Bridging Machine Learning and Optimization with JuMP

Andrew Rosemberg

∗AI4OPT, Georgia Institute of Technology, arosemberg3@gatech.edu

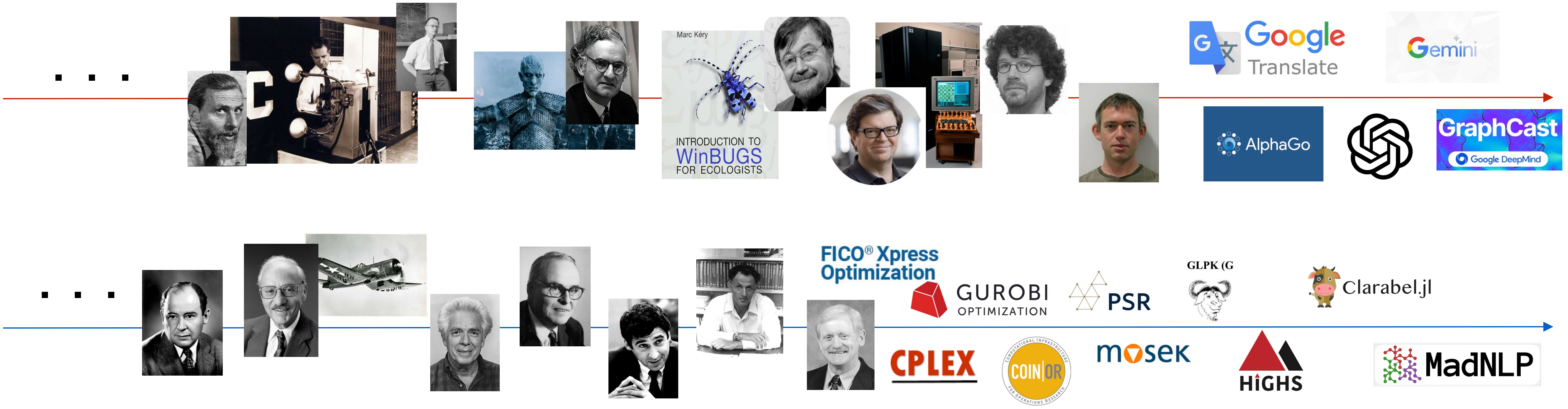
Highlights

AI Institute for Advances in Optimization

- AI & Optimization: Friends and Foes.
- JuMP and Deep Learning: How to Train Optimization Surrogates?
- Julia’s Multiple Dispatch: It is Great when Things Just Work.
- JuMP and Reinforcement Learning: How to Train Policies?
- Hopes and Plans for the Future!

AI & Optimization: Friends and Foes

AI Institute for Advances in Optimization



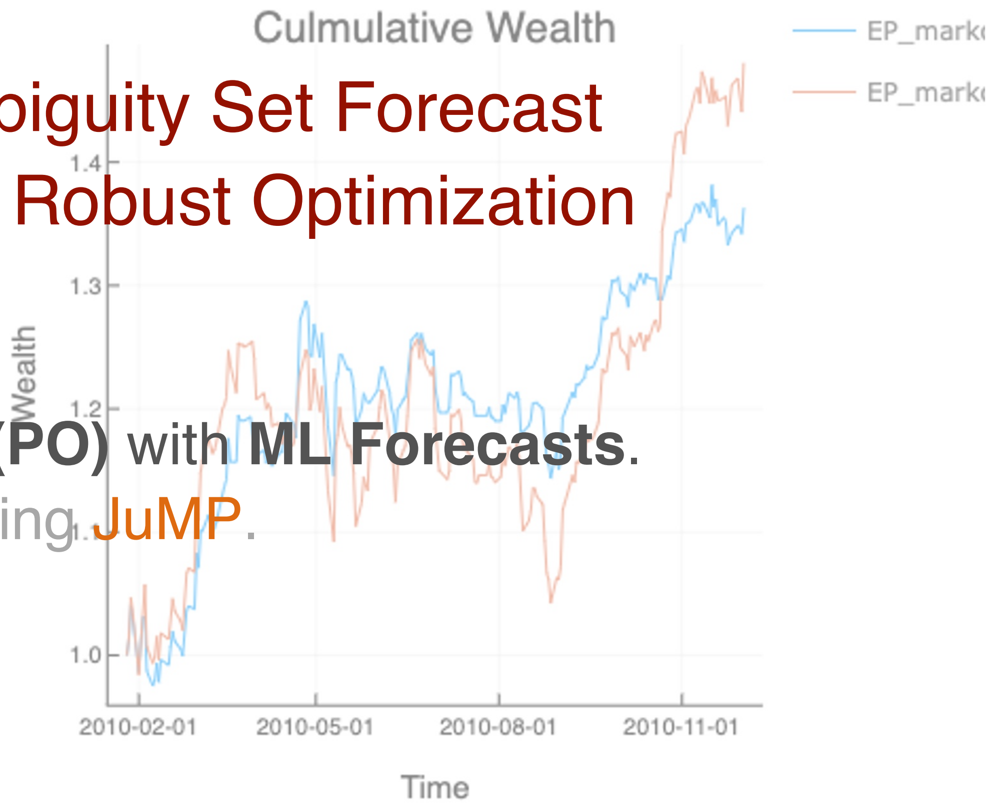
- AI & ML mostly concerned with Descriptive / Predictive Problems
  - RL: Failures to then positive results
- Optimization focused on Prescriptions
  - Uncertainty: Stochastic / Robust Optimization

AI & Optimization: Finally, Friends?

AI Institute for Advances in Optimization

- Predict than Optimize
  - Point Prediction / Scenario Selection / Ambiguity Set Forecast
  - Deterministic / Stochastic / Distributionally Robust Optimization

At [redacted],
JuMP allowed complicated **Portfolio Optimization (PO)** with **ML Forecasts**.
» PortfolioOpt.jl: Portfolio Optimization (PO) using JuMP.



- Predict than Optimize – than loop
  - Application's aware objective for ML models

DiffOpt (Differential Optimization) allowed us to feedback financial PO performance to ML-Models.

AI & Optimization: Teammates!

AI Institute for Advances in Optimization

- Parametric - Optimization Problems
  - Repetitive (Learnable) structure – What can we do?

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & \mathbf{c}_\beta^\top \mathbf{x} \\
\text{s.t.} \quad & \mathbf{A}_\beta \mathbf{x} = \mathbf{b}_\beta \\
& \mathbf{x} > 0
\end{aligned}
$$

Proxy

$$\beta \rightarrow f(x)$$

$$\beta \rightarrow x$$

$$\beta \rightarrow \mathbf{c}_\beta^\top \mathbf{x}$$

Old News: Learning Value Functions

AI Institute for Advances in Optimization

$$\mathbf{x} \to \mathbb{E}[\mathcal{Q}(\mathbf{x}, \omega)]$$

$$1^{\text{st}} \text{ Stage} \begin{cases} \min\limits_{\mathbf{x}} & c^T\mathbf{x} + \mathbb{E}[\mathcal{Q}(\mathbf{x}, \omega)] \\ \text{s.t.} & \mathbf{x} \in \mathcal{X} \end{cases}$$

Convexity for Tractability

$$2^{\text{nd}} \text{ Stage} \left\{ \mathcal{Q}(\mathbf{x}, \omega) = \left\{ \begin{array}{ll} \min\limits_{y} & g^T y \\ \text{s.t.} & Ay = b(\omega) - H\mathbf{x} \\ & y \in \mathcal{Y} \end{array} \right\} . \right.$$

Using Value Functions Approximations

- Cutting Planes

$$\begin{aligned} \min_{\mathbf{x}} \quad & c^T\mathbf{x} + \Theta \\ \text{s.t.} \quad & \Theta \geq \bar{g}_k + \bar{\pi}_k^T H(-\mathbf{x}) \qquad \forall k = 1..K \end{aligned}$$

$\bar{\pi}$ $\bar{g}$

$\mathbf{x}$

$$\begin{aligned} \min_{y} \quad & g^T y \\ \text{s.t.} \quad & Ay = b(\omega) - H \ldots \pi(\omega) \\ & y \in \mathcal{Y} \end{aligned}$$

Expensive

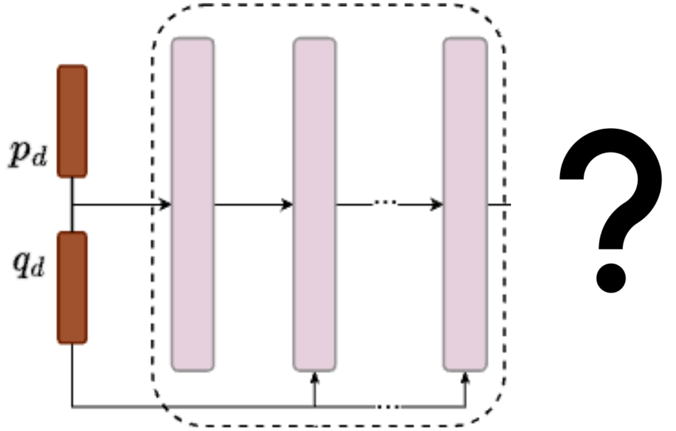
Using Value Functions Approximations

- Cutting Planes

$$\begin{aligned} \min_{\mathbf{x}} \quad & c^T\mathbf{x} + \Theta \\ \text{s.t.} \quad & \Theta \geq \bar{g}_k + \bar{\pi}_k^T H(-\mathbf{x}) \qquad \forall k = 1..K \end{aligned}$$

$\bar{\pi}$ $\bar{g}$

$\mathbf{x}$



$\mathbf{x} \rightarrow \mathbb{E}[\mathcal{Q}(\mathbf{x}, \omega)]$

Input-Convex Neural Network (ICNN)

AI Institute for Advances in Optimization

$$\mathbf{x}^k = h^k(\mathbf{x}^{k-1}) = \mathrm{ReLU}(W^k\mathbf{x}^{k-1} + H^k\mathbf{x}^0 + d^k), \qquad (9)$$

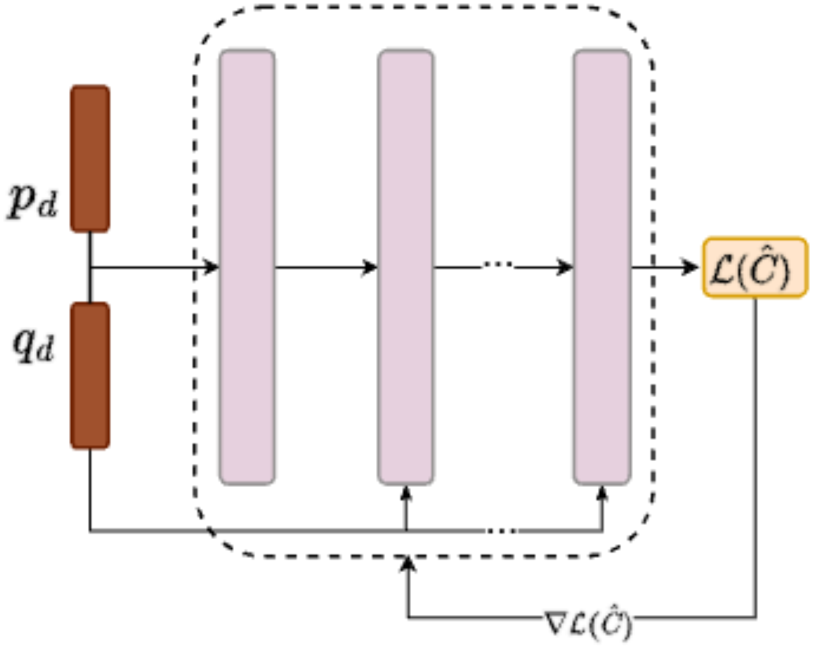
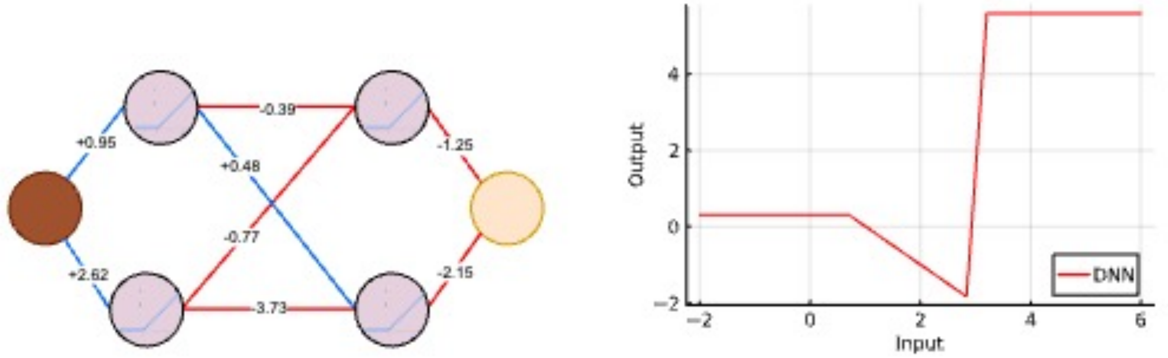
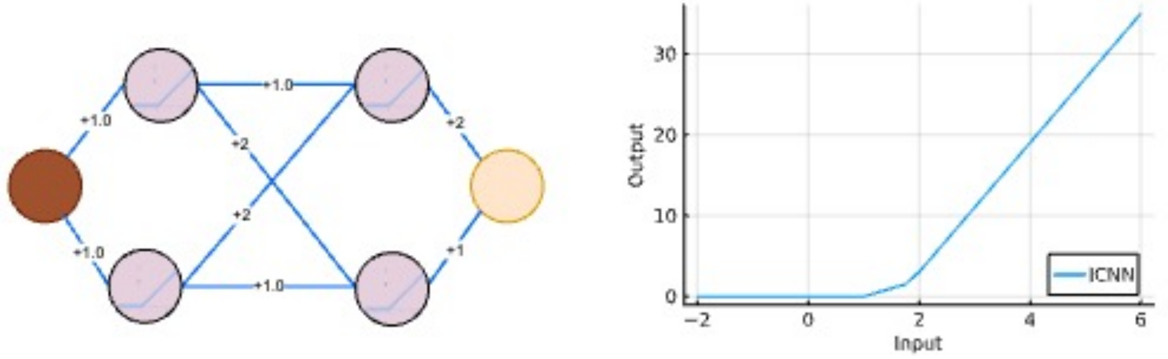


Fig. 2. Fully Connected ICNN



(a) Example DNN (left) and its Output (right). The DNN defines a non-convex function.



(b) Example ICNN (left) and its Output (right). All ICNN weights are positive and it defines a convex function.

Fig. 1. Illustration of Input-Convex Neural Networks.

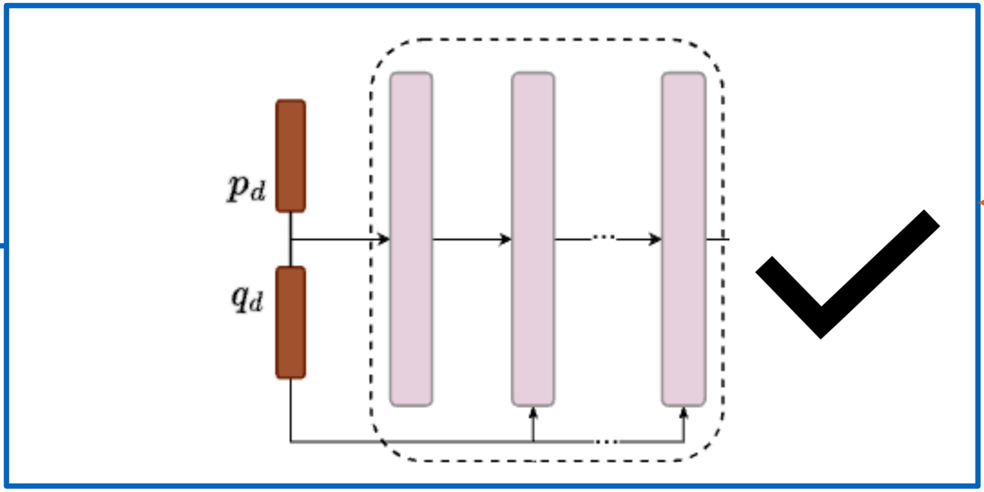
Using Value Functions Approximations

- Cutting Planes

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & c^T\mathbf{x} + \Theta \\
\text{s.t.} \quad & \Theta \geq \bar{g}_k + \bar{\pi}_k^T H(-\mathbf{x}) \qquad\qquad \forall k = 1..K
\end{aligned}
$$

$\bar{\pi}$ $\bar{g}$

$\mathbf{x}$



Can ICNNs approximate OPF value functions?

$$1^{\text{st}}\ \text{Stage}\begin{cases}\min\limits_{\mathbf{x}} & c^T\mathbf{x} + \mathbb{E}[\mathcal{Q}(\mathbf{x},\omega)] \\ \text{s.t.} & \mathbf{x} \in \mathcal{X}\end{cases}$$

$$2^{\text{nd}}\ \text{Stage}\left\{\ \mathcal{Q}(\mathbf{x},\omega) = \right.$$

Model 1 The AC-OPF Model

$$
\begin{aligned}
\min \quad & \sum_{i \in \mathcal{N}} c_i \mathbf{p}_i^{\text{g}} & \\
\text{s.t.} \quad & \mathbf{S}_i^{\text{g}} - \text{S}_i^{\text{d}} - (Y_i^s)^\star |\mathbf{V}_i|^2 = \sum_{ij \in \mathcal{E} \cup \mathcal{E}^R} \mathbf{S}_{ij}^{\text{f}} & \forall i \in \mathcal{N} \\
& \mathbf{S}_{ij}^{\text{f}} = (Y_{ij} + Y_{ij}^c)^\star |\mathbf{V}_i|^2 - Y_{ij}^\star \mathbf{V}_i \mathbf{V}_j^\star & \forall ij \in \mathcal{E} \\
& \mathbf{S}_{ji}^{\text{f}} = (Y_{ij} + Y_{ji}^c)^\star |\mathbf{V}_j|^2 - Y_{ij}^\star \mathbf{V}_i^\star \mathbf{V}_j & \forall ij \in \mathcal{E} \\
& |\mathbf{S}_{ij}^{\text{f}}|, |\mathbf{S}_{ji}^{\text{f}}| \leq \bar{s}_{ij} & \forall ij \in \mathcal{E} \\
& \underline{\text{v}}_i \leq |\mathbf{V}_i| \leq \bar{\text{v}}_i & \forall i \in \mathcal{N} \\
& \underline{\text{p}}_i^{\text{g}} \leq \mathbf{p}_i^{\text{g}} \leq \bar{\text{p}}_i^{\text{g}} & \forall i \in \mathcal{N} \\
& \underline{\text{q}}_i^{\text{g}} \leq \mathbf{q}_i^{\text{g}} \leq \bar{\text{q}}_i^{\text{g}} & \forall i \in \mathcal{N}
\end{aligned}
$$

Yes, they can!

AI Institute for Advances in Optimization

- Demonstrated ICNN efficacy on **large-scale systems**, showing they match DNNs with most optimality gaps below 0.5%.

- For **AC-OPF**, **SOC** relaxation, and **DC-OPF** formulations.

- **Bounds** on ICNN generalization error based on training data performance.

New tool to help efficiently solve larger OPF applications!

Yes, we can!

AI Institute for Advances in Optimization

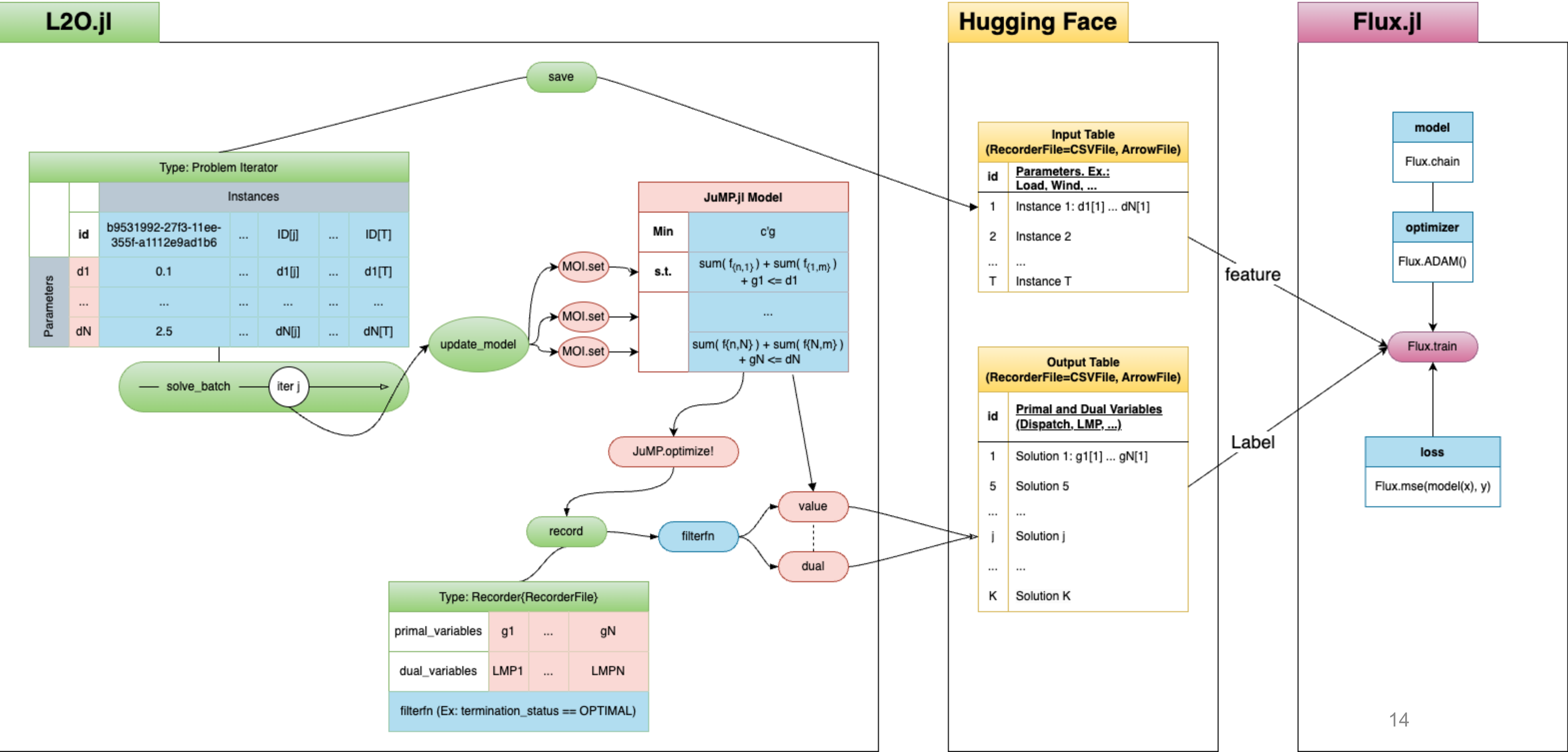
$$\text{gap} = \frac{|\tilde{z} - z^*|}{|z^*|}.$$

TABLE II
ICNN PERFORMANCE RESULTS.

		Mean gap (%)				Worst gap (%)			
System	OPF	ICNN	DNN	DC	SOC	ICNN	DNN	DC	SOC
ieee300	DC	0.15	0.19	0.00	-	1.58	1.90	0.00	-
	SOC	0.31	0.37	5.43	0.00	5.77	4.96	10.79	0.00
	AC	0.39	0.39	7.35	1.98	15.81	14.56	21.95	16.47
pegase1k	DC	0.28	0.33	0.00	-	2.35	1.83	0.00	-
	SOC	0.33	0.82	1.60	0.00	2.22	2.15	1.88	0.00
	AC	0.33	0.68	2.91	1.32	2.37	1.95	3.49	1.89
pegase2k	DC	0.22	0.30	0.00	-	3.45	3.21	0.00	-
	SOC	1.03	0.32	2.15	0.00	3.15	2.36	2.35	0.00
	AC	0.24	0.27	3.02	0.80	8.59	8.89	8.887	9.13
rte6k	DC	0.27	0.38	0.00	-	1.76	1.15	0.00	-
	SOC	0.29	0.57	2.67	0.00	1.69	5.52	3.17	0.00
	AC	0.25	0.33	3.05	0.33	2.71	3.08	3.67	0.36

L2O: JuMP + POI + Flux = ♥

AI Institute for Advances in Optimization



It’s Great when Things Just Work

AI Institute for Advances in Optimization

- What if I want to represent my entire NN in JuMP?

```
function NNlib.relu(ex::AffExpr)
    model = owner_model(ex)
    relu_out = @variable(model, lower_bound = 0.0)
    @constraint(model, relu_out >= ex)
    return relu_out
end
```

```
icnn = Chain(Dense(1000, 10, relu), Dense(10, 10, relu), Dense(10, 1))
train_icnn(icnn)

model = Model(Highs.Optimizer)
@variable(model, x[1:1000] >= 0)
@constraint(model, sum(x) == 1.0)
@objective(model, Min, icnn(x))

JuMP.optimize!(model)
```

```
function NNlib.relu(ex::AffExpr)
    tol = 0.00001
    model = owner_model(ex)
    aux = @variable(model, binary = true)
    relu_out = @variable(model, lower_bound = 0.0)
    @constraint(model, relu_out >= ex * (1-tol))
    @constraint(model, relu_out <= ex * (1+tol) + big_M * (1 - aux))
    @constraint(model, relu_out <= big_M * aux)
    return relu_out
end
```

Learning OPF Value Functions with ICNN

Andrew Rosemberg , Mathieu Tanneau, Bruno Fanzeres, Joaquim Garcia‡ and Pascal Van Hentenryck

This research is partly funded by NSF award 2112533.

What else? Proxies as Direct Policies!

AI Institute for Advances in Optimization

- Multistage Stochastic Programing

$$\min_{(\mathbf{y}_1,\mathbf{x}_1)\in\mathcal{X}_1(\mathbf{x}_0)} f(\mathbf{x}_1,\mathbf{y}_1) + \mathbf{E}[\min_{(\mathbf{y}_2,\mathbf{x}_2)\in\mathcal{X}_2(\mathbf{x}_1,w_2)} f(\mathbf{x}_2,\mathbf{y}_2) + \mathbf{E}[\cdots +$$
$$\mathbf{E}[\min_{(\mathbf{y}_t,\mathbf{x}_t)\in\mathcal{X}_t(\mathbf{x}_{t-1},w_t)} f(\mathbf{x}_t,\mathbf{y}_t) + \mathbf{E}[\cdots]]]$$

$$\begin{aligned} V_t(\mathbf{x}_{t-1},w_t) = \min_{\mathbf{x}_t,\mathbf{y}_t} \quad & f(\mathbf{x}_t,\mathbf{y}_t) + \mathbf{E}[V_{t+1}(\mathbf{x}_t,w_{t+1})] \\ \text{s.t.} \quad & \mathbf{x}_t = \mathcal{T}(\mathbf{x}_{t-1},w_t,\mathbf{y}_t) \\ & h(\mathbf{x}_t,\mathbf{y}_t) \geq 0 \end{aligned}$$

What else? Proxies as Direct Policies!

AI Institute for Advances in Optimization

- Multistage Stochastic Programing

$$\min_{(\mathbf{y}_1,\mathbf{x}_1)\in\mathcal{X}_1(\mathbf{x}_0)} f(\mathbf{x}_1,\mathbf{y}_1) + \mathbf{E}[\min_{(\mathbf{y}_2,\mathbf{x}_2)\in\mathcal{X}_2(\mathbf{x}_1,w_2)} f(\mathbf{x}_2,\mathbf{y}_2) + \mathbf{E}[\cdots + \mathbf{E}[\min_{(\mathbf{y}_t,\mathbf{x}_t)\in\mathcal{X}_t(\mathbf{x}_{t-1},w_t)} f(\mathbf{x}_t,\mathbf{y}_t) + \mathbf{E}[\cdots]]]$$

$$\begin{aligned} \pi_t^*(\{w_j\}_{j=2..t},\mathbf{x}_0) \in \arg\min_{\mathbf{x}_t,\mathbf{y}_t} \quad & f(\mathbf{x}_t,\mathbf{y}_t) + \mathcal{V}_{t+1}(\mathbf{x}_t) \\ \text{s.t.} \quad & \mathbf{x}_t = \mathcal{T}(\mathbf{x}_{t-1},w_t,\mathbf{y}_t) \\ & h(\mathbf{x}_t,\mathbf{y}_t) \geq 0 \end{aligned}$$

Two-stage general decision rules (TS-GDR)

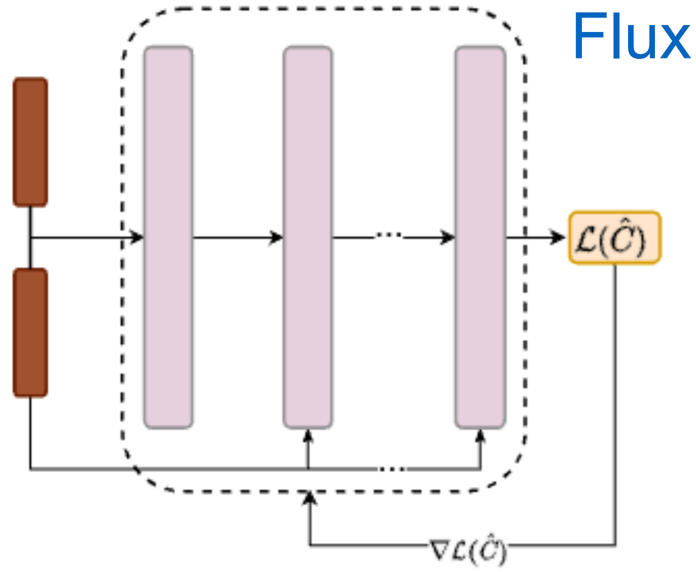
AI Institute for Advances in Optimization

JuMP

$$
\begin{aligned}
\mathcal{Q}(w;\theta) = \min_{\mathbf{x},\mathbf{y},\delta} \quad & \sum_t f(\mathbf{x}_t,\mathbf{y}_t) + C_t^{\delta}||\delta_t|| & \\
\text{s.t.} \quad & \mathbf{x}_t = \mathcal{T}(\mathbf{x}_{t-1}, w_t, \mathbf{y}_t) & \forall t \\
& \mathbf{x}_t + \delta_t = \boxed{\pi_t(\{w_j\}_{j=2..t}, \mathbf{x}_0; \theta_t)} : \lambda_t & \forall t \\
& h(\mathbf{x}_t,\mathbf{y}_t) \geq 0 & \forall t
\end{aligned}
$$

s1

Flux



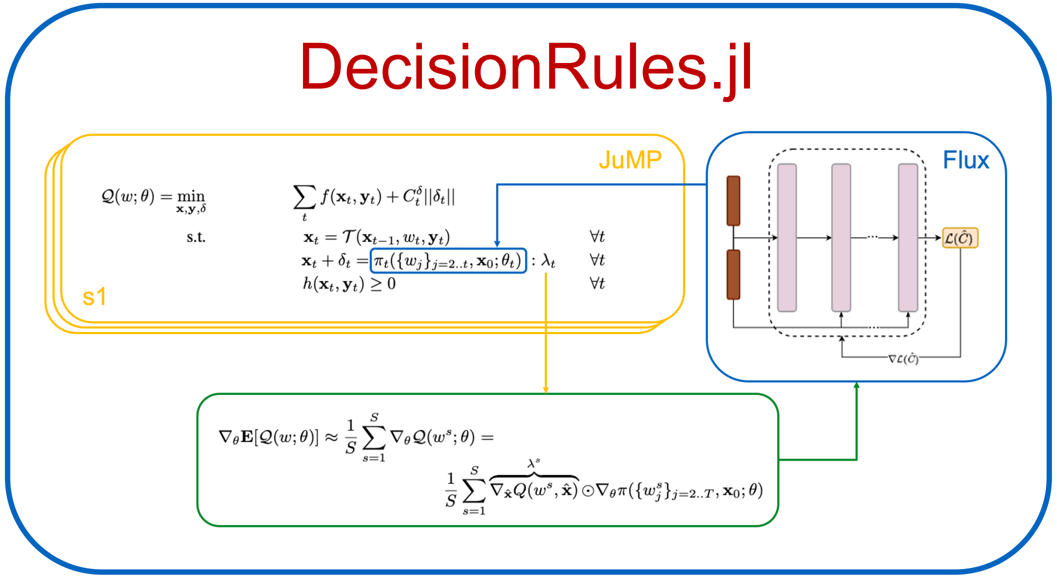
$$
\nabla_\theta \mathbf{E}[\mathcal{Q}(w;\theta)] \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\theta \mathcal{Q}(w^s;\theta) =
$$

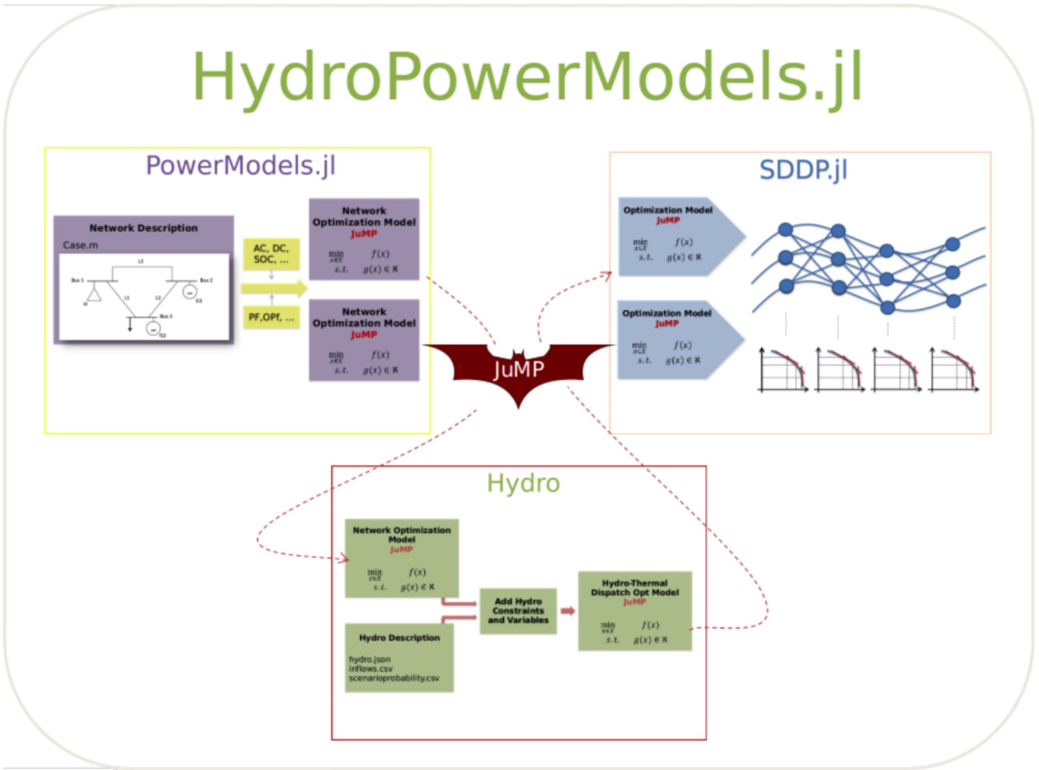
$$
\frac{1}{S}\sum_{s=1}^{S} \overbrace{\nabla_{\hat{\mathbf{x}}} Q(w^s, \hat{\mathbf{x}})}^{\lambda^s} \odot \nabla_\theta \pi(\{w_j^s\}_{j=2..T}, \mathbf{x}_0; \theta)
$$

TS-GDR vs SDDP

Table 3: Comparison of SDDP and ML Decision Rule for Bolivia with AC Implementation.

Model	Plan	Imp Cost (USD)	GAP (%)	Training (Min)	Execution (Min)
TS-DDR	AC	301851(±4876)	-	60	0.067
SDDP	SOC	302816(±5431)	0.32(±2.34)	-	480
TS-LDR	AC	319326(±4715)	5.79(±1.30)	226.01	0.067
SDDP	DCLL	323895(±3944)	7.30(±2.27)	-	320





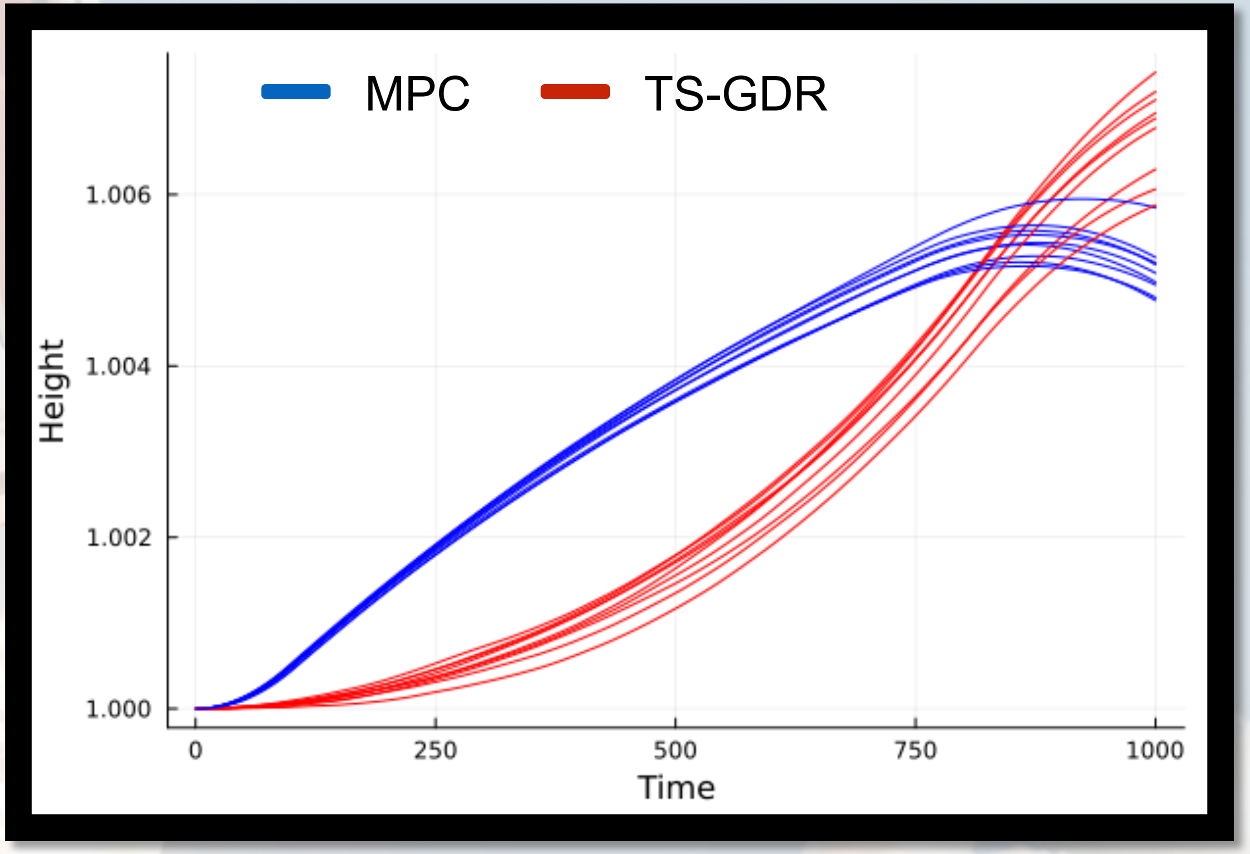
What about Control Problems?

AI Institute for Advances in Optimization

- Stochastic Goddard Rocket

$$
\begin{aligned}
\max_{h,v,m,u} \quad & h_T \\
\text{s.t.} \quad & \frac{h_t - h_{t-1}}{\Delta t} = v_{t-1} \quad \text{for } t = 2, \ldots, T, \\
& \frac{v_t - v_{t-1}}{\Delta t} = \frac{u_{t-1} - D(h_{t-1}, v_{t-1})}{m_{t-1}} - g(h_{t-1}) \boxed{- w_{t-1}} \quad \text{for } t = 2, \ldots, T, \\
& \frac{m_t - m_{t-1}}{\Delta t} = -\frac{u_{t-1}}{c} \quad \text{for } t = 2, \ldots, T, \\
& D(h, v) = D_c v^2 \exp\left(-\frac{h_c(h - h_0)}{h_0}\right), \\
& g(h) = g_0 \left(\frac{h_0}{h}\right)^2 \\
& v_1 = v_0,\ h_1 = h_0,\ m_1 = m_0,\ u_T = 0.0, \\
& v_t \geq 0,\ m_t \geq m_T,\ 0 \leq u_t \leq u_t^{\max} \quad \text{for } t = 1, \ldots, T.
\end{aligned}
$$

$$\boxed{w_t \in \mathcal{N}(0, 1)}$$



Godspeed!

Two-stage general decision rules (TS-GDR)

AI Institute for Advances in Optimization

Andrew Rosemberg, Alexandre Street, Davi M. Valladão, Pascal Van Hentenryck

AI Institute for Advances in Optimization

LAMPS Laboratory of Applied Mathematical Programming and Statistics

Georgia Institute of Technology

PUC RIO

This research is partly funded by NSF award 2112533.

The work of Alexandre Street and Davi Valladão was partially supported by CAPES, CNPq and FAPERJ.

NSF