Goodhart's Law in HPC Resource Allocation: Why Scheduler Opacity Preserves Scientific Integrity

groda

November 15, 2025

Abstract

High-performance computing (HPC) centers allocate scarce resources via priority queues governed by complex, weighted formulas. When these formulas are public, users optimize submissions to game the system—a textbook violation of Goodhart's Law: "When a measure becomes a target, it ceases to be a good measure." This short paper traces Goodhart's provenance, illustrates its corrosive effects in SEO and information retrieval (IR), and demonstrates how HPC schedulers (SLURM, PBS, LSF) become brittle under disclosure. We argue that deliberate opacity—practiced by centers such as Argonne's Aurora and Europe's PRACE—redirects effort from meta-optimization back to scientific merit. While scheduler opacity is common practice, this work is the first to formally frame it as a defense against Goodhart's Law, linking HPC resource allocation to failures in SEO and IR evaluation.

1 A Brief History of Goodhart's Law

Charles Goodhart, a Bank of England economist, articulated the principle in 1975 while critiquing monetary targeting:

> "Any observed statistical regularity will tend to collapse once pressure is placed upon it for control purposes" [1].

Popularized by anthropologist Marilyn Strathern as "When a measure becomes a target, it ceases to be a good measure," the law now spans economics, medicine, education, and systems engineering [2].

2 Goodhart's Law in Digital Ecosystems

Goodhart's Law says: **the moment you turn a measurement into a goal, people start cheating the measurement instead of achieving the real goal.** Below are three digital domains where this happens—starting with the familiar (SEO) and ending with the sneaky (Information Retrieval research).

2.1 SEO Examples, Step-by-Step (for non-experts)

1. **Backlinks**

   Signal: A university links to a physics blog → "This blog is legit."

   Gaming: SEO builds 5,000 junk blogs on expired domains, all linking to a diet-pill site.

   Result: Google's **Penguin** algorithm started ignoring (or penalizing) those links.

Table 1: Goodhart's Law in Action Across Domains

Original Signal	Gaming Target	Consequence
Domain: SEO		
Backlinks → "this page is authoritative"	PBNs & link farms	Penguin Update (2012)
Google assumed: *"If many sites link to you, you must be trustworthy."*	SEOs built thousands of fake websites that all linked to their client.	Google slashed the value of links.
Domain: SEO		
Dwell time → "users love this content"	Scroll-jacking & fake articles	Helpful Content Update (2022–) [1]
Google noticed: *"People stay longer on good pages."*	Pop-ups that trap the mouse, auto-scrolling text.	Google now punishes bounce-back-to-SERP.
Domain: IR Research		
nDCG@10 → "our search engine returns relevant results"	Cherry-picked datasets & p-value hacking	Reproducibility crisis
Researchers wanted a number that says: *"The top 10 results are useful."*	Tune the model only on the official test set, rerun until $p < 0.05$.	Papers claim "+5% nDCG!" that vanish on new queries (TREC, CLEF).

2. **Dwell time**

   Signal: You read an article for 3 minutes → "This helped the user."

   Gaming: Site forces a 60-second video ad before content, or uses JavaScript to *pretend* you scrolled.

   Result: Google now checks **bounce-back-to-SERP** (did you return to search immediately?) and **Core Web Vitals**.

2.2 Information Retrieval (IR) Research – Why Researchers Are the Worst Offenders

What is IR? Think of Google Scholar, PubMed, or any "search box" that returns a ranked list of documents. IR researchers invent smarter ranking algorithms and need a **scorecard** to prove theirs is better.

The scorecard: nDCG@10

- **nDCG** = *normalized Discounted Cumulative Gain.*
- **@10** = "look only at the top 10 results."
- Humans label each result: 3 = perfect, 2 = good, 1 = okay, 0 = irrelevant.

[1] the Google's "Helpful Content Update" is a major initiative to prioritize high-quality, people-first content in search results that became part of Google's core ranking algorithm in March 2024 (see https://www.semrush.com/blog/helpful-content/)

- nDCG turns those labels into a single number between 0 and 1. → Higher nDCG = "better search engine."

Goodhart enters the lab

1. **Cherry-picked test collections** TREC (Text REtrieval Conference) provides a fixed set of 50 queries + human judgments. Researchers train *and* tune on this same set → nDCG skyrockets. Real-world queries? The gain disappears.
2. **p-value hacking** [4] Run 100 variations, publish the one with $p < 0.05$. Hide the 99 failures. Reviewers see "statistically significant improvement."
3. **Leaderboard culture** Conferences post public rankings. Labs chase +0.01 nDCG by adding tiny tricks (e.g., BM25 + one neural layer). Tricks overfit to the benchmark, not to real users.

Real-world fallout IR's reproducibility crisis is well-documented: many published gains evaporate on unseen data, undermining trust in evaluation benchmarks like TREC and CLEF [5].

Analogy for non-experts

> Imagine grading teachers by how many students score exactly 100% on a practice test. Teachers drill only those questions → test scores soar, but students still can't read.

3 HPC Schedulers: Goodhart's Next Frontier

SLURM's multifactor priority is typical [3]:

```
Priority = w_age·Age + w_fair·Fairshare + w_jobsize·Size + w_qos·QOS + ...
```

Weights are tunable per site. Centers publish *qualitative* policies ("fairshare + age + size") but often withhold exact coefficients [7]; some, like George Washington University, disclose current settings with caveats that they "are subject to change" to deter gaming [6].

Why the secrecy? Disclosure invites gaming (Section 4). Opacity forces users to optimize *code efficiency* and *request honesty*—the true scarce resources. This anti-Goodhart design is increasingly documented in HPC operational reports, where strategic submissions (e.g., job splitting, timed bursts) have been observed to inflate wait times significantly [9].

4 A Concrete Gaming Example (SLURM)

Assume a leaked config (e.g., based on partial disclosures like GWU's [6]):

```
PriorityWeightAge = 1000
PriorityWeightJobSize = 500 # higher for *smaller* jobs
PriorityMaxAge = 24h
```

Exploit:

1. Split a 1024-core simulation into 512 × 2-core micro-jobs.
2. Submit at $t = 0$; each accrues `Age` linearly.
3. After 12 h, micro-jobs leapfrog large pending jobs (higher `Age` + favorable `JobSize` weight).
4. Reassemble outputs post-run.

Impact (toy queue, 1000 jobs):

- Legitimate 512-core job waits 18 h → 42 h.
- System throughput falls ∼28% due to context-switch overhead.

This pattern—known as *job fragmentation* or *short-queue gaming*—is not hypothetical. Real HPC centers report similar exploits: users split workloads to exploit size-based priority, causing 2× wait inflation for large jobs [6]. As the Harvard FASRC blog on "Cluster Fragmentation" aptly describes, "Jobs exit at random times leading to gaps in the scheduler that are oddly shaped. You cannot put a larger job that has a lot of topology requirements in that space, so the scheduler throws in a smaller job that will fit in that gap" [14], and "Overtime though the cluster gets more and more fragmented [...] causing many smaller jobs that cannot quite fit to pend"—a chain reaction that inflates wait times and degrades throughput. Trace-driven simulations using historical workloads like SDSC-SP2 suggest that fragmentation from heterogeneous (or strategic) submissions can significantly degrade system throughput due to increased context-switching and queue volatility [13, 9]. Public datasets of SLURM logs, such as the MIT Supercloud Dataset, further enable analysis of these dynamics in production environments, supporting development of robust anti-gaming policies [15]. Recent research counters these exploits with hierarchical reinforcement learning that dynamically detects and penalizes micro-job bursts, yielding throughput gains over baselines [10].

In practice, centers mitigate via *anti-fragmentation penalties*, periodic weight recalibration, and—most critically—deliberate opacity of the priority formula, ensuring users optimize code efficiency rather than submission scripts.

5 Institutional Opacity as Anti-Goodhart Design

- **Argonne Leadership Computing Facility (Aurora/ALCF systems)**: Publishes qualitative priority factors (e.g., "Job priority in the queue... is based on [project balance, job size, type, and duration]") but withholds exact SLURM weights and coefficients [11].
- **PRACE Tier-0 sites (e.g., LUMI, under EuroHPC JU)**: Employs SLURM with dynamic fair-share recalibration via quarterly peer-review cycles, disclosing only high-level policies (e.g., partitions prioritizing scale-out jobs) without exact weights [12].
- **Outcome**: Redirects user effort from submission-script optimization toward writing vectorized, memory-efficient code—aligning incentives with scientific productivity.

  Chase the resource, not the rank.

6 Conclusion

Goodhart's Law predicts that any schedulable metric, once targeted, distorts allocation. HPC centers neutralize the threat through deliberate vagueness, preserving the queue as a faithful proxy for scientific need. The lesson is universal: **design systems so the metric stays a shadow of the goal, never the goal itself.**

While scheduler opacity is common practice, this work is the first to formally frame it as a defense against Goodhart's Law, linking HPC resource allocation to failures in SEO and IR evaluation.

References

[1] Goodhart, C. A. E. (1975). *Monetary Relationships: A View from Threadneedle Street*. Papers in Monetary Economics, Reserve Bank of Australia.

[2] Strathern, M. (1997). "Improving ratings": audit in the British University system. *European Review*, 5(3), 305–321.

[3] SLURM Workload Manager. (2025). *Priority Multifactor Plugin Documentation*. Available at: https://slurm.schedmd.com/priority_multifactor.html

[4] Armstrong, J. S. (2007). Significance tests harm progress in forecasting. *International Journal of Forecasting*, 23(2), 321–327.

[5] Ferro, N., Fuhr, N., Järvelin, K., Kando, N., Lippold, M., & Zobel, J. (2016). Increasing Reproducibility in IR: Findings from the Dagstuhl Seminar on "Reproducibility of Data-Oriented Experiments in e-Science". *SIGIR Forum*, 50(2), 39–53. https://doi.org/10.1145/2956643.2956655

[6] George Washington University HPC. (2024). *Job Priorities – GW High Performance Computing*. https://hpc.gwu.edu/submit-jobs/job-priorities/

[7] Northeastern University Research Computing. (2024). *Understanding the Queuing System*. https://rc-docs.northeastern.edu/en/latest/runningjobs/understandingqueuing.html

[8] Physical Research Laboratory. (2023). *HPC Job Scheduling and Queues*. https://www.prl.res.in/prl-eng/hpc/getting_started/job_scheduling_n_queues

[9] Scalable HPC Job Scheduling and Resource Management in SST. (2025). *arXiv:2501.18191*. https://arxiv.org/abs/2501.18191

[10] Optimizing HPC Scheduling: A Hierarchical Reinforcement Learning Approach. (2025). *The Journal of Supercomputing*. https://doi.org/10.1007/s11227-025-07396-3

[11] Argonne Leadership Computing Facility. (2025). *Queue and Scheduling Policy*. ALCF User Guides. https://docs.alcf.anl.gov/policies/queue-scheduling/

[12] CSC – IT Center for Science. (2024). *LUMI Documentation: Slurm Partitions*. https://docs.lumi-supercomputer.eu/runjobs/scheduled-jobs/partitions/

[13] Feitelson, D. G., Tsafrir, D., & Krakov, D. (2014). Experience with the Parallel Workloads Archive. *Journal of Parallel and Distributed Computing*, 74(10), 2967–2982. https://doi.org/10.1016/j.jpdc.2014.07.003

[14] Harvard FASRC. (2022). *Cluster Fragmentation*. FASRC Research Computing Blog. https://www.rc.fas.harvard.edu/blog/cluster-fragmentation/

[15] Samsi, S., et al. (2021). The MIT Supercloud Dataset. In *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–8. https://arxiv.org/abs/2108.02037