Path Structured Schrödinger Bridge for Probabilistic Learning of Hardware Resource Usage by Control Software

Georgiy Antonovich Bondar

Department of Applied Mathematics at University of California, Santa Cruz

2024 American Control Conference, July 12

In collaboration with

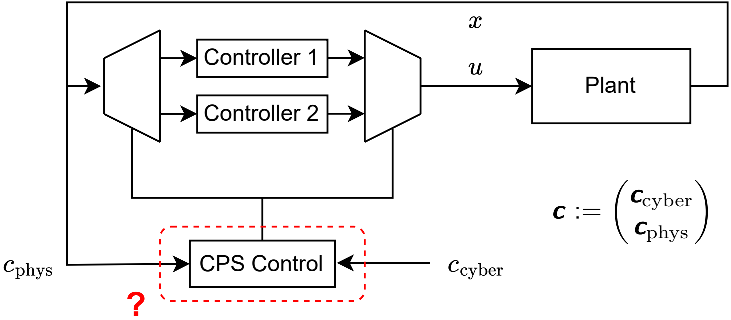
Robert Gifford (UPenn)

Linh T.X. Phan (UPenn)

Abhishek Halder (Iowa State)

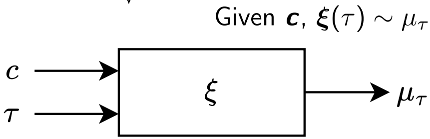
2111688

Motivation



$$\boldsymbol{c} := \begin{pmatrix} \boldsymbol{c}_{\text{cyber}} \\ \boldsymbol{c}_{\text{phys}} \end{pmatrix}$$

$\Downarrow$



$\boldsymbol{c}$-dependent stochastic model of resource usage

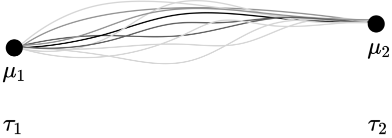
This model must be learned from data! $(\mu_{\tau_1}, \mu_{\tau_2}, \ldots, \mu_{\tau_s})$

The Classical (Bimarginal) Schrödinger Bridge Problem

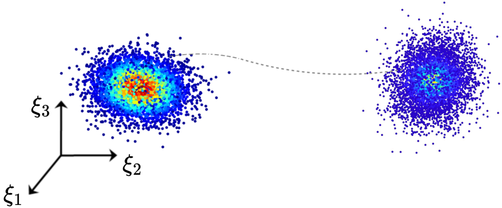
Problem (Discrete SBP)

Let $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \Delta^{n-1} \subset \mathbb{R}^n$ *and* $C \in \mathbb{R}^{n\times n}_{\geq 0}$.

$$\min_{M \in \mathbb{R}^{n\times n}_{\geq 0}} \ \langle C + \varepsilon \log M, M \rangle \quad \textit{subject to} \quad \mathrm{proj}_\sigma (M) = \boldsymbol{\mu}_\sigma \quad \forall \sigma \in \{1, 2\}$$



Darker path $\rightarrow$ higher entropy



Most likely distributional path

The Discrete Multimarginal Schrödinger Bridge Problem

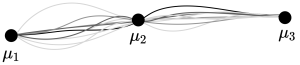
Problem (MSBP)

Let $s \in \mathbb{N}_{\geq 2}$, $\boldsymbol{\mu}_{i \in [\![s]\!]} \in \Delta^{n-1} \subset \mathbb{R}^n$, *and* $\boldsymbol{C} \in (\mathbb{R})^{\otimes s}_{\geq 0}$.

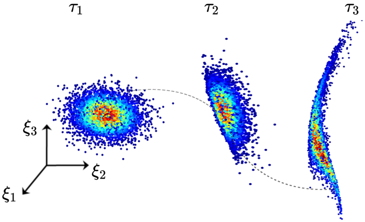
$$\min_{\boldsymbol{M} \in (\mathbb{R}^n)^{\otimes s}_{\geq 0}} \langle \boldsymbol{C} + \varepsilon \log \boldsymbol{M}, \boldsymbol{M} \rangle$$

$$\textit{subject to} \;\; \mathrm{proj}_\sigma(\boldsymbol{M}) = \boldsymbol{\mu}_\sigma \quad \forall \sigma \in [\![s]\!]$$

$\boldsymbol{M}_{\mathrm{opt}}$ most likely distributional path



Strictly convex program in n^s decision variables



Not computationally tractable!

Numerical Solution of the MSBP

The Sinkhorn Iterative Scheme

1. Define $\boldsymbol{K} := \exp(-\boldsymbol{C}/\varepsilon) \in (\mathbb{R}^n)^{\otimes s}_{>0}$ and initialize $\boldsymbol{u}_\sigma := \exp(\boldsymbol{\lambda}_\sigma/\varepsilon)$.
2. Perform the Sinkhorn iterations until (linear) convergence:

$$\boldsymbol{u}_\sigma \leftarrow \boldsymbol{u}_\sigma \otimes \boldsymbol{\mu}_\sigma \oslash \mathrm{proj}_\sigma(\boldsymbol{K} \odot \boldsymbol{U}) \quad \forall \sigma \in [\![s]\!]$$

3. $\boldsymbol{M}_{\mathrm{opt}} = \boldsymbol{K} \odot \boldsymbol{U}$, where $\boldsymbol{U} := \otimes_{\sigma=1}^{s} \boldsymbol{u}_\sigma \in (\mathbb{R}^n)^{\otimes s}_{>0}$.

$$\left[\mathrm{proj}_\sigma\left(\boldsymbol{M}\right)_j\right] = \sum_{i_1,\ldots,i_{\sigma-1},i_{\sigma+1},\ldots,i_s} \boldsymbol{M}_{i_1,\ldots,i_{\sigma-1},j,i_{\sigma+1},\ldots,i_s}$$

$$\downarrow$$

$\mathcal{O}(n^s)$ – exponential complexity in s!

Numerical Solution of the Path-Structured MSBP

$$\mu_1 - \mu_2 - \cdots - \mu_\sigma - \cdots - \mu_s$$

Main Idea

Exploit structure of $\boldsymbol{K} = \exp(-\boldsymbol{C}/\varepsilon)$ to efficiently compute $\mathrm{proj}_\sigma(\boldsymbol{K} \odot \boldsymbol{U})$

Path-structured cost: $[\boldsymbol{C}_{i_1,\dots,i_s}] = \sum_{\sigma=1}^{s-1} \left[C^{\sigma\to\sigma+1}_{i_\sigma,i_{\sigma+1}}\right]$

$$\mathrm{proj}_\sigma(\boldsymbol{K} \odot \boldsymbol{U}) = \left(\boldsymbol{u}_1^\top K^{1\to2} \prod_{j=2}^{\sigma-1} \mathrm{diag}(\boldsymbol{u}_j) K^{j\to j+1}\right)^\top \odot \boldsymbol{u}_\sigma \odot \left(\left(\prod_{j=\sigma+1}^{s-1} K^{j-1\to j}\,\mathrm{diag}(\boldsymbol{u}_j)\right) K^{s-1\to s} \boldsymbol{u}_s\right) \quad \forall \sigma \in [\![s]\!]$$

$\mathcal{O}\left((s-1)n^2\right)$

Linear in s!

The MSBP for Software Resource Usage Prediction

Fix a context $\boldsymbol{c} := \begin{pmatrix} \boldsymbol{c}_{\text{cyber}} & \boldsymbol{c}_{\text{phys}} \end{pmatrix}^\top$.

Profiling: n times, s snapshots at $\tau_1 \equiv 0 < \tau_2 < \cdots < \tau_{s-1} < \tau_s = t$

$$\boldsymbol{\xi}^{i \in [\![n]\!]}(\tau_\sigma), \quad \textit{scattered data} \text{ at } \tau_\sigma$$

$$\Downarrow$$

$$\boldsymbol{\mu}_\sigma := \frac{1}{n} \sum_{i=1}^{n} \delta(\boldsymbol{\xi} - \boldsymbol{\xi}^i(\tau_\sigma)), \quad \text{marginals}$$

$\Downarrow$ Solve path-structured MSBP ($\boldsymbol{M}_{\text{opt}}$)

$$\hat{\mu}_\tau := \sum_{i=1}^{n} \sum_{j=1}^{n} \left[M_{i,j}^{\sigma \to \sigma+1} \right] \delta(\boldsymbol{\xi} - \hat{\boldsymbol{\xi}}(\tau, \boldsymbol{\xi}^i(\tau_\sigma), \boldsymbol{\xi}^j(\tau_{\sigma+1}))), \quad \text{prediction}$$

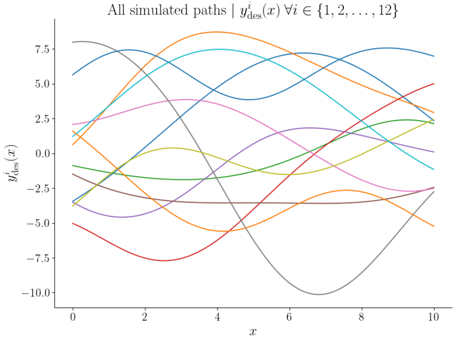
Case Study: A KBM Path-tracking NMPC

$$\boldsymbol{c}_{\mathrm{cyber}} = \begin{pmatrix} \text{alloc. last-level cache} \\ \text{alloc. memory bandwidth} \end{pmatrix}, \; \boldsymbol{c}_{\mathrm{phys}} = y_{\mathrm{des}}(x) \in \mathrm{GP}\left([x_{\min}, x_{\max}]\right)$$

$$\boldsymbol{\xi} := \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix} = \begin{pmatrix} \text{instructions retired} \\ \text{LLC requests} \\ \text{LLC misses} \end{pmatrix}$$

Plant: Kinematic bicycle model

Controller: Nonlinear MPC



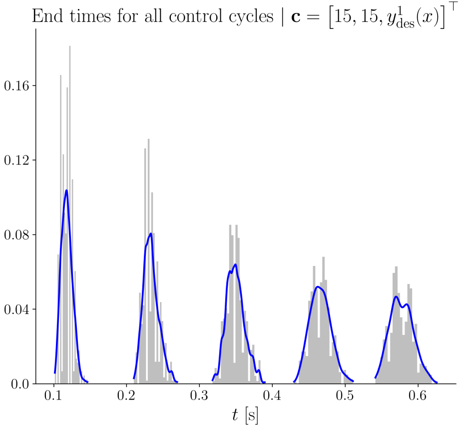
Case Study (Profiling)

$$n = 500\ ,\ \boldsymbol{c}_{\text{cyber}} = \begin{bmatrix}15 & 15\end{bmatrix}^\top\ ,\ \boldsymbol{c}_{\text{phys}} = y^1_{\text{des}}(x)$$

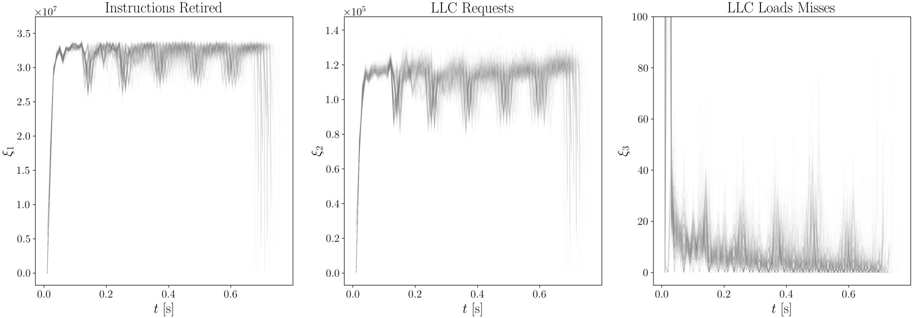
30MB LLC, memory bandwidth

Each profile $n_c = 5$ control cycles

Cycle No.	Mean	Std. Dev.
#1	0.1181	0.0076
#2	0.2336	0.0106
#3	0.3495	0.0127
#4	0.4660	0.0143
#5	0.5775	0.0159



Case Study (Profiling)



Sampling period = 5ms

Hardware-level stochasticty, fixed context $\boldsymbol{c}$

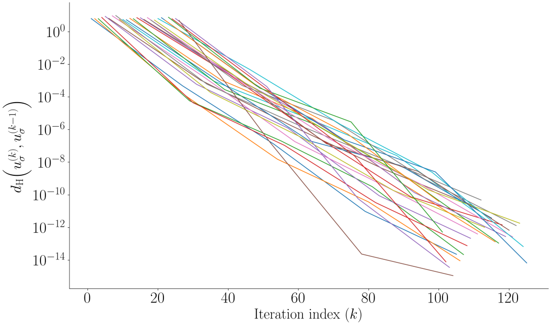
Case Study (Solving MSBP)

Num. marginals: $s := 1 + n_c(s_{\text{int}} + 1)$; Euclidean $C^{\sigma\to\sigma+1}$ $\forall\sigma \in [\![s-1]\!]$

$[\boldsymbol{C}_{i_1,\ldots,i_s}] = \sum_{\sigma=1}^{s-1}\left[C^{\sigma\to\sigma+1}_{i_\sigma,i_{\sigma+1}}\right]$

Rapid linear convergence of Sinkhorn iterations

E.g. $s_{\text{int}} = 4 \Rightarrow n^s = 500^{26}$; convergence in $\approx$ 10s in MATLAB



Hilbert (proj.) metric $d_{\mathrm{H}}(\boldsymbol{u},\boldsymbol{v}) = \log\left(\dfrac{\max_{i=1,\ldots,n} u_i/v_i}{\min_{i=1,\ldots,n} u_i/v_i}\right)$, $\boldsymbol{u},\boldsymbol{v} \in \mathbb{R}^n_{>0}$

Case Study (Results)

$s_{\text{int}} \in \{0, 1, 2, 3, 4\}$; $s_{\text{int}} + 1$ interpolations/control cycle

s_{int}	W_1	W_2	W_3	W_4	W_5
0	2.0489	-	-	-	-
1	2.2695	1.1750	-	-	-
2	5.7717	0.9163	0.3794	-	-
3	2.2413	1.6432	1.2345	0.6010	-
4	0.6372	1.2691	0.9176	0.6689	0.2111

Table: Number of intracycle marginals s_{int} vs. Wasserstein distances $W_j := W(\hat{\mu}_{\hat{\tau}_j}, \mu_{\hat{\tau}_j})$, where $j \in [\![s_{\text{int}} + 1]\!]$. All entries are scaled up by 10^4.

$$\uparrow s_{\text{int}} \quad \Longrightarrow \quad \downarrow \mathbb{E}[W_j]$$

Case Study (Results)

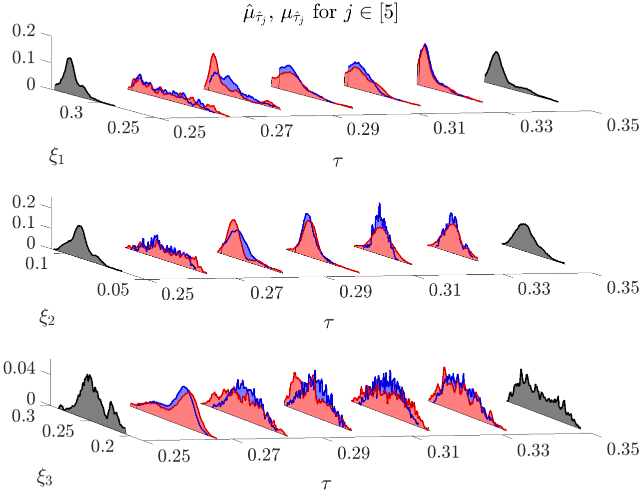
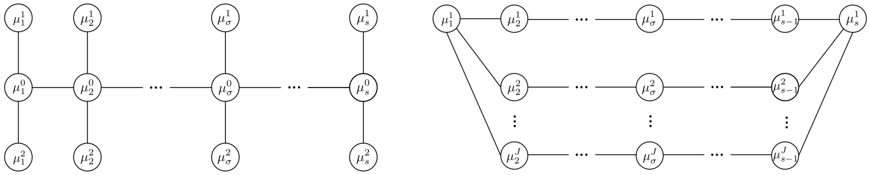


Figure: Predicted $\hat{\mu}_{\hat{\tau}_j}$ vs. measured $\mu_{\hat{\tau}_j}$ at times $\hat{\tau}_{j\in[\![5]\!]}$ during the 3rd control cycle with $s_{\text{int}} = 4$. Distributions at the control cycle boundaries are in *black*.

Ongoing Work

Extension to multi-core software $\Rightarrow$ more complex graph structure of $\boldsymbol{C}$:



Preprint: arXiv:2405.12463

Dynamic scheduling using the learned model for prediction (joint with UPenn)

Thank You

2111688