Bayesian Query Super-Optimization

Jeffrey Tao, Natalie Maus, Haydn Jones, Jacob Gardner, Ryan Marcus | DB@Penn

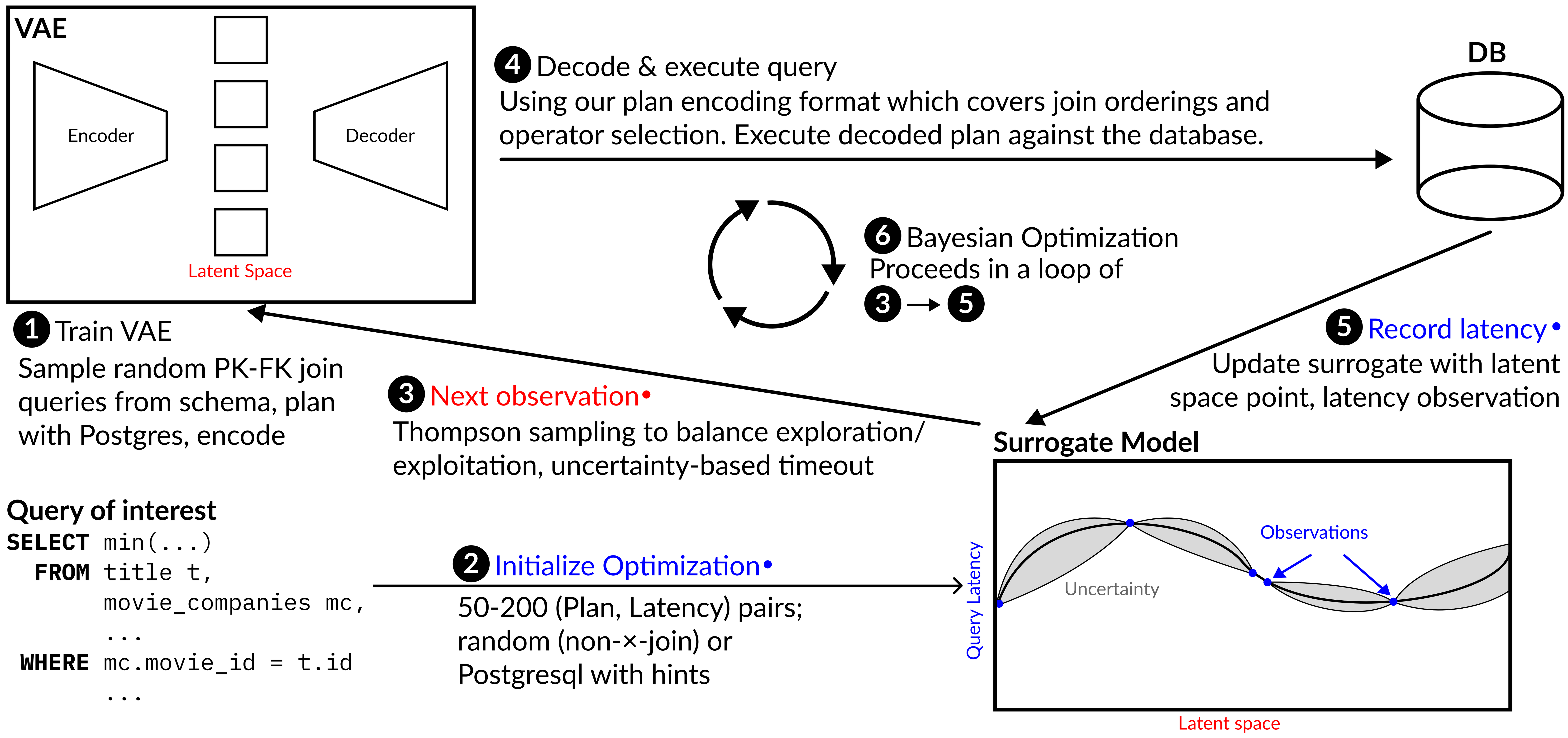


Figure 1. Our system super-optimizes queries by searching the space of possible query plans using Bayesian Optimization.

Motivation

You have a set of queries that are well-known, run frequently, and yet are *under-optimized*!

What if you optimized your queries offline?

1. The space of possible plans is *vast*.
2. Executing non-optimal queries is *expensive*!

Bayesian Optimization over structured inputs[1] with censored observations is *sample-efficient* and minimizes the impact of bad plans with *timeouts*.

Results

Evaluated over the JOB vs. PostgreSQL w/ optimal hints (Bao [2]) and reinforcement learning (Balsa [3]).

- After a few hours optimizing each query (parallelizable across queries), beats optimal hints on all queries and reduces total JOB execution time by ~1/3.
- Most optimization runs bottom out in low 100s of query executions.
- Finds strictly better plans than other methods
- Random query plan search (with timeouts) is unreasonably effective!

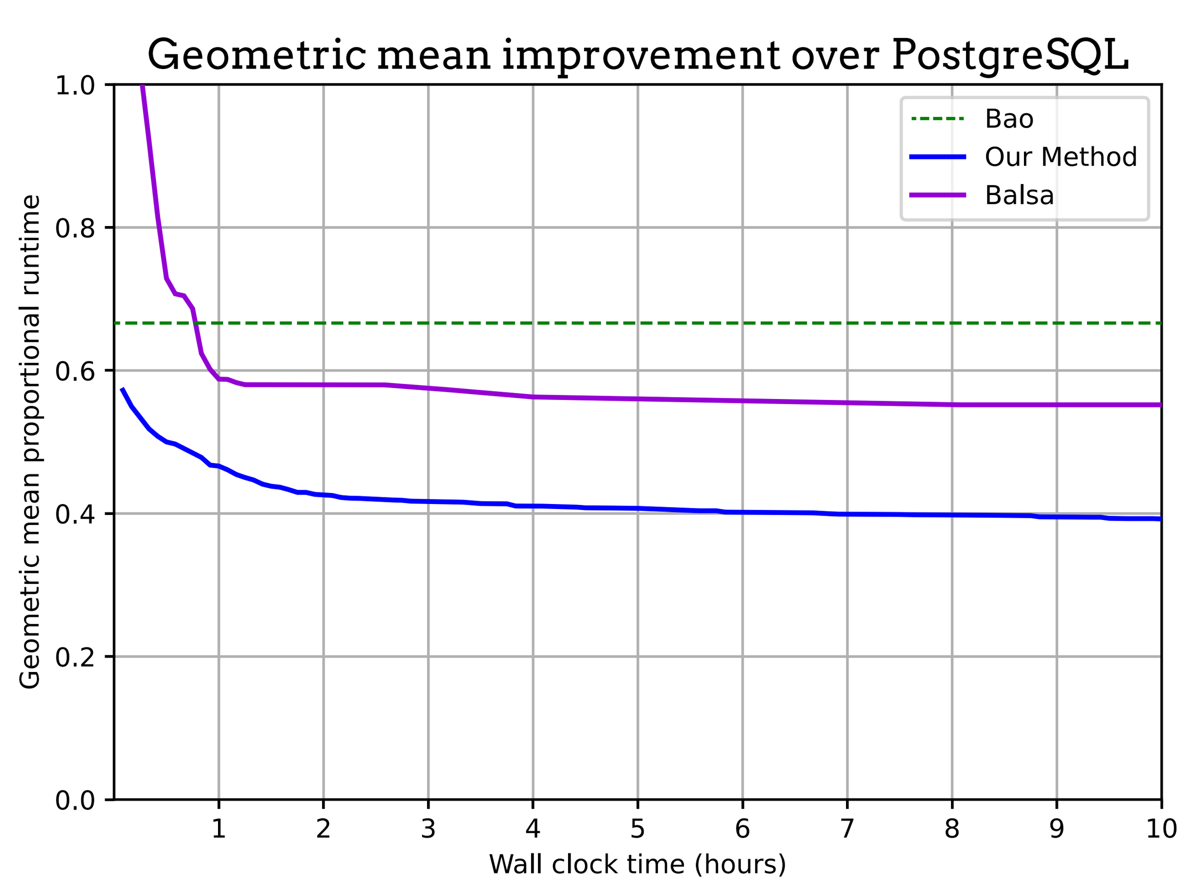


Figure 2. 1.0 is parity with PostgreSQL, lower is better. Our method produces the most per-query improvement.

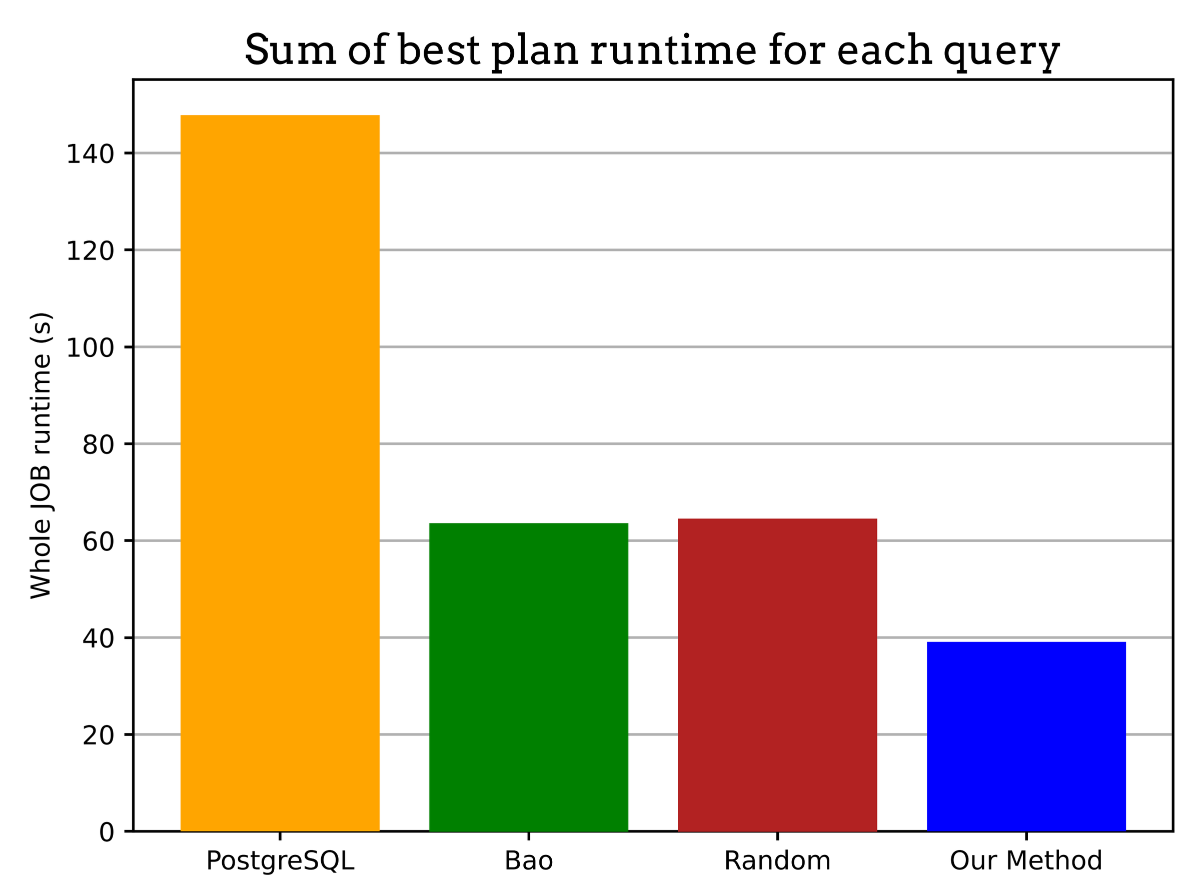
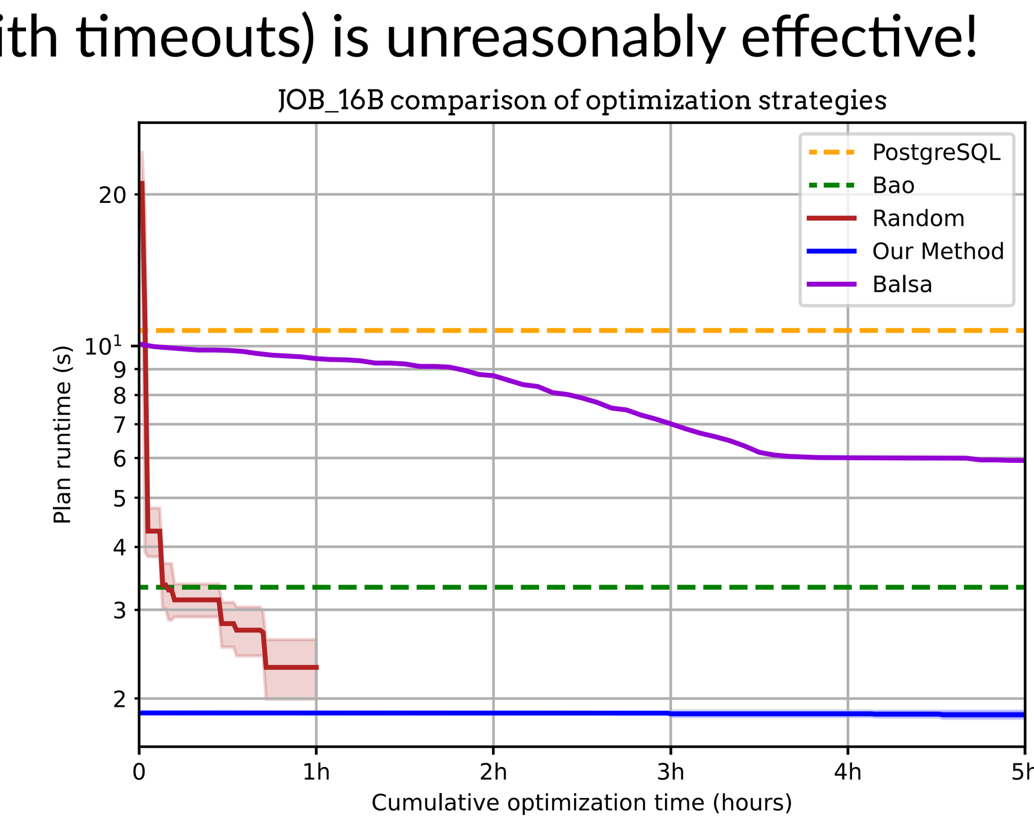
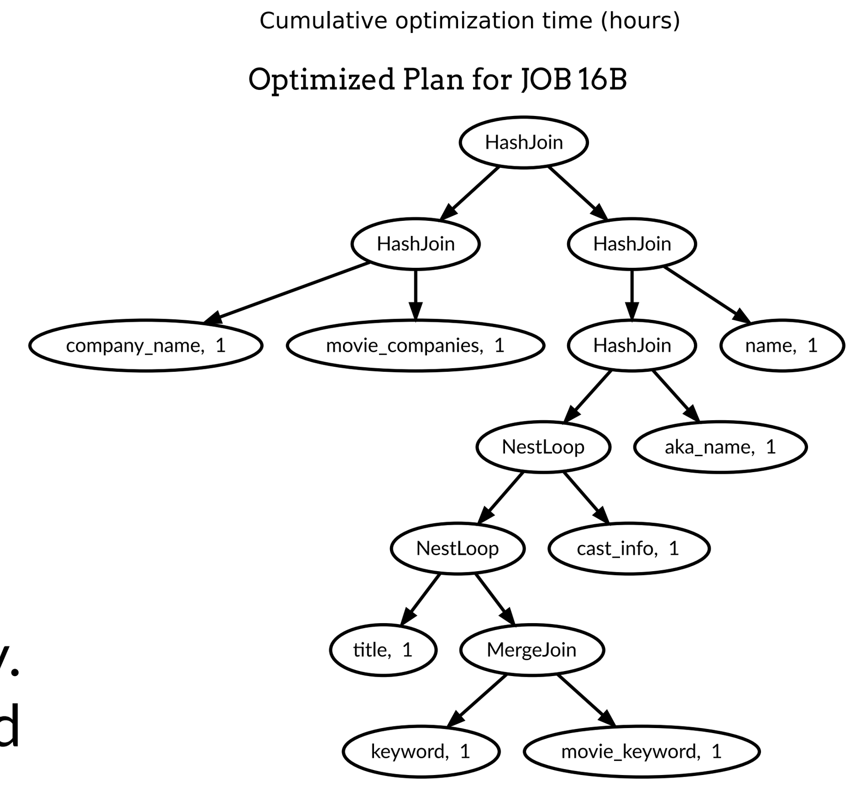


Figure 3. Lower is better. Our method reduces the total workload execution time by the most.





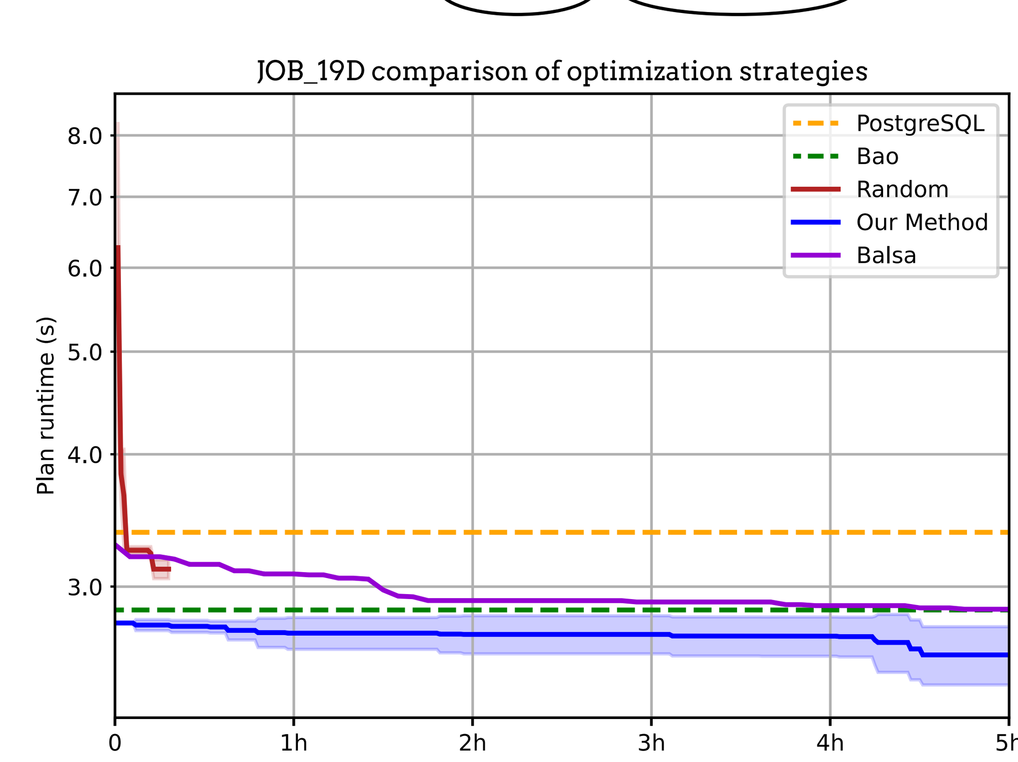


Figure 4 (right). Whether our method significantly improves upon the baselines varies per-query. Our method finds plans equivalent to or better than the best baseline on all queries. Optimized plans vary in shape (across JOB, 47 left deep, 51 zigzag, 14 bushy).

Future Work

- Jump-start optimization using the optimal Postgres hint set as a timeout
- Multi-task optimization of a whole workload by targeting the most-promising queries
- Train generative model for few-shot optimization of arbitrary queries

speculative.tech/nedb2024

[1] Maus et al., Local Latent Space Bayesian Optimization over Structured Inputs, NeurIPS '22
[2] Marcus et al., Bao: Making Learned Query Optimization Practical, SIGMOD '21
[3] Yang et al., Balsa: Learning a Query Optimizer Without Expert Demonstrations, SIGMOD '22