Fast, Precise Thompson Sampling for Bayesian Optimization

David Sweet

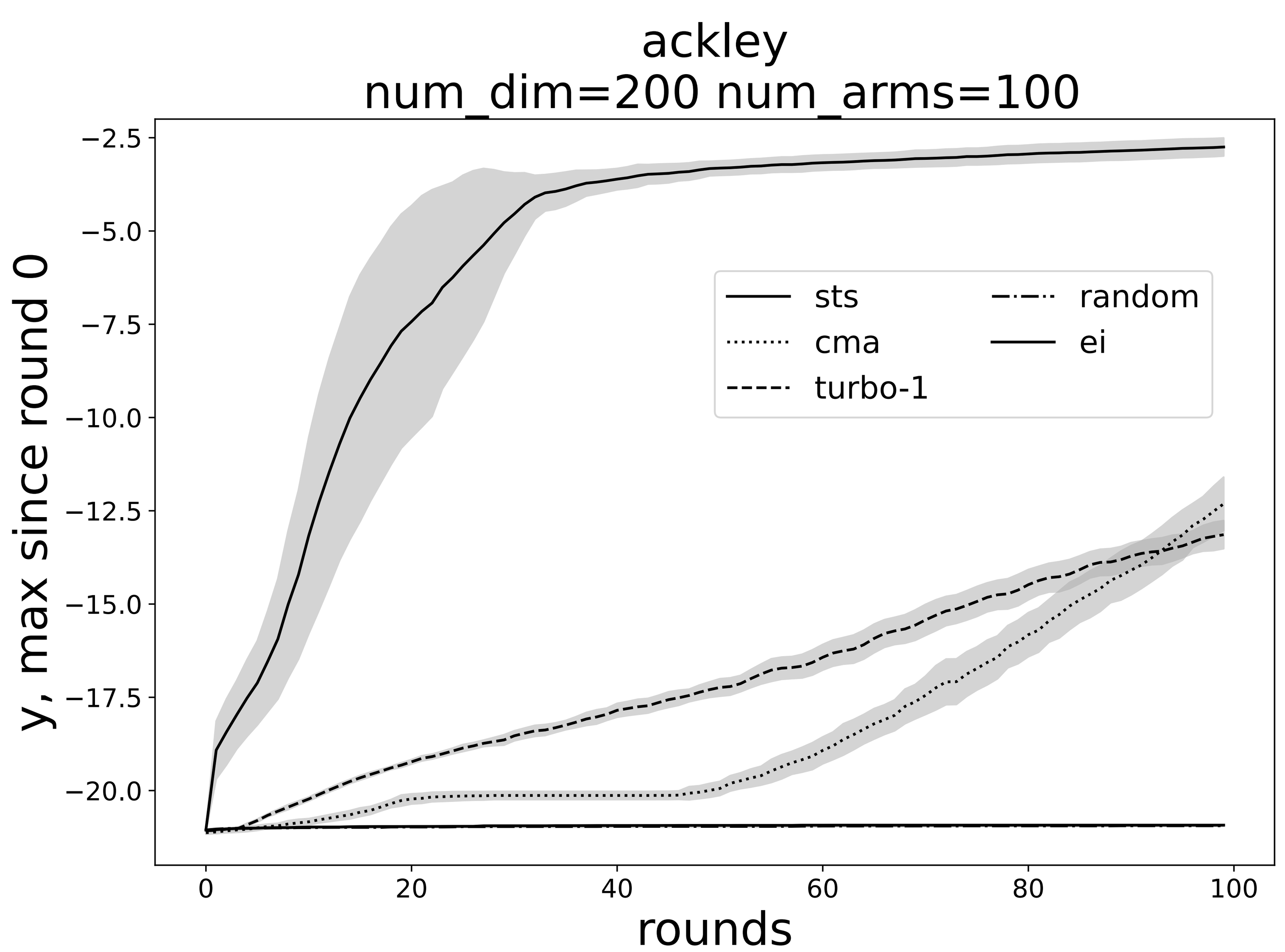
Ack: Siddhant Jadhav

YUBO: Yeshiva University Bayesian Optimization

Main Points

- Thompson sampling (TS) underperforms EI, UCB, etc.
  - STS, our method, uses hit-and-run to sample more efficiently than TS
  - STS outperforms TS, EI, UCB, etc. on **single-arm optimizations** in dimensions 1-300
- Batching (proposing multiple arms at once) is a general BO challenge
  - Minimal Terminal Variance (MTV) is a TS batching method
  - MTV+STS performs well at **batch optimization** in dimensions 1-300
- Our **scoring methodology** is resilient, accurate, and summarizable

Scoring acquisition methods



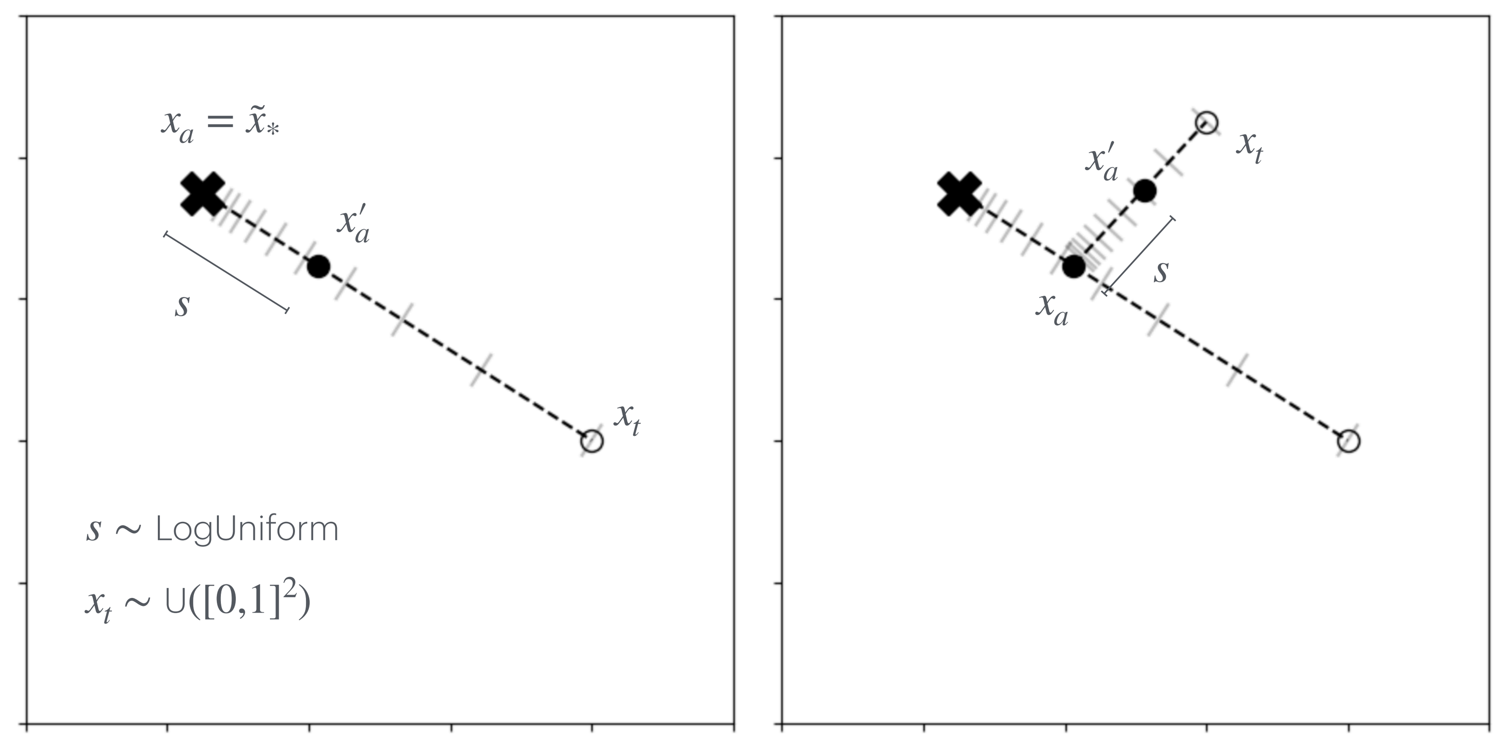
Optimization of Ackley function in 200 dimensions, 100 arms/round. *cma* CMA-ES, evolution strategy; *turbo-1* TuRBO, trust-region BO; *ei* Expected Improvement; *random* chooses arms randomly

STS finds high values very quickly for this function. To compare many methods over many functions we develop a scoring methodology that is

- **Resilient**: Rank-transforms performance for resilience to outliers
- **Accurate**: Randomly distorts test functions to reduce "center bias", e.g., where a test function may, perchance, have its optimum where a method initializes (e.g., the center of the bounding box)
- **Summarizable**: Range-normalizes ranks to make results from different test functions comparable

Stagger Thompson sampling (STS)

A Hit-and-Run Thompson sampler

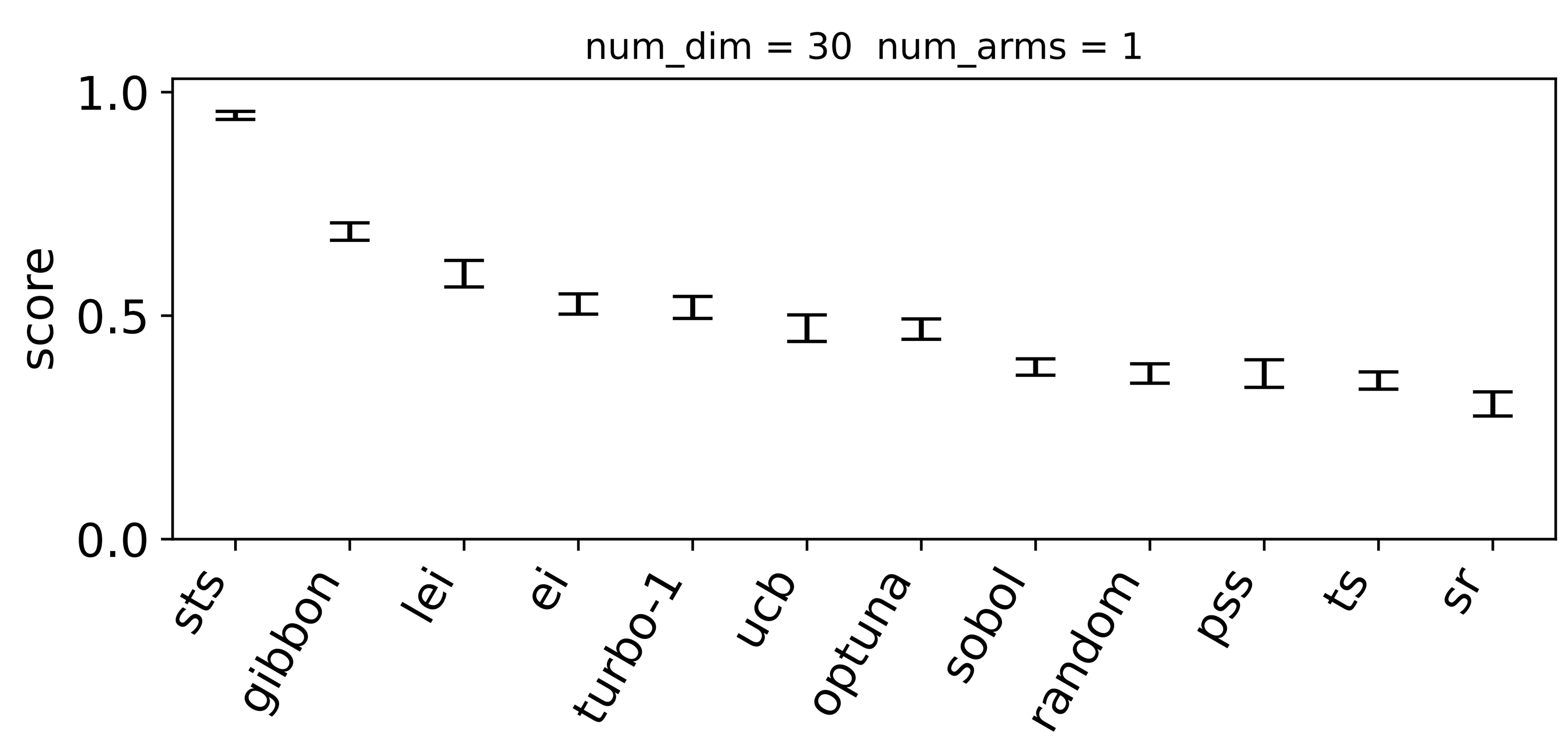


A Thompson sample is a draw from $p_*(x)$, the probability that x is the maximizer. STS iterates:

1. Sample a target uniformly in the bounding box: $x_t \sim U([0,1]^2)$
2. Propose $x'_a = x_a + (\sim LogUniform)(x_t - x_a)$
3. Thompson sample: $[y, y'] \sim \mathcal{GP}([x_a, x'_a])$; if $y' > y$, $x_a \leftarrow x'_a$

Step 3 is a Metropolis filter (see paper).

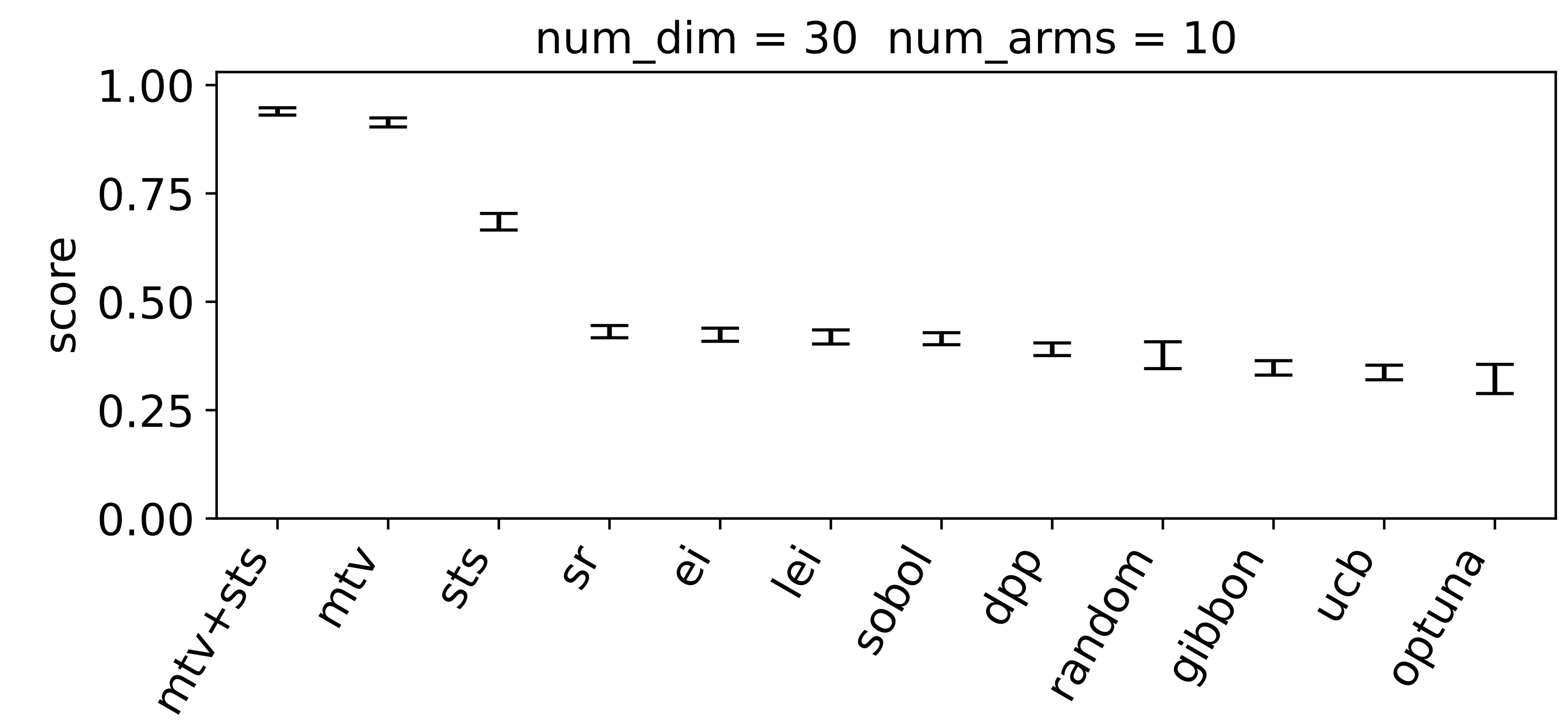
Comparisons: One arm/round



Optimization of nine test functions in 30 dimensions for 30 rounds, with 30 replications (each with a different random distortion of the test function) with 1 arm/round. *gibbon* - an entropy-based method; *lei* is an improved EI; *optuna* - an open-source Bayesian optimizer; *pss* - original MTV sampler; *ts* - standard Thompson sampler w/1024 candidates; *sr* - simple regret. Similar results hold in dimensions 1-300.

Comparisons: Multiple arms/round

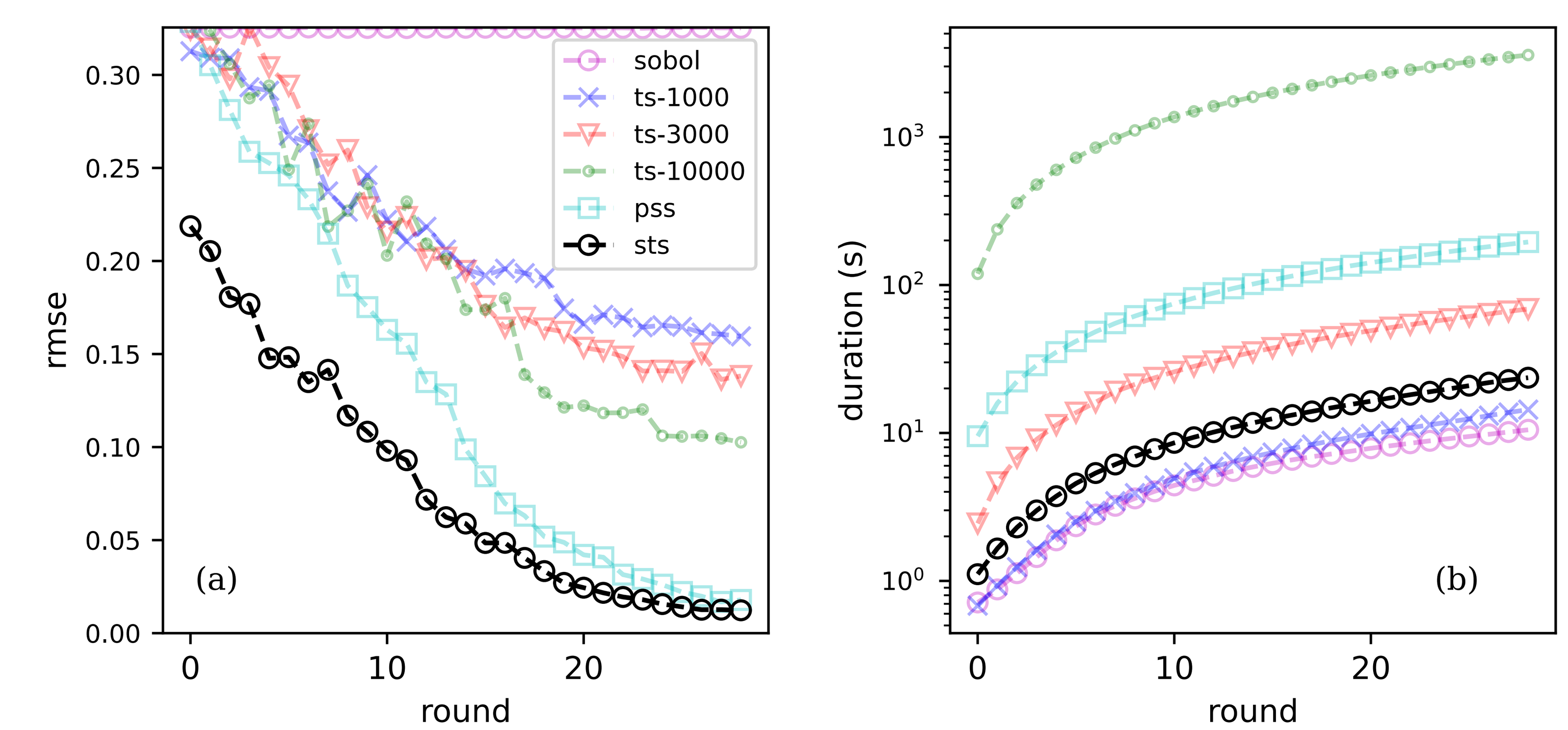
Batching via Minimal Terminal Variance (MTV) + STS



Optimization of nine test functions in 30 dimensions for 3 rounds with 30 replications. *mtv+sts* minimizes $MTV(x_a) \approx \sum_i \sigma^2(x_i | x_a)$ where $x_i \sim p_*(x)$ are sampled by STS and $\sigma^2(x_i | x_a)$ is approximated by a GP. *mtv* minimizes $MTV(x_a)$ where $x_i \sim p_*(x)$ are sampled by PSS. *dpp* is a TS batching method that promotes diversity among the arms. *gibbon* optimizes arms in a batch sequentially, with each arm conditioned on the previous ones. All other methods except *random* and *sobol* maximize the acquisition value over all arms jointly, referred to with a "q" prefix: qEI, qUCB, qLEI, qSR. Similar results hold for dimensions 1-300.

Precision and speed

STS's samples (a) collect more tightly around the maximizer than standard Thompson sampling, and (b) take less time to compute. Sobol' (uniform, not Thompson) sampling is included for comparison.



github.com/yubo-research/yubo