am really good at solving these "simulate all" problems.
  - Need to use machine learning in a statistically sound way that exploits the fact that we create the data!

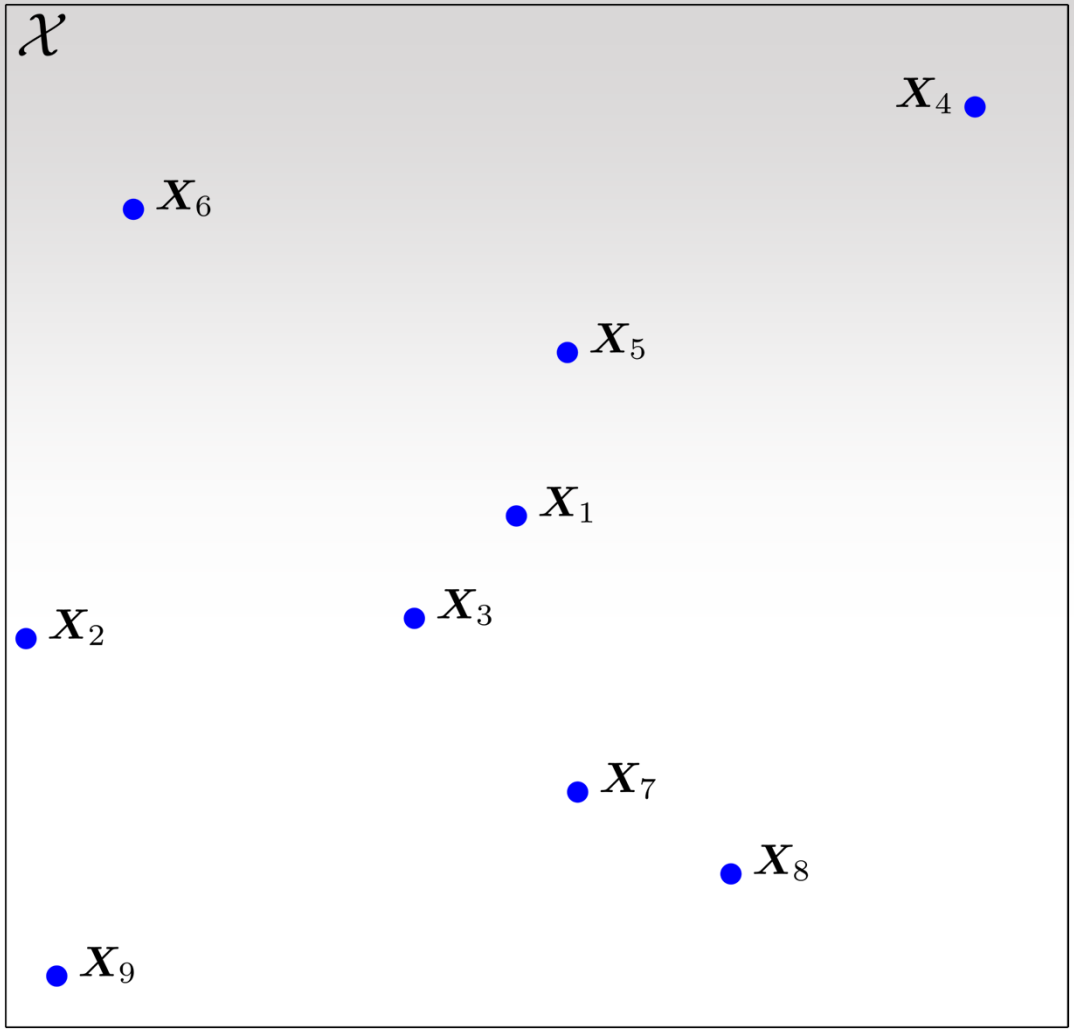
The idea in pictures:

- $\mathcal{X}$ is the space of covariate values
- Covariate is a random vector
- Distribution previously estimated or have data



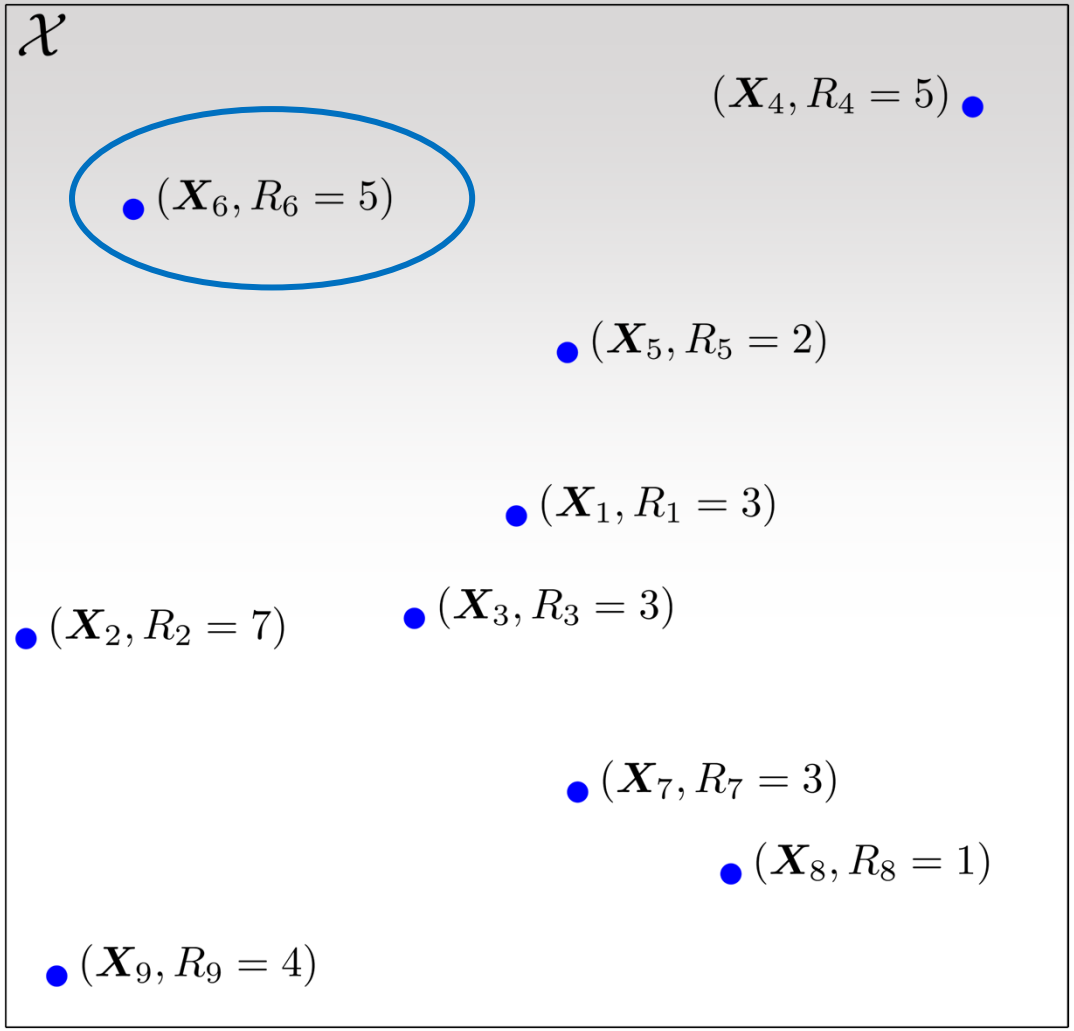
The offline experiment:

- Simulate m i.i.d. (and fake) covariate values $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_m$ ($m = 9$ here)
- Why simulate rather than design? Good question!



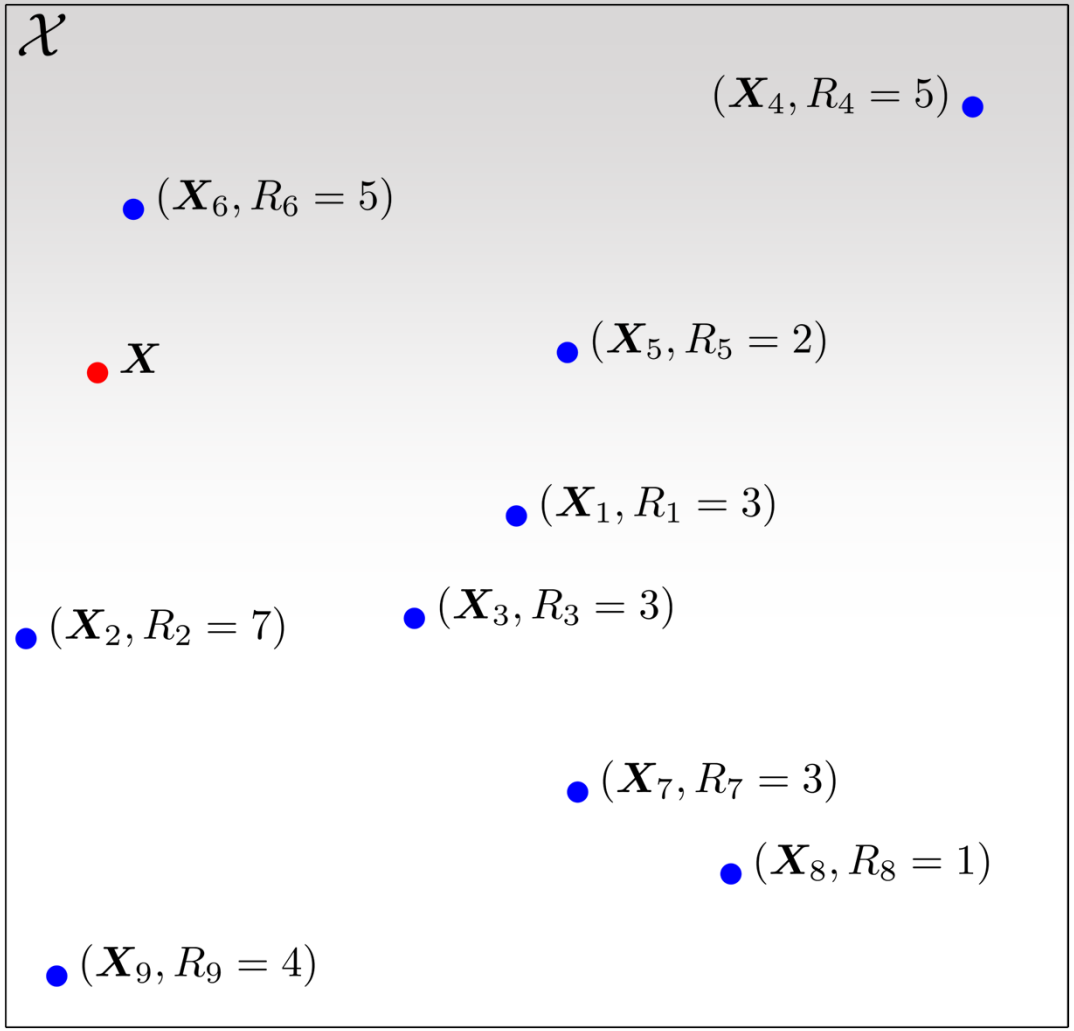
Offline optimization:

- Run simulation optimization at each $\boldsymbol{X}_i$
- Selected decision is R_i ($k = 7$ decisions here)
- Record the point estimates $\widehat{\mu}_j(\boldsymbol{X}_i), j = 1, 2, \ldots, k$ $i = 1, 2, \ldots, m$
- This is our database ($\mathcal{D}_m, \mathcal{R}_m, \widehat{\mu}$'s)



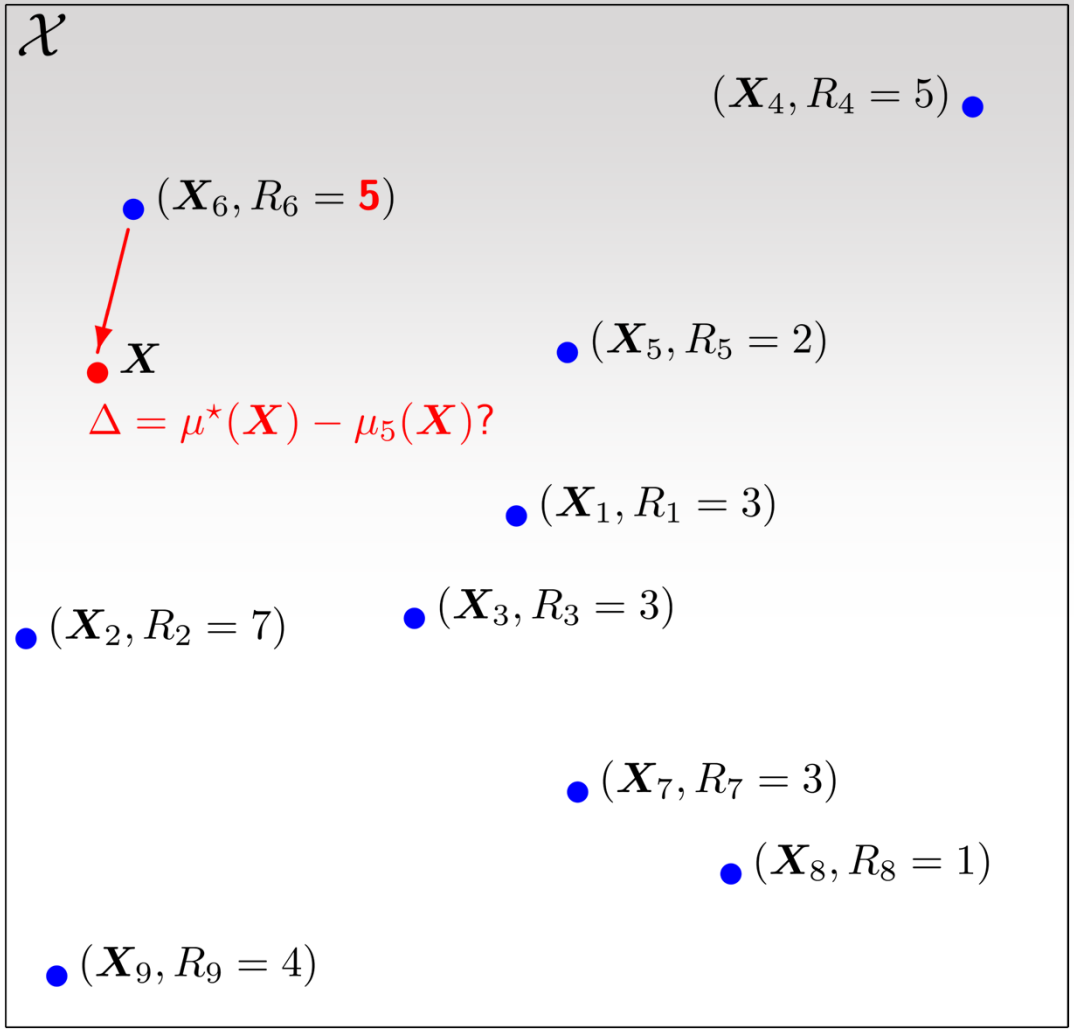
How will we use the database?

- Covariate $\bullet$ $\boldsymbol{X}$ happens.
- Use database to make a decision $1, 2, \ldots, k$.



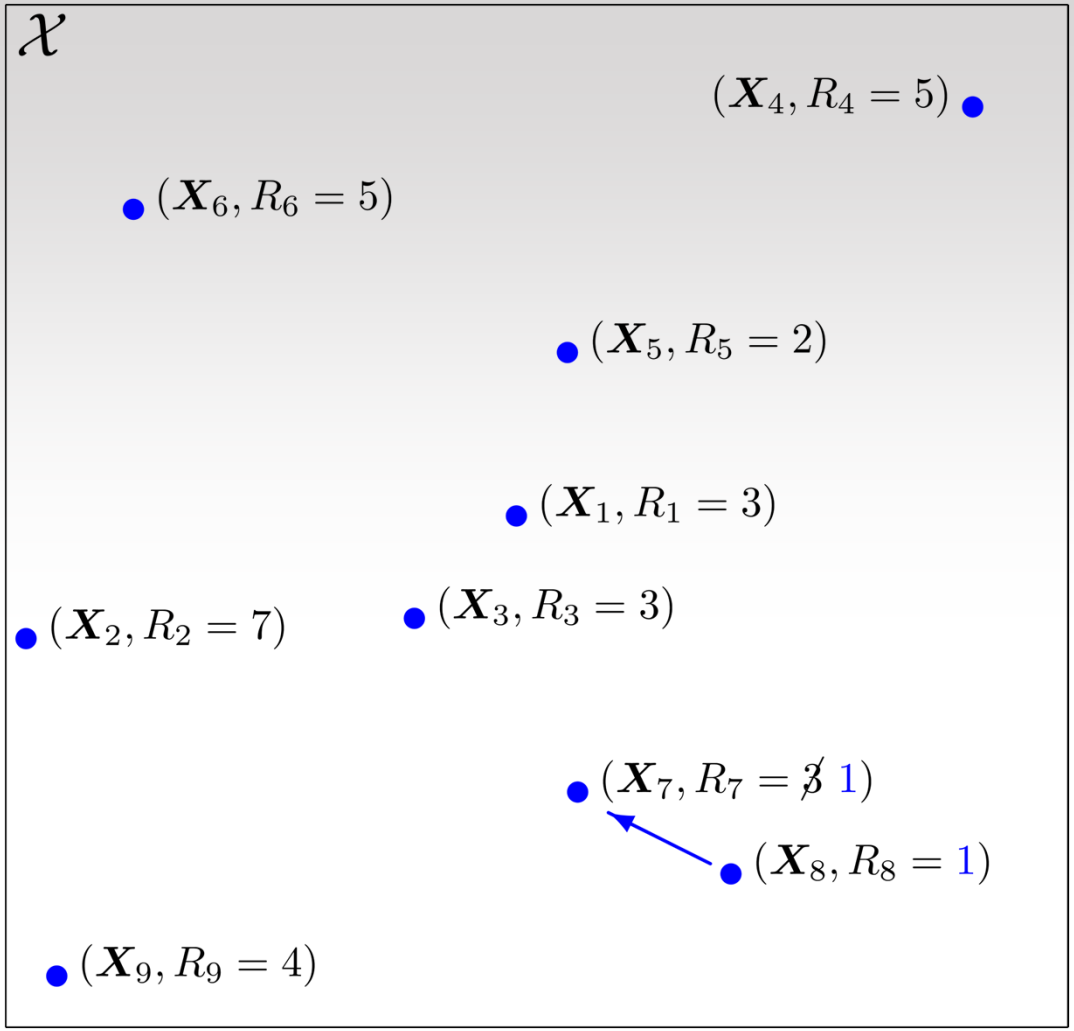
How will we use?

- Covariate • $\boldsymbol{X}$ happens.
- Use database to make a decision $1, 2, \ldots, k$.
- Choose same decision as nearest neighbor.
- What can I say about the optimality gap Δ?



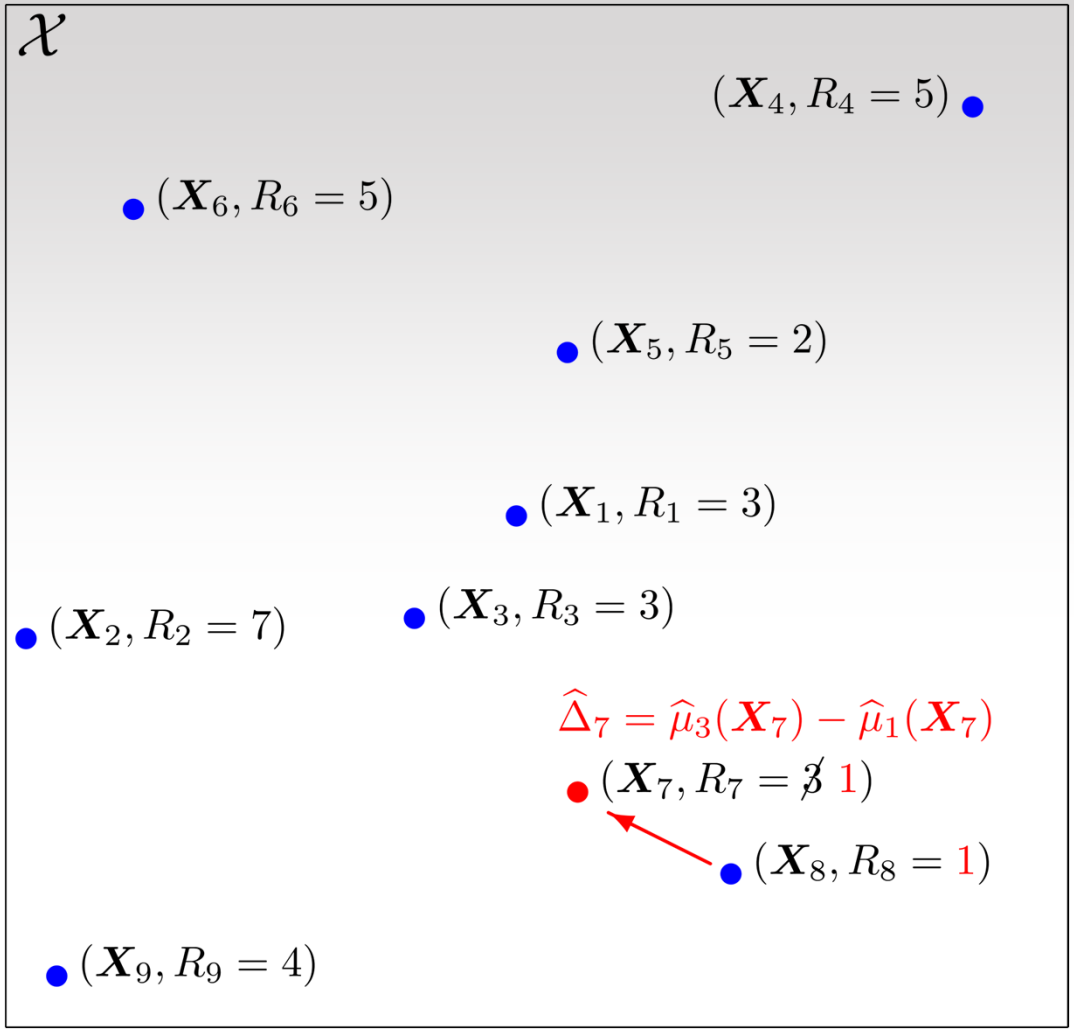
Offline assessment:

- Estimate the optimality gap at each design point
- Use leave-one-out logic and point estimates



Offline assessment:

- Estimate the optimality gap at each design point
- Use leave-one-out logic and point estimates
- Obtain optimality gaps $\widehat{\Delta}_1, \widehat{\Delta}_2, \ldots, \widehat{\Delta}_m$
- Provides $1-\alpha$ quantile of the optimality gap distribution, $\widehat{\Delta}_m^{1-\alpha}$

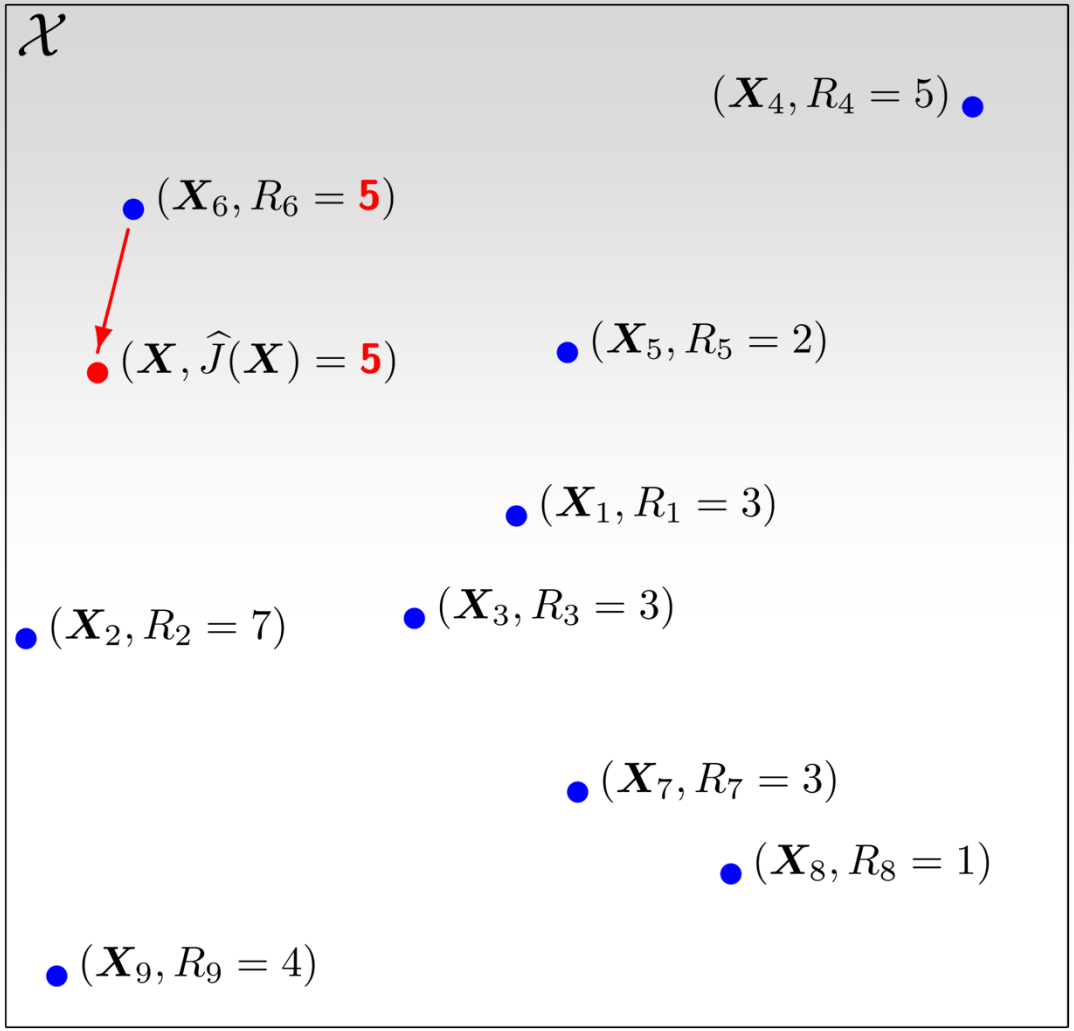


Online classification:

- Given new covariate • $\boldsymbol{X}$ classify as NN in database
- Claim optimality gap bound

$$\Pr\left\{\Delta \leq \widehat{\Delta}_m^{1-\alpha}\right\} \geq 1-\alpha$$

- Add design points if $\widehat{\Delta}_m^{1-\alpha}$ too large.



Why this idea?

- Exploits what we know how to do well: little simulation optimizations.
- Leverages that we classify by value so classification rate is **not** the objective.
- Building the database can be completely parallelized; only the leave-one-out assessment requires a coupled operation.
- Performance guarantee does not depend on any assumptions about $\mu_i(\mathbf{x})$.
- 1NN requires no tuning and is easy to implement.
- Post-decision, the new • can be optimized and added to the database.

Extra for experts

- If this smells like conformal inference then your nose is working!
  - Exchangability doesn't hold so some clever conditioning is required.
  - Have to show that plug-in optimality gap estimators are ok too.
- Other classifiers can be used!
- We do better than competitors even when they exploit heroic assumptions (e.g., response is linear in the covariates), blah, blah, blah.
- Works great on the Yahoo! data.
- Keslin, Nelson, Pagnoncelli, Plumlee & ⋆Rahimian⋆ (2025). Ranking & contextual selection. *Operations Research* **73**, 2695–2707.



Some closing thoughts

- Digital twins are creating new statistical methodology problems that are yet to be solved ($\Longrightarrow$ good research topic for IE, CS & Stat!).
- "Anyone doing simulation experiments who is not thinking about how to parallelize them is missing the point!"

  —BL Nelson from *Statements That Are Likely To Get Me In Trouble*

- If you haven't had enough, see
  - Nelson (2024). Rebooting simulation. *IISE Transactions* **56**, 385–397.
  - Hong & Nelson (2026). Fifty years of stochastic simulation: Where we are and where we need to go. *European Journal of Operational Research* **330**, 701–714.

Questions I think you will ask me

- Why not search? R&S seems crude.
- Continuous decision variables?
- Uncertainty quantification?
- What are those Digital Twin topics?
- Regression instead of classification?
- Oh yeah, that question.
- Bandits?
- Simulations for “policy?”
- What about AI & simulation?



S
C
U
D
R
Q
B
P
A