Offline Contextual Bayesian Optimization

Ian Char, Youngseog Chung, Willie Neiswanger, Kirthevasan Kandasamy,
Andrew Oakleigh Nelson, Mark D Boyer, Egemen Koleman, Jeff Schneider

Overview

- In many applications there are several systems or “tasks” to do black-box optimization on.
- Often one optimizes offline (e.g. via a simulator) and both task and configuration can be chosen for evaluation.

We present an algorithm that prudently selects next tasks and configurations to test, and we use this algorithm to learn optimal controls for a nuclear fusion reactor.

Multi-task Thompson Sampling

Algorithm 1 Multi-Task Thompson Sampling (MTS)

Input: capital T, initial capital t_{init}, mean function μ, kernel function σ.
Do random search on tasks in round-robin fashion until t_{init} evaluations are expended.
for $t = t_{init} + 1$ **to** T **do**
  Draw $\widetilde{f} \sim GP(\mu, \sigma)|D_{t-1}$.
  Set $x_t = \underset{x \in \mathcal{X}}{\text{argmax}} \left[\left(\max_{a \in \mathcal{A}} \widetilde{f}(x, a) - \max_{a \in \mathcal{A}_t(x)} \widetilde{f}(x, a) \right) \omega(x) \right]$.
  Set $a_t = \underset{a \in \mathcal{A}}{\text{argmax}} \widetilde{f}(x_t, a)$.
  Observe $y_t = f(x_t, a_t)$.
  Update $D_t = D_{t-1} \cup \{(x_t, a_t, y_t)\}$.
end for
Output: $\hat{h}$

- Use Gaussian Process to model reward.
- Draw samples at every iteration.
- Use samples to identify task where greatest improvement can be made.

Theoretical Guarantees

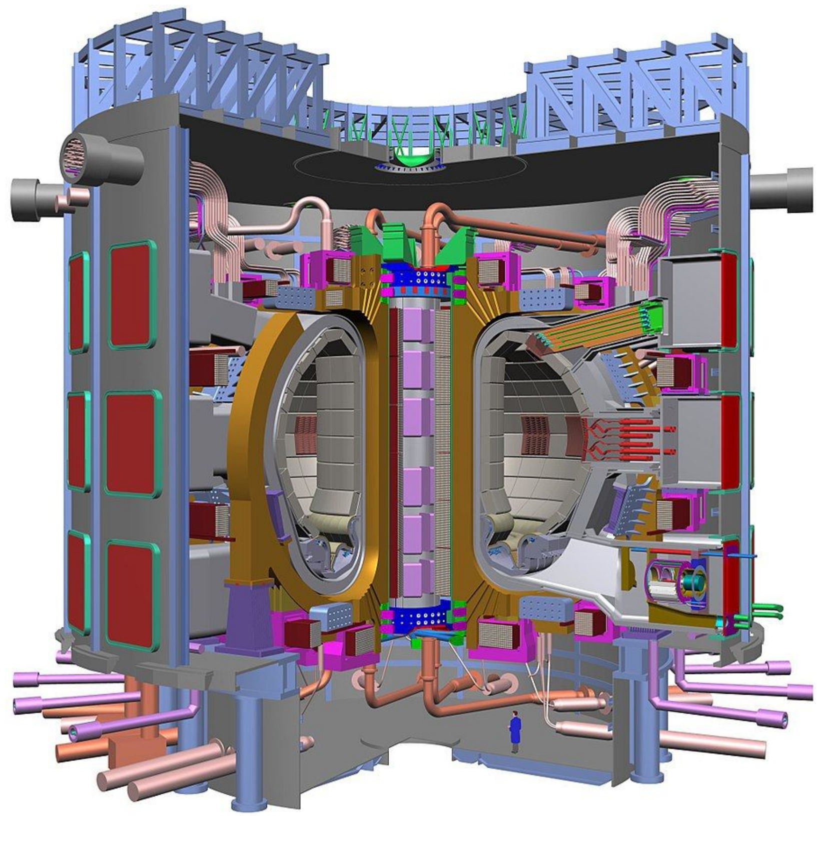
Theorem 1. *Define the maximum information gain to be $\gamma_T := \max_{D_T} I(D_T; f)$, where $I(\cdot;\cdot)$ is the Shannon mutual information. Assume that $\mathcal{X}$ and $\mathcal{A}$ are finite. Then if Algorithm 1 is played for T rounds where $t_{init} = 0$*

$$\mathbb{E}\left[R_{T,f}\right] \leq |\mathcal{X}| \left(\frac{1}{T} + \sqrt{\frac{|\mathcal{X}||\mathcal{A}|\gamma_T}{2T}} \right)$$

- Here $R_{T,f}$ is the normalized *simple regret* summed across all tasks for reward function f after T evaluations.
- Expectation is over reward functions f and randomness in environment.
- This result has the same order as single task regret bounds.

Application to Nuclear Fusion

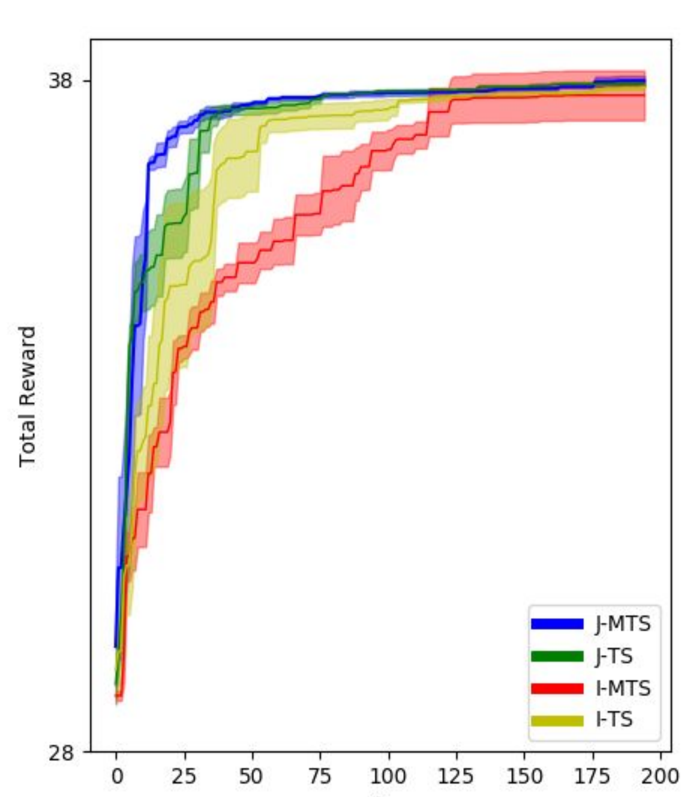
- Nuclear fusion is an extremely promising energy source:
  - Its byproducts are relatively harmless.
  - It has a virtually limitless fuel source.
  - Unlike fission, it has no risk of meltdown.
- One obstacle is that plasma is unstable, leading to short reactions.
- We apply MTS to learn controls for increased plasma stability.

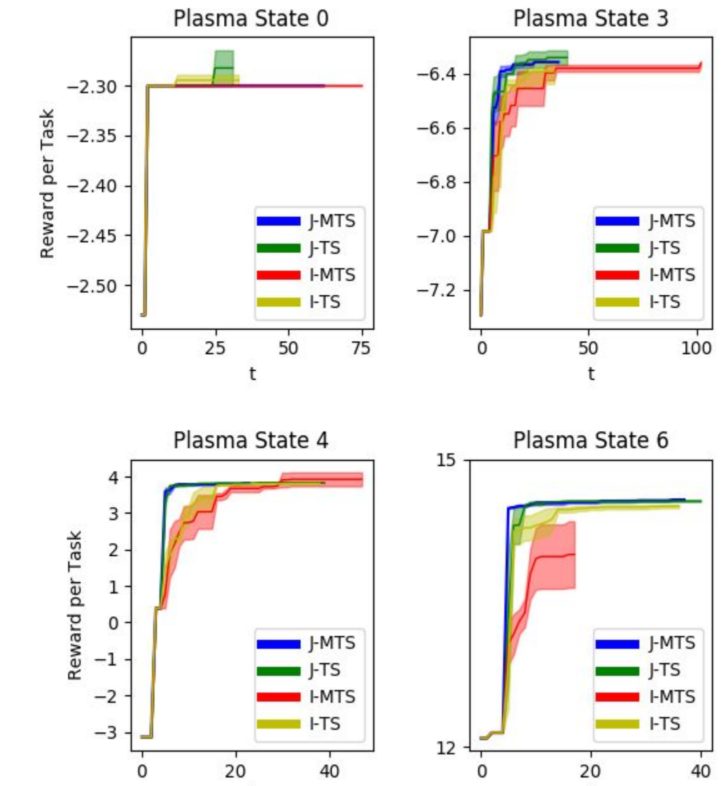


Artist's rendition of International Thermonuclear Experimental Reactor (ITER) published by U.S. Department of Energy.

Learning Controls Offline

- We learn the optimal controls of a tokamak for 7 states of plasma using the predictive mode of the TRANSP simulator.
- For each plasma state (task), we optimize a reward measure which is a combination of plasma stability and reaction efficiency
- States and controls are embedded into a 2D space.

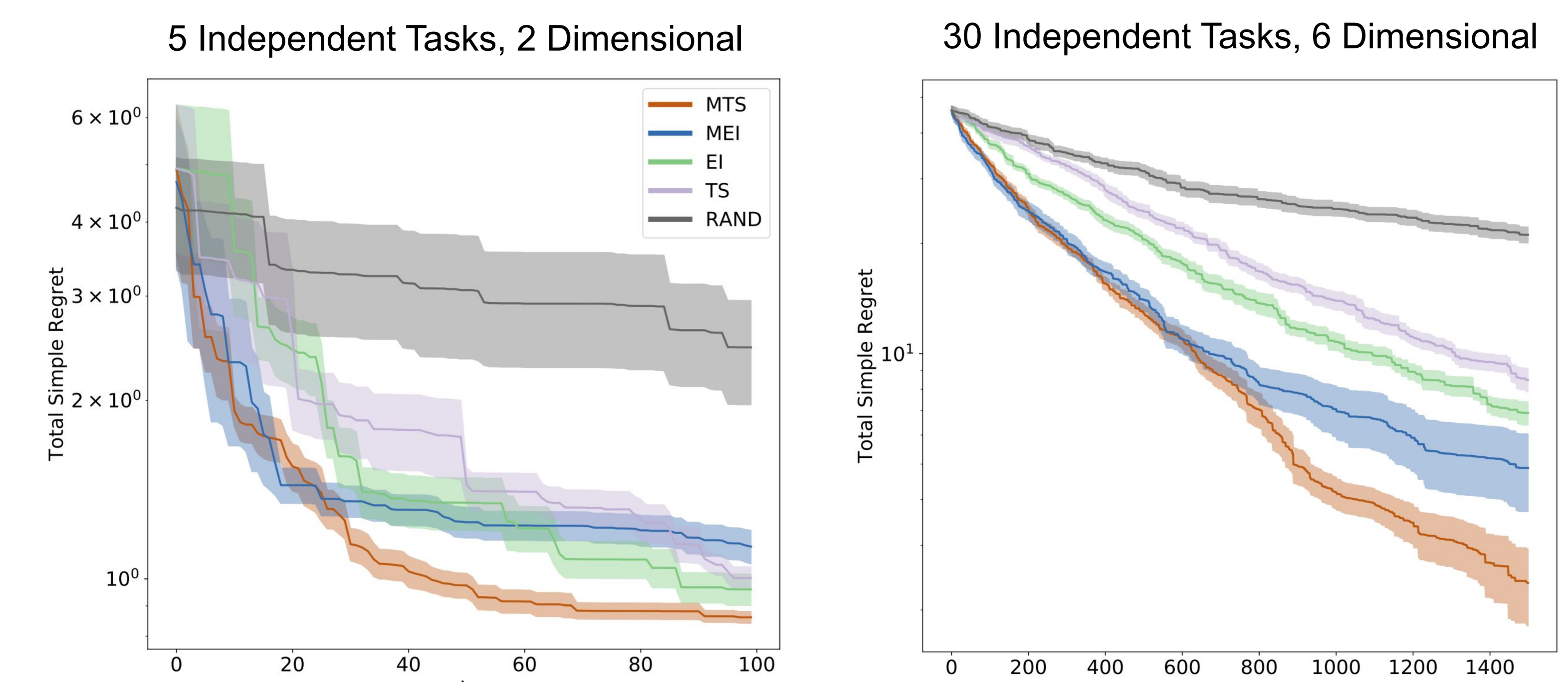




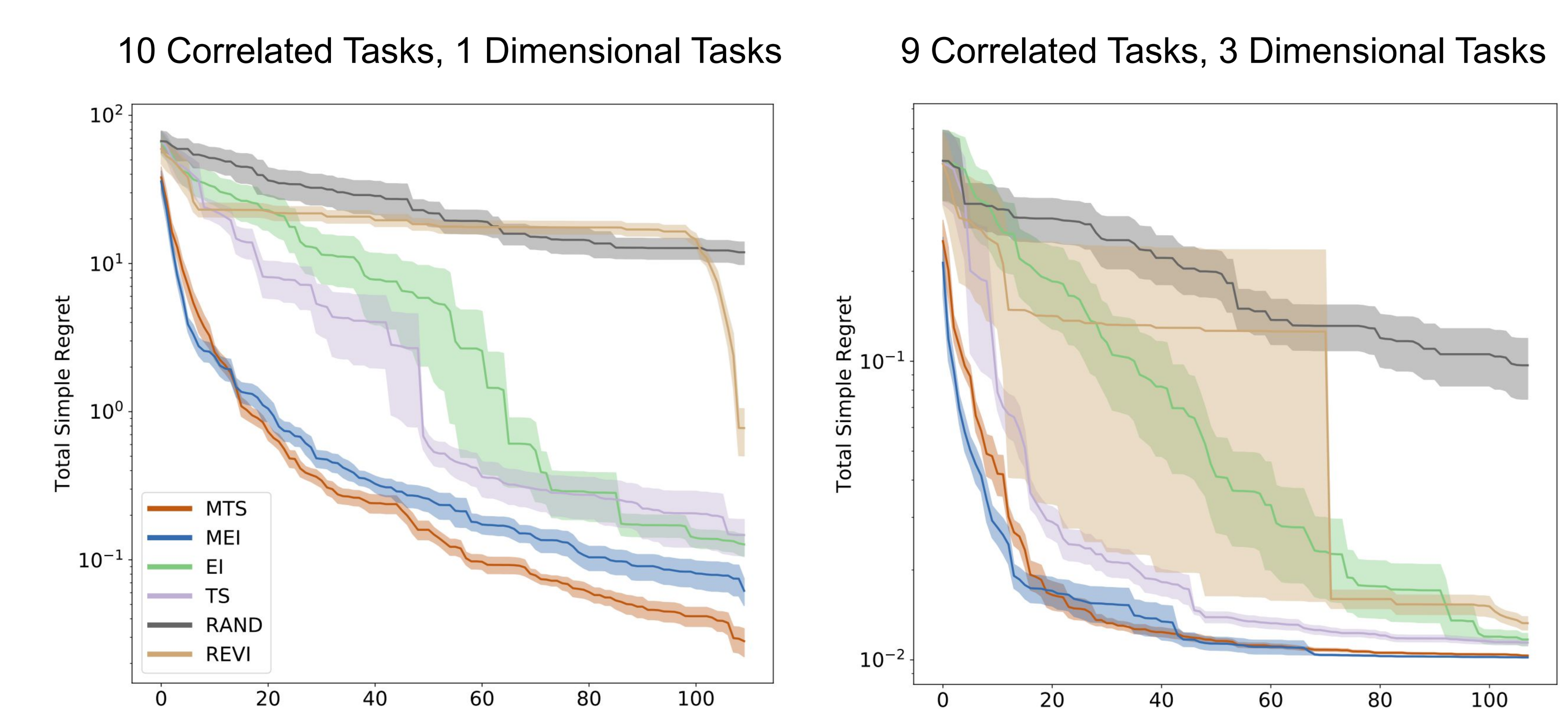
- When reward for tasks are jointly modeled, MTS finds optimal controls more efficiently.
- A joint model over task and action space also provides robustness in handling noisy evaluations (J-MTS vs. I-MTS)
- In the future, we hope to expand number of states and use the results to create a closed loop controller to deploy on a physical device.

Synthetic Experiments

- When tasks are independent, each can be modeled with its own GP.



- When tasks are correlated, a single GP can jointly model all reward structures.



- MTS dominates when resources are uniformly distributed (EI and TS).
- Often distributes resources in smarter way than EI and knowledge gradient based methods.

Modelling Varying Task Difficulty

- Model selection is vital when jointly modelling tasks.
- When lengthscales do not vary with task, performance of MTS suffers.
- We leverage Gibbs kernel to make locally stationary model.

$$\left(\prod_{i=1}^{P_X} \exp\left(\frac{-(x_i - x_i')^2}{2\ell_i^2}\right)\right)\left(\prod_{j=1}^{P_A} \sqrt{\frac{2\ell_j(x)\ell_j(x')}{\ell_j^2(x) + \ell_j^2(x')}} \exp\left(\frac{-(a_j - a_j')^2}{\ell_j^2(x) + \ell_j^2(x')}\right)\right)$$