

The Voice of Optimization

Bartolomeo Stellato

joint work with Dimitris Bertsimas



Stanford, 24 Oct 2019





Inventory management

minimize $\sum_{t=0}^{T-1} h(x_t) + o(u_t)$

subject to $x_{t+1} = x_t + u_t - d_t$
 $x_0 = x_{\text{init}}$
 $0 \leq u_t \leq M$

A photograph showing a close-up view of a metal shelving unit in a warehouse. The shelves are filled with numerous cardboard boxes, each with a white shipping label. The boxes are stacked in several layers across the visible height of the shelves.

Inventory management

minimize $\sum_{t=0}^{T-1} h(x_t) + o(u_t)$

subject to $x_{t+1} = x_t + u_t - d_t$

$x_0 = x_{\text{init}}$

$0 \leq u_t \leq M$

Inventory



Inventory management

minimize $\sum_{t=0}^{T-1} h(x_t) + o(u_t)$

subject to $x_{t+1} = x_t + u_t - d_t$

$x_0 = x_{\text{init}}$

$0 \leq u_t \leq M$

Inventory 

Order 



Inventory management

minimize $\sum_{t=0}^{T-1} h(x_t) + o(u_t)$

subject to $x_{t+1} = x_t + u_t - d_t$

Inventory

$$x_0 = x_{\text{init}}$$
$$0 \leq u_t \leq M$$

Order

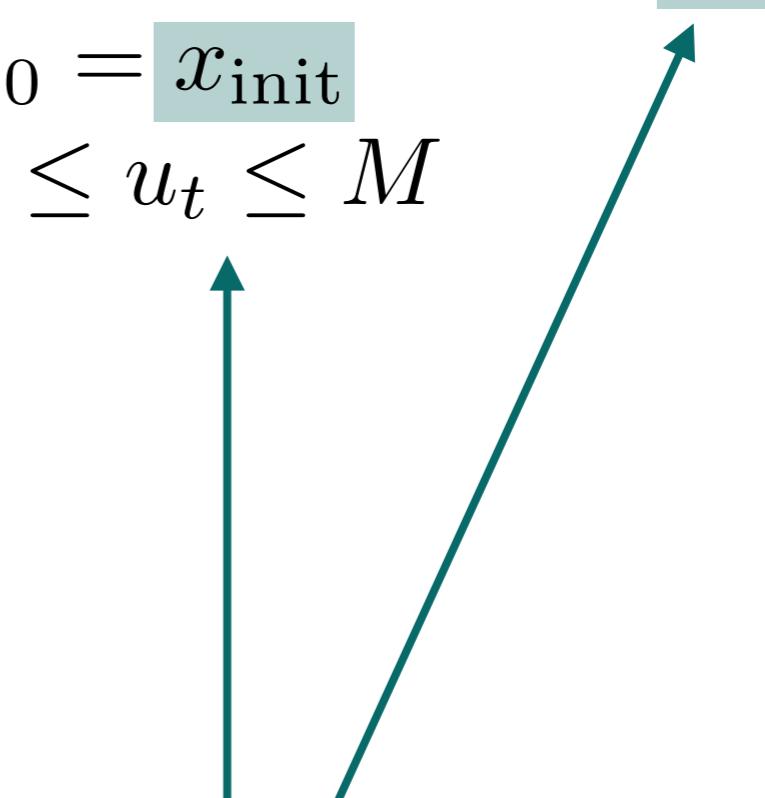
Demand

A photograph showing a close-up view of a metal shelving unit in a warehouse. The shelves are filled with numerous cardboard boxes, each with a white shipping label. The boxes are stacked in several layers across the visible height of the shelves.

Inventory management

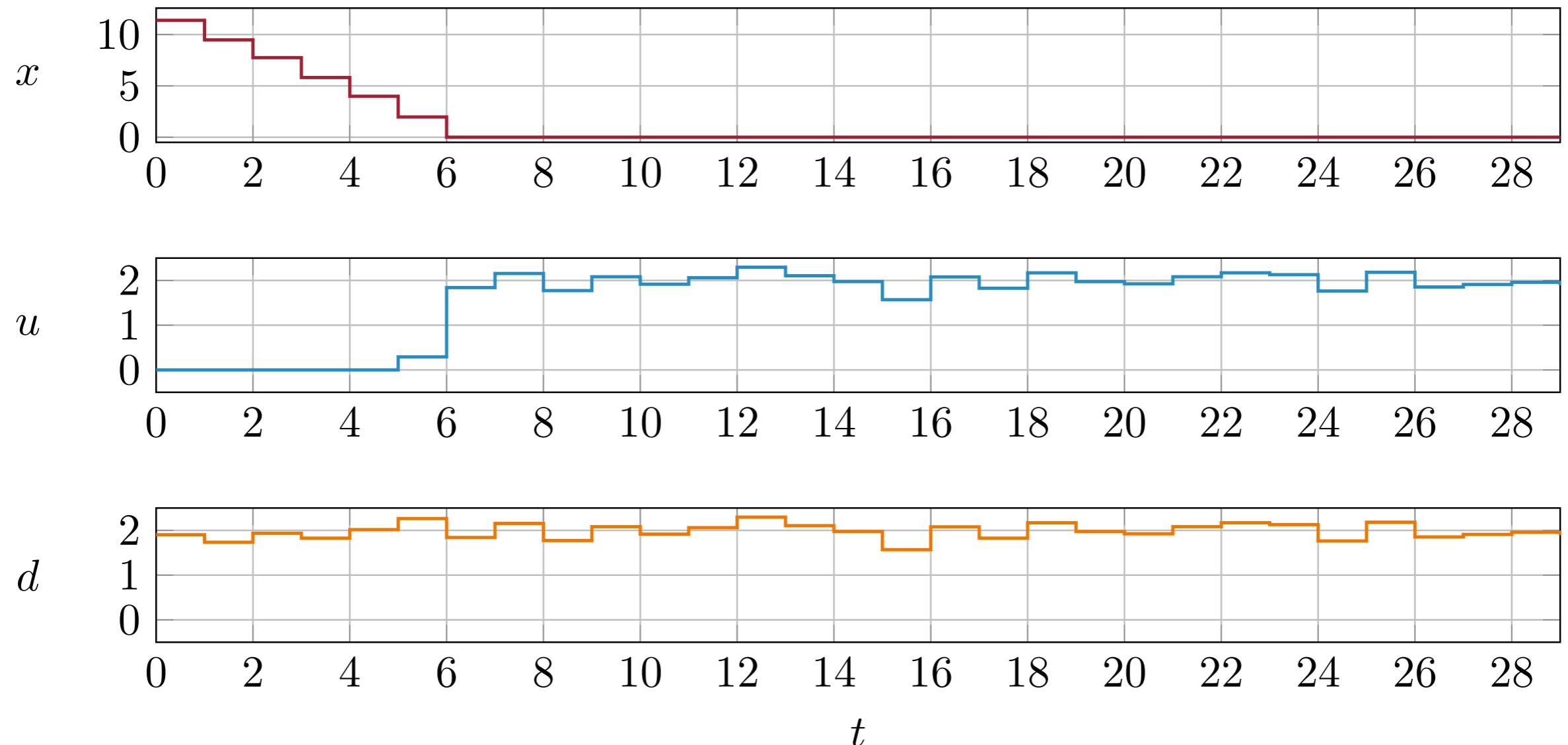
minimize $\sum_{t=0}^{T-1} h(x_t) + o(u_t)$

subject to $x_{t+1} = x_t + u_t - d_t$
 $x_0 = x_{\text{init}}$
 $0 \leq u_t \leq M$

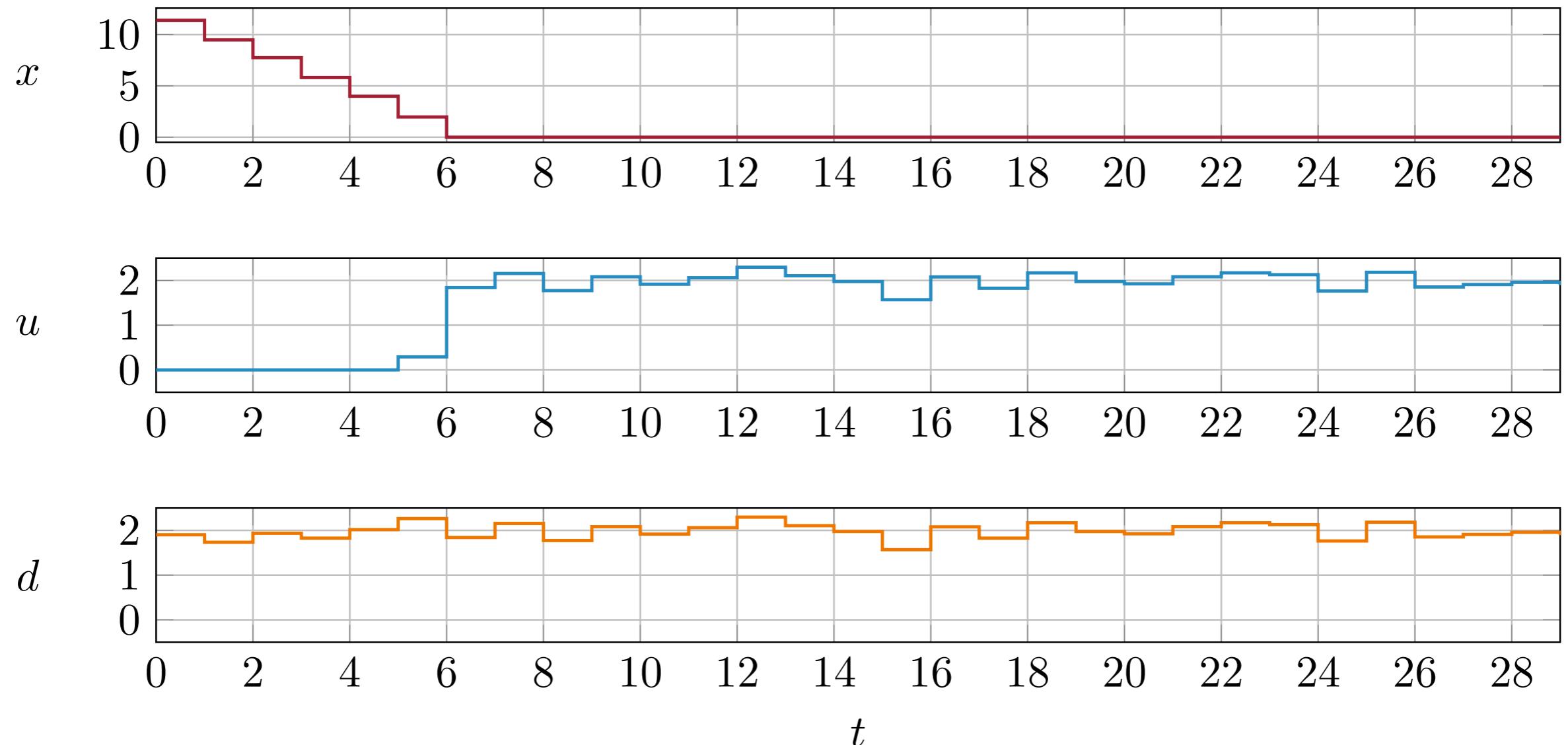


Parameters

How do x and u depend on the parameters?



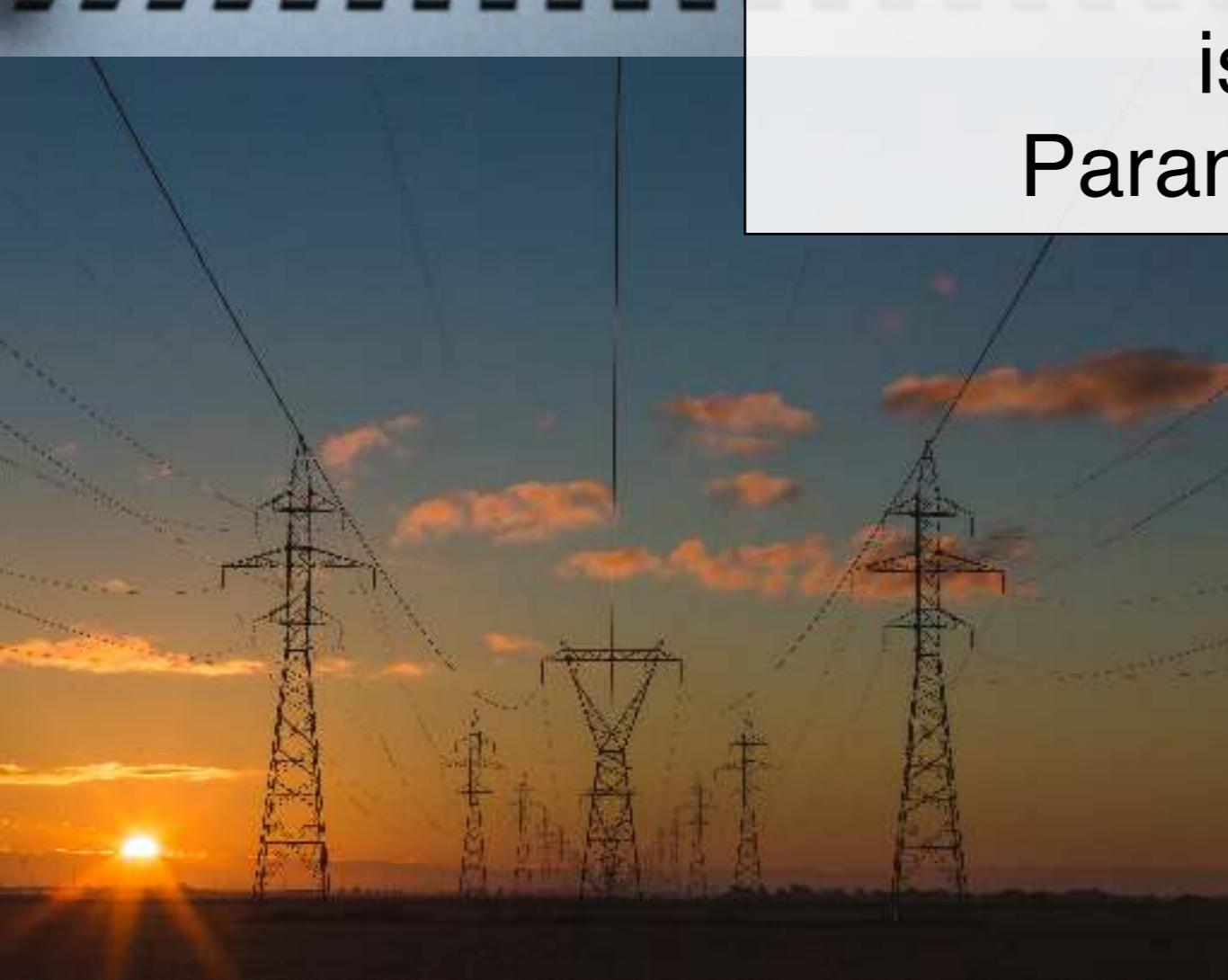
How do x and u depend on the parameters?



x_{init} ? d_t ?

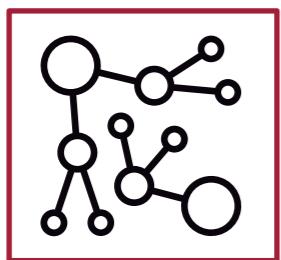


Real-world Optimization
is
Parametric

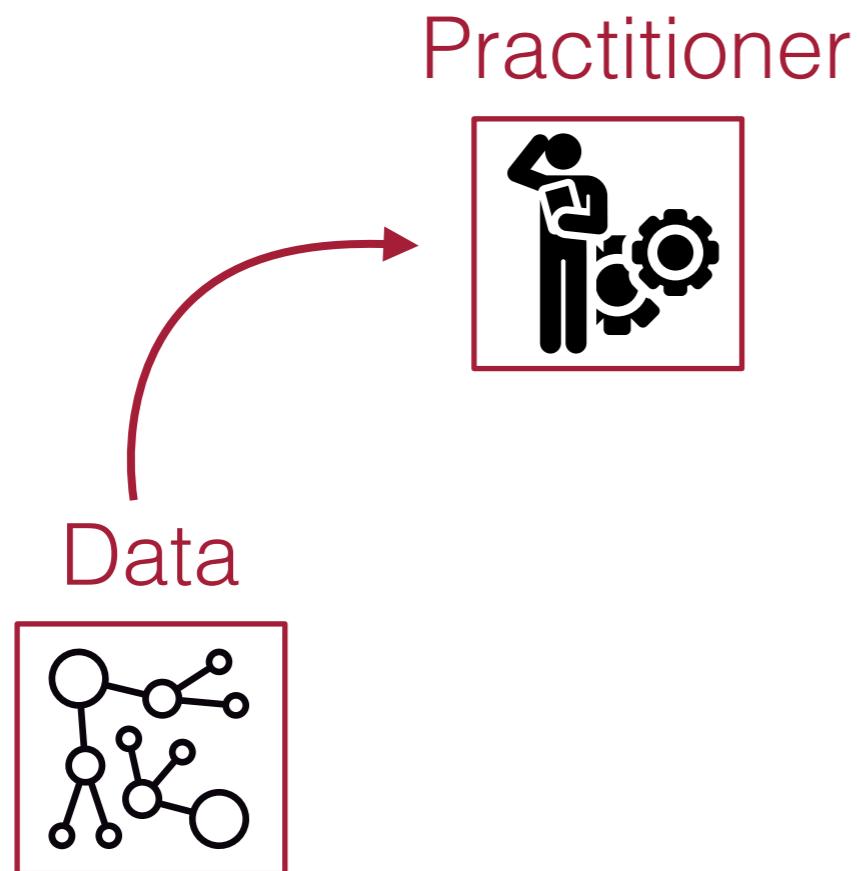


The decision-making workflow

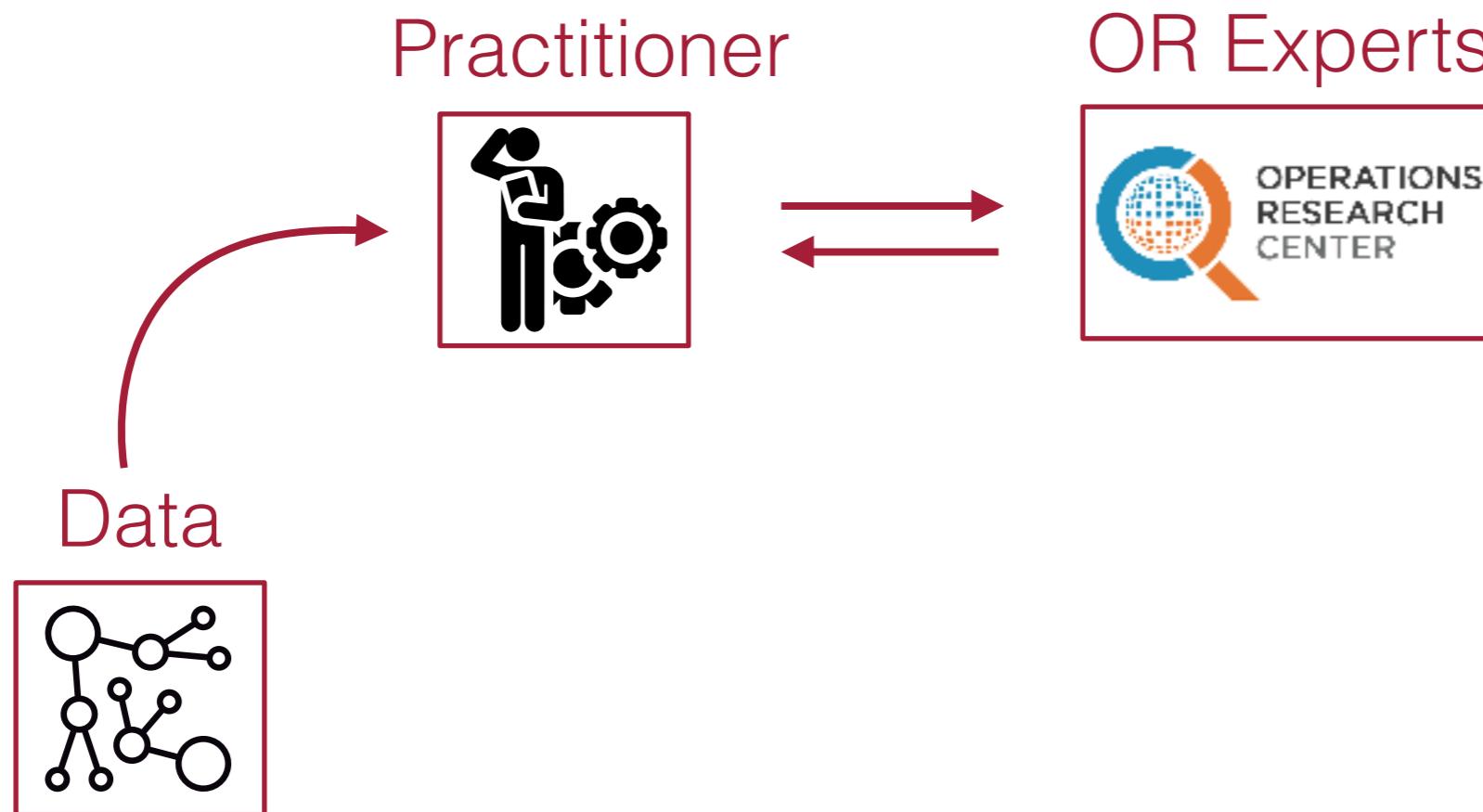
Data



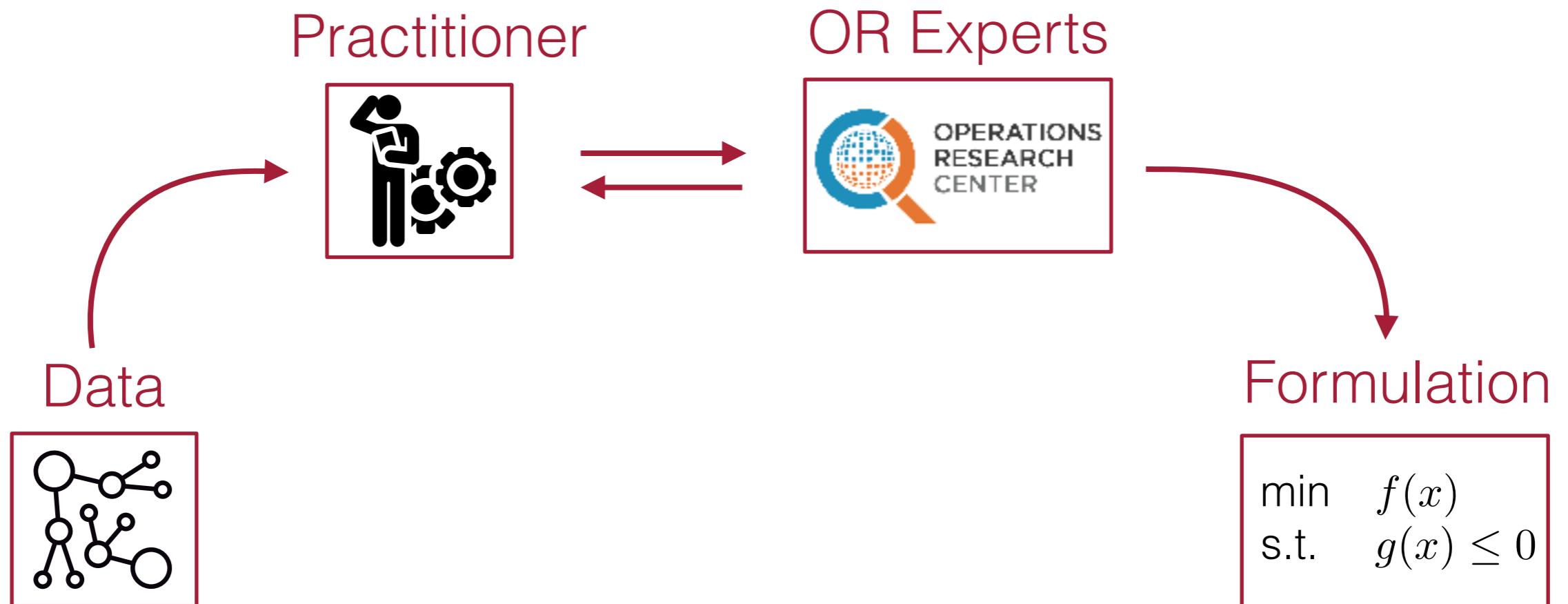
The decision-making workflow



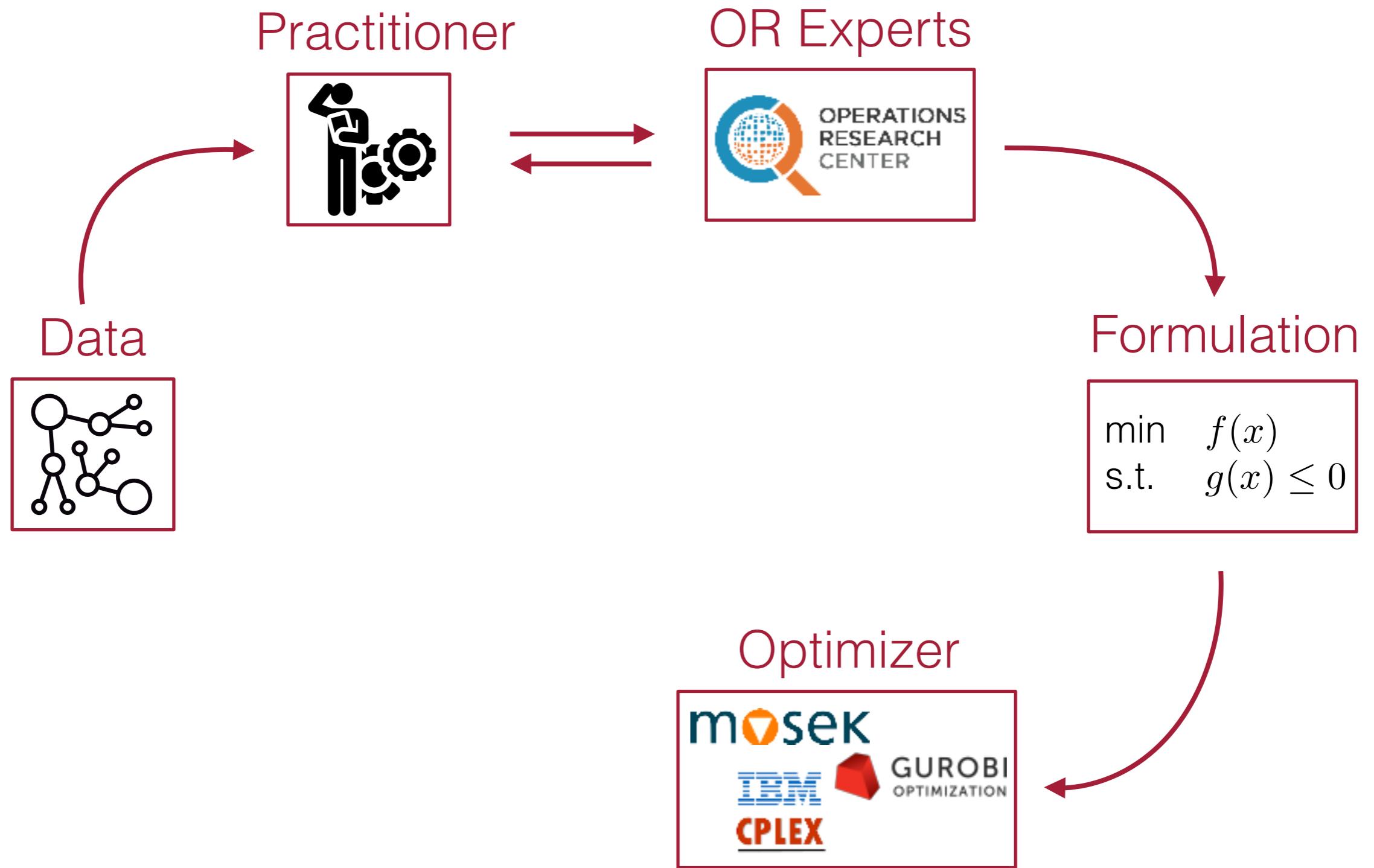
The decision-making workflow



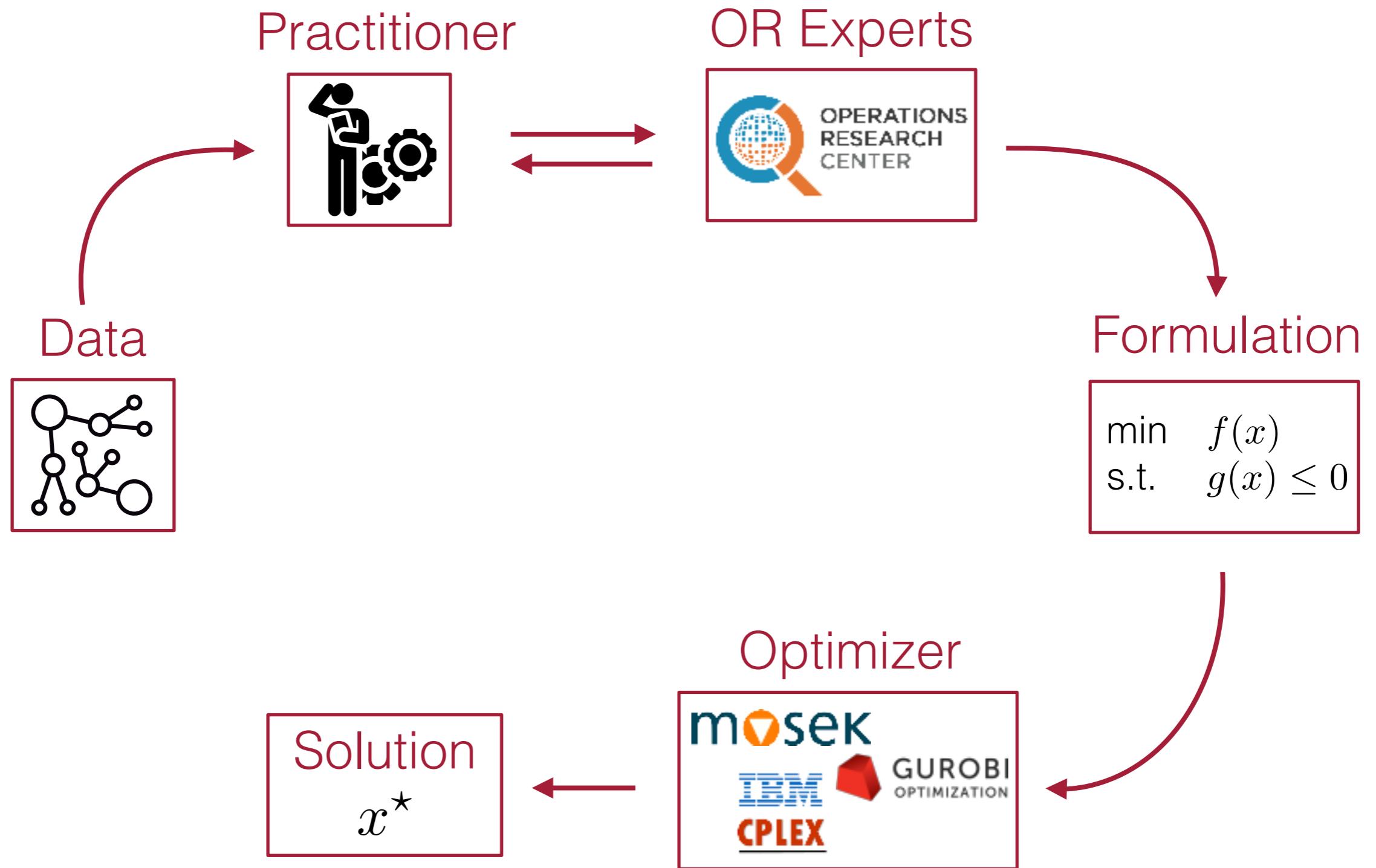
The decision-making workflow



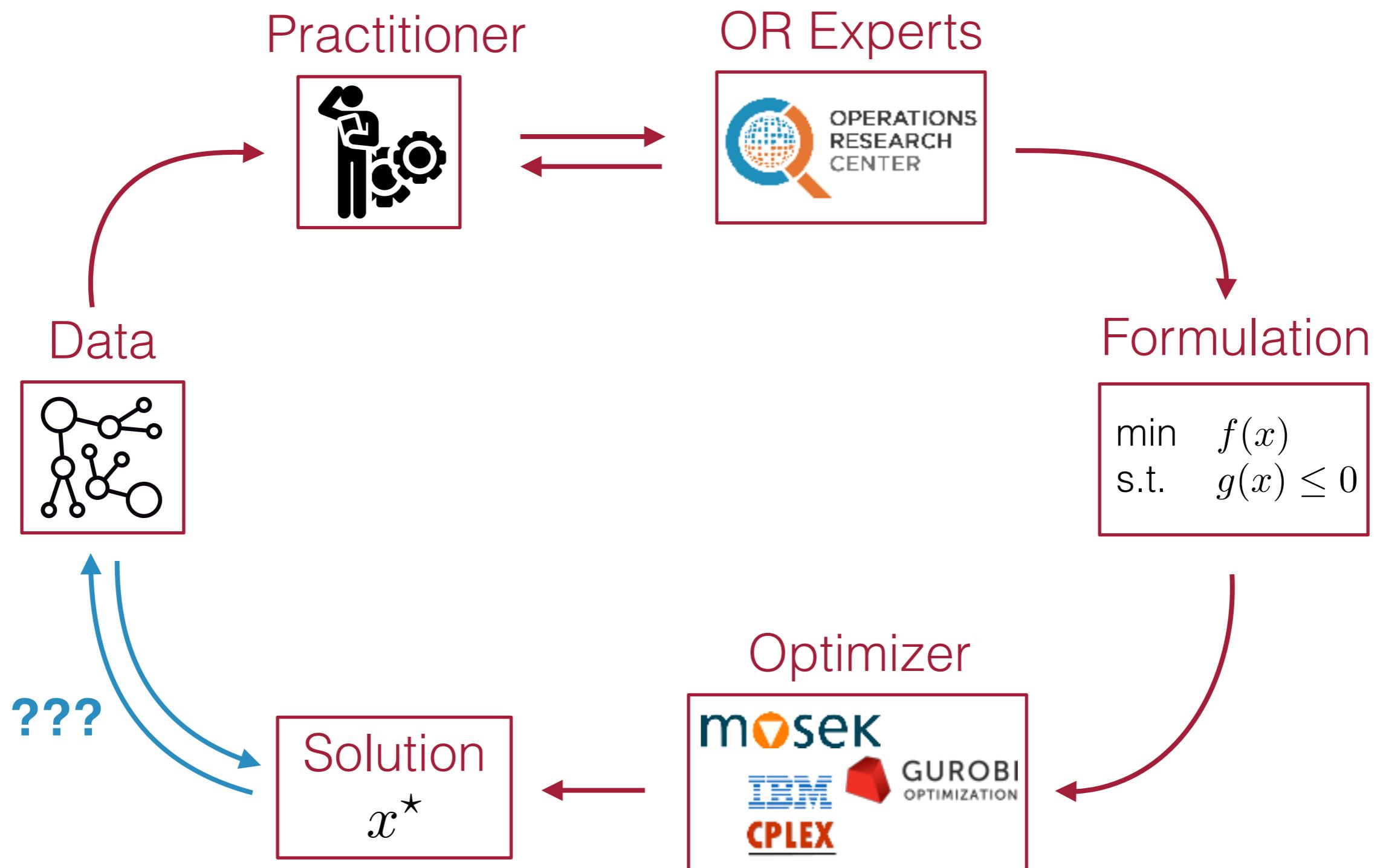
The decision-making workflow



The decision-making workflow



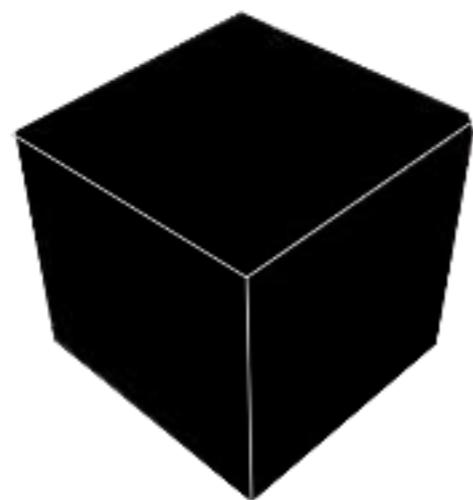
The decision-making workflow



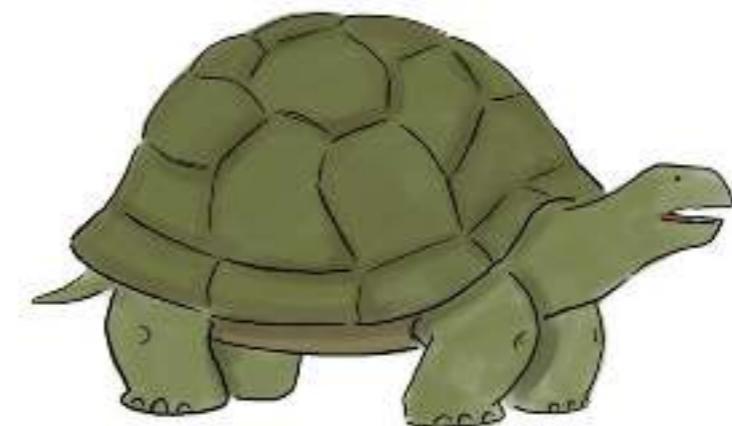
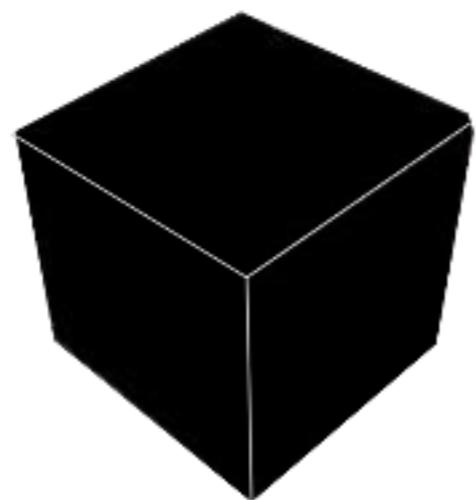
Traditional optimization



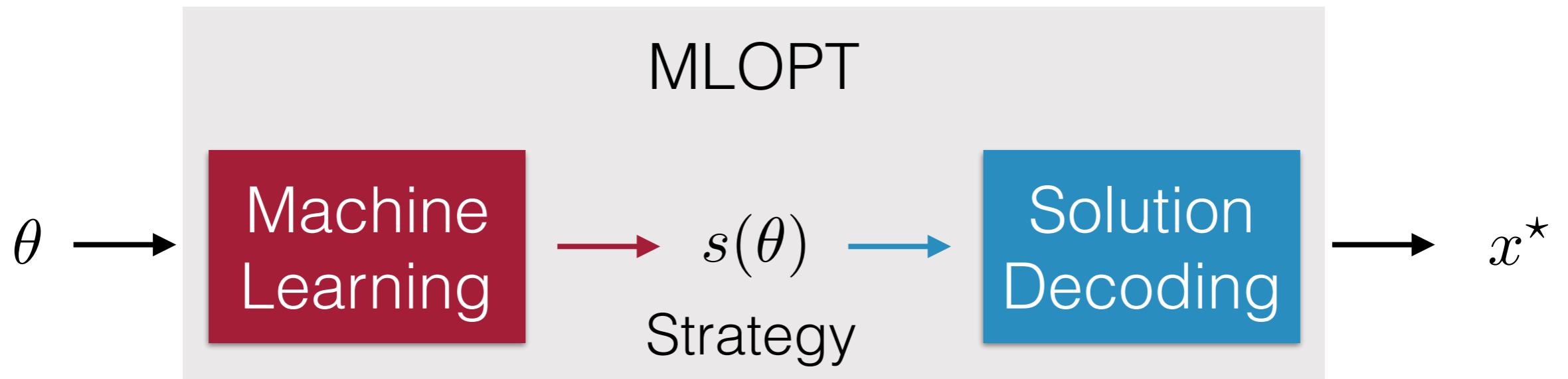
Traditional optimization



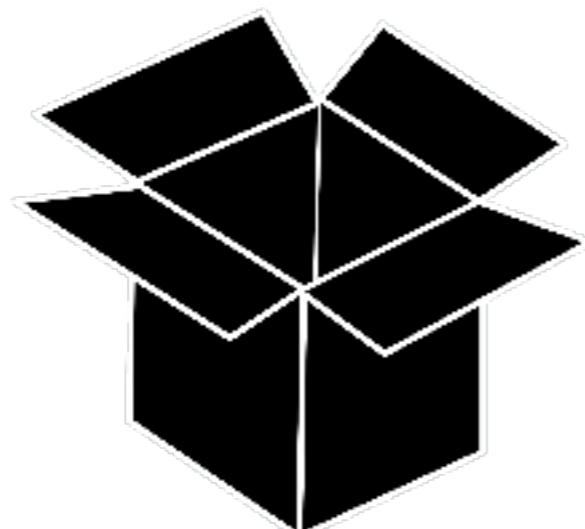
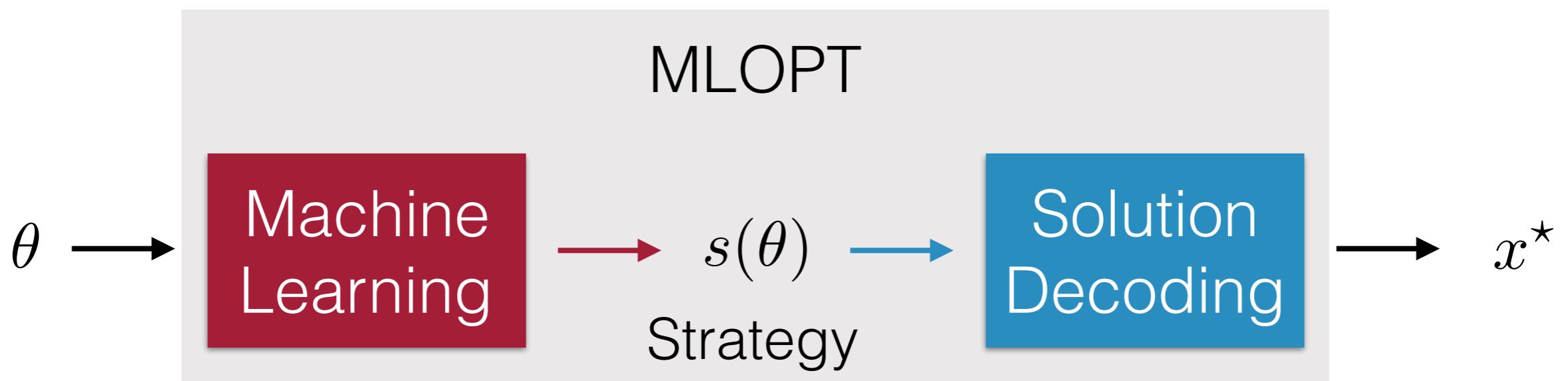
Traditional optimization



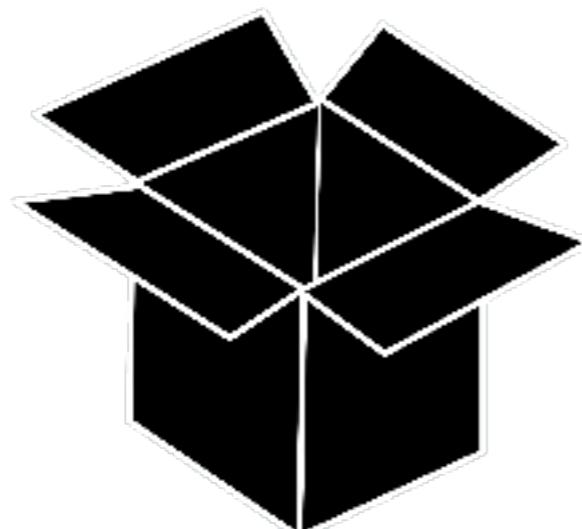
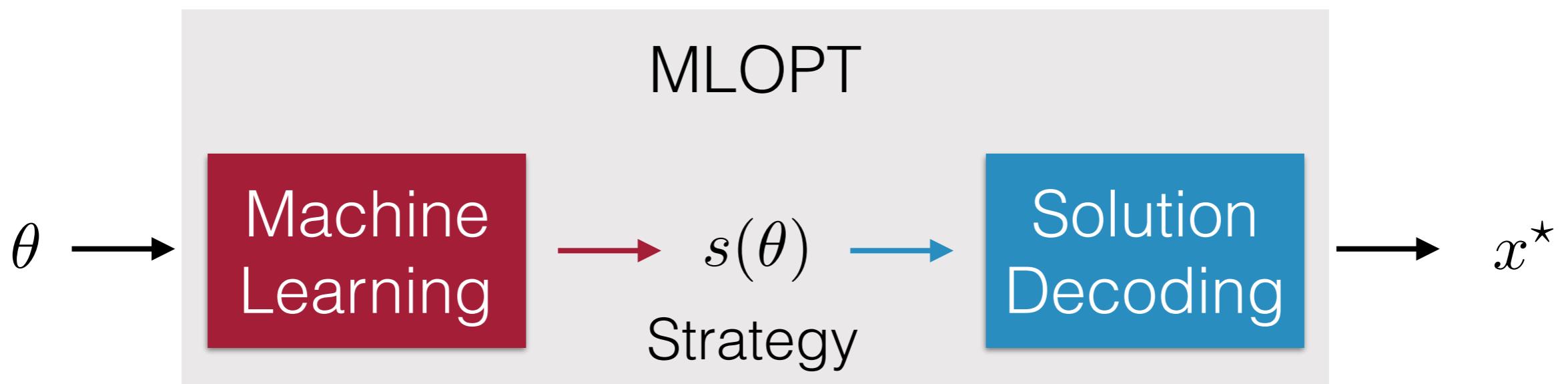
Machine Learning Optimizer



Machine Learning Optimizer

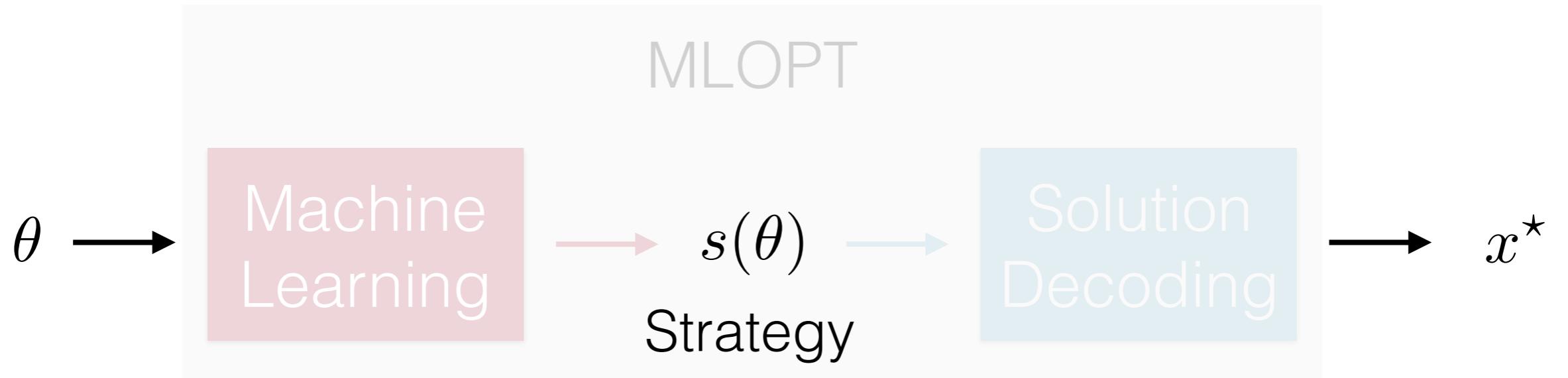


Machine Learning Optimizer



Optimal strategies

Strategy

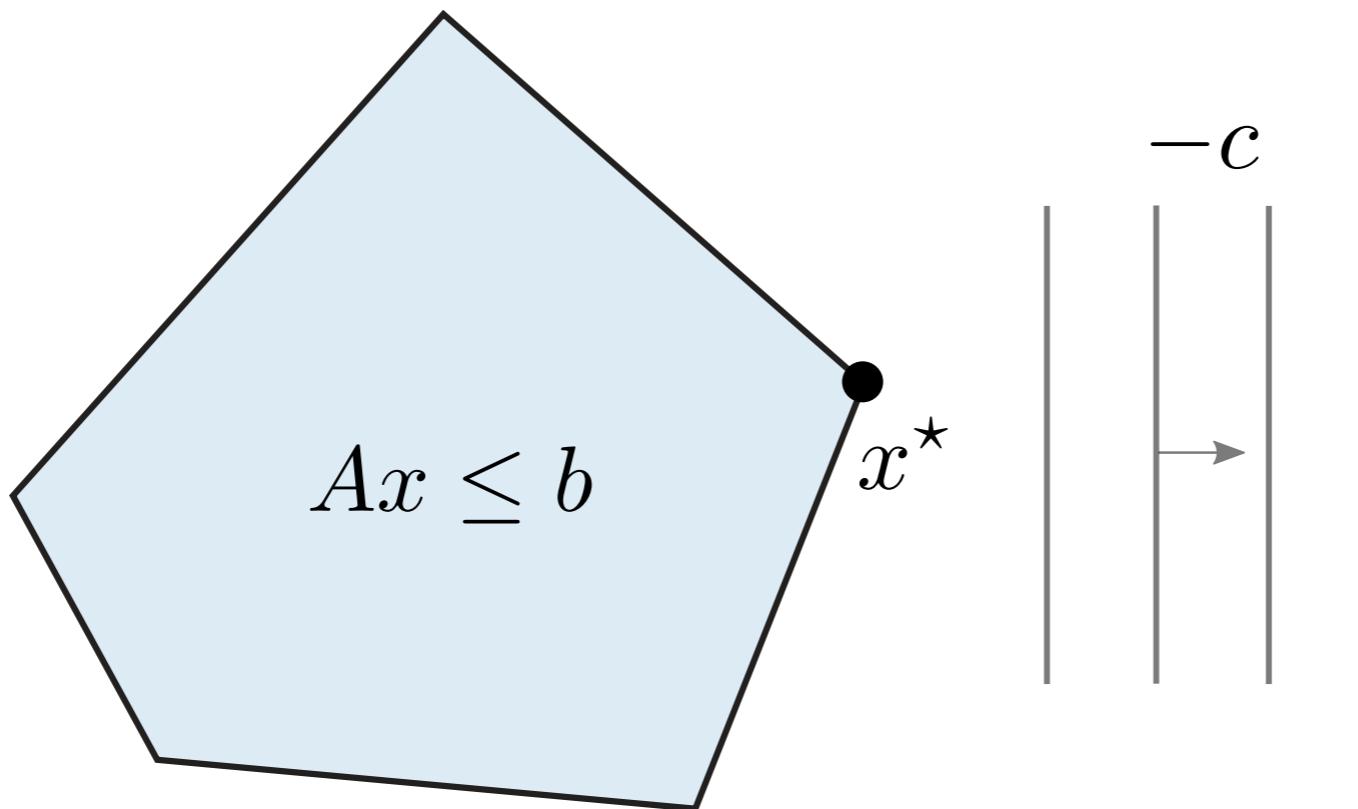


*The complete information to efficiently get
the optimal solution*

$$s(\theta)$$

Linear optimization

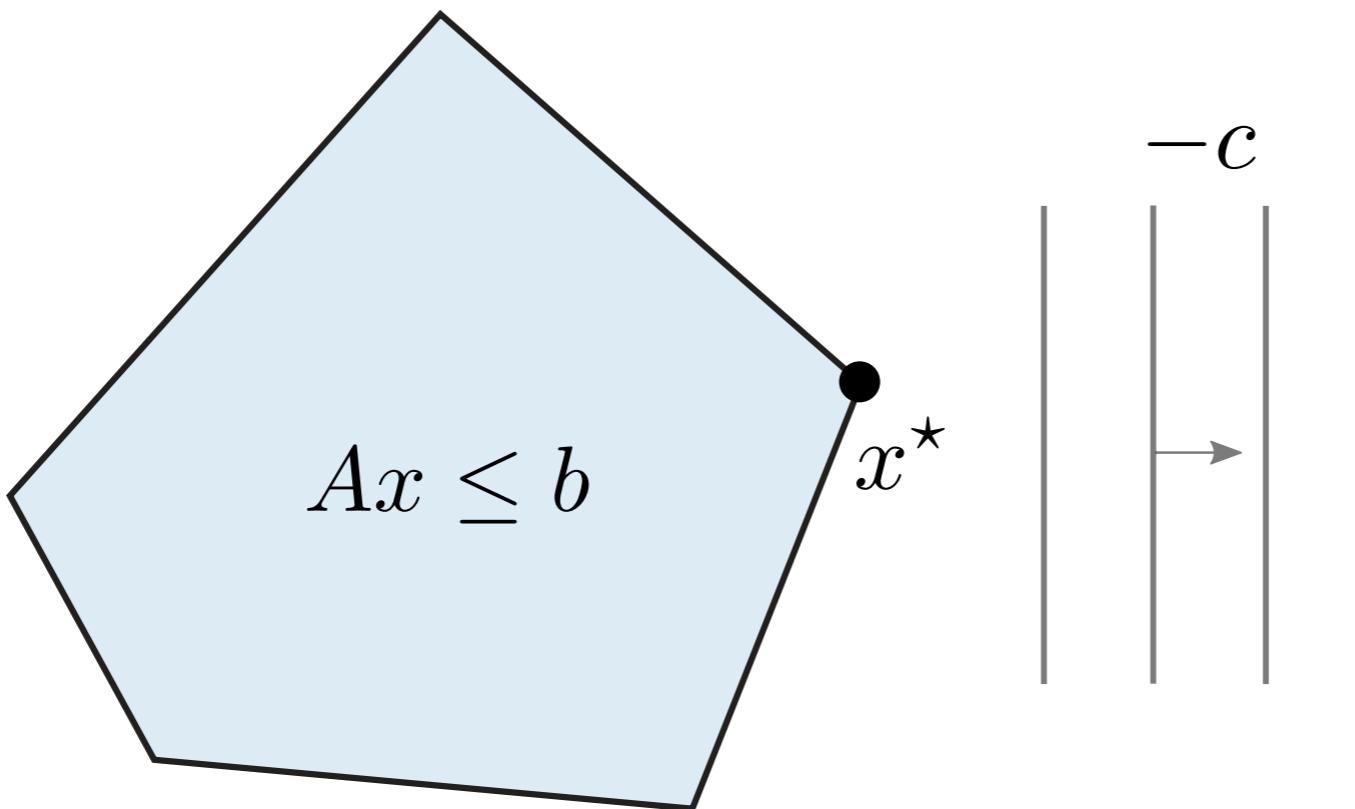
$$\begin{array}{ll}\text{minimize} & c(\theta)^T x \\ \text{subject to} & A(\theta)x \leq b(\theta)\end{array}$$



Linear optimization

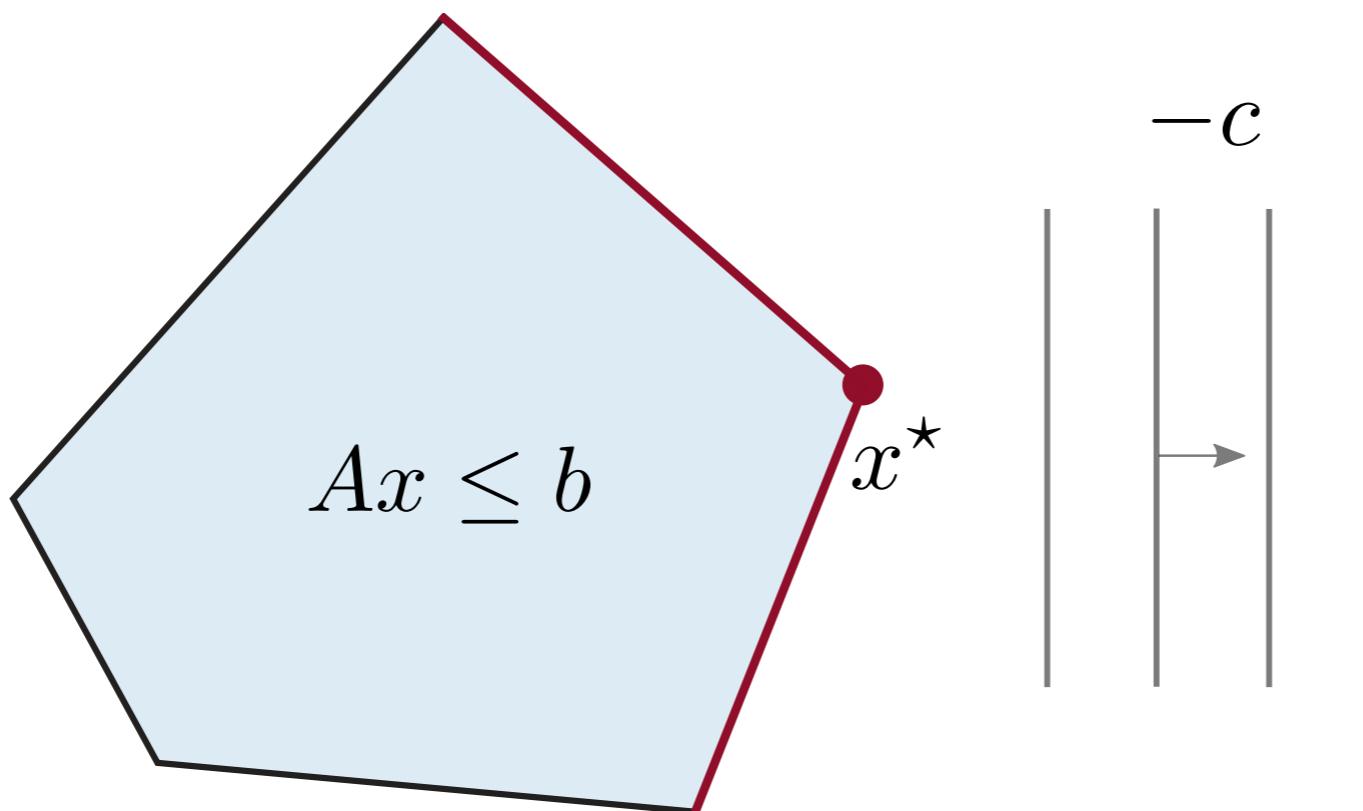
$$\begin{array}{ll}\text{minimize} & c(\theta)^T x \\ \text{subject to} & A(\theta)x \leq b(\theta)\end{array}$$

Parameters



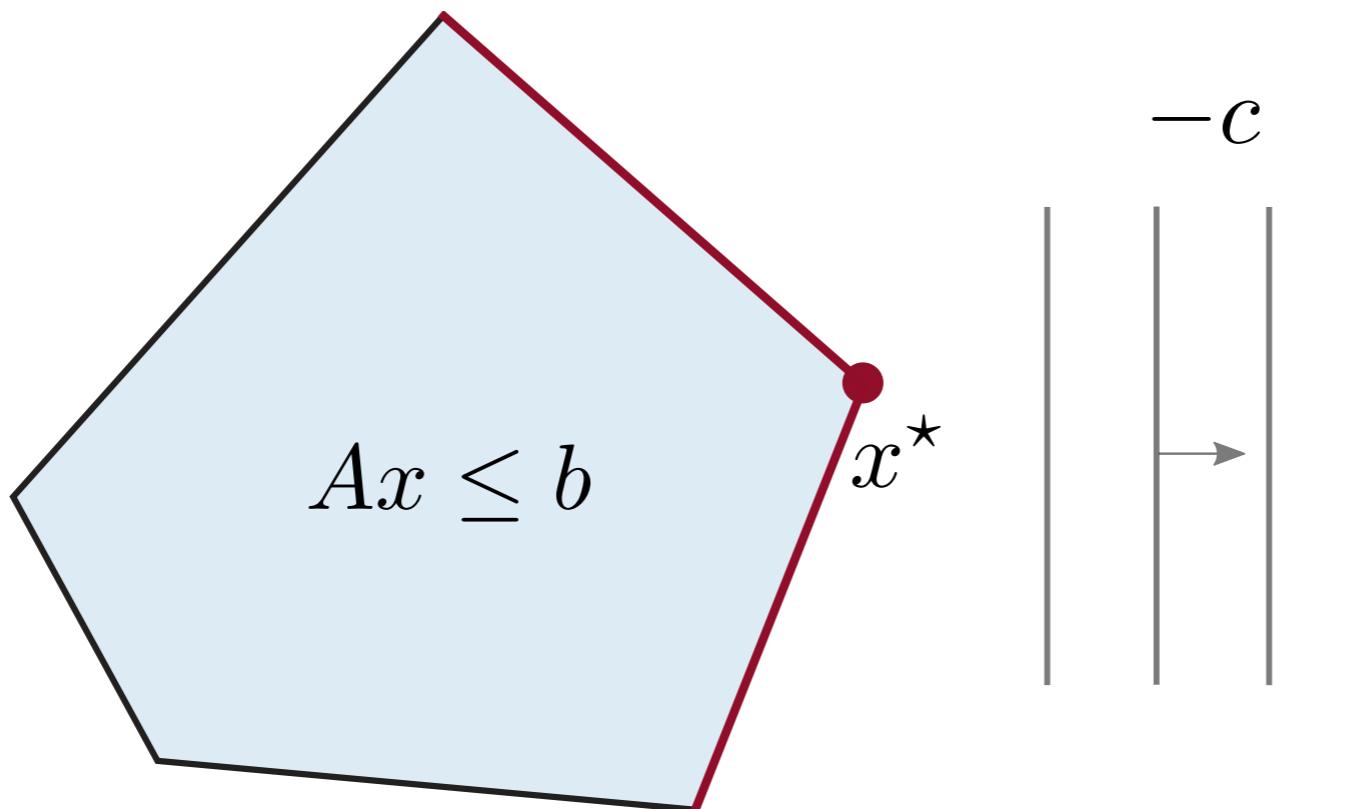
Tight constraints

$$\mathcal{T}(\theta) = \{i \mid A_i(\theta)x^* = b_i(\theta)\}$$



Tight constraints

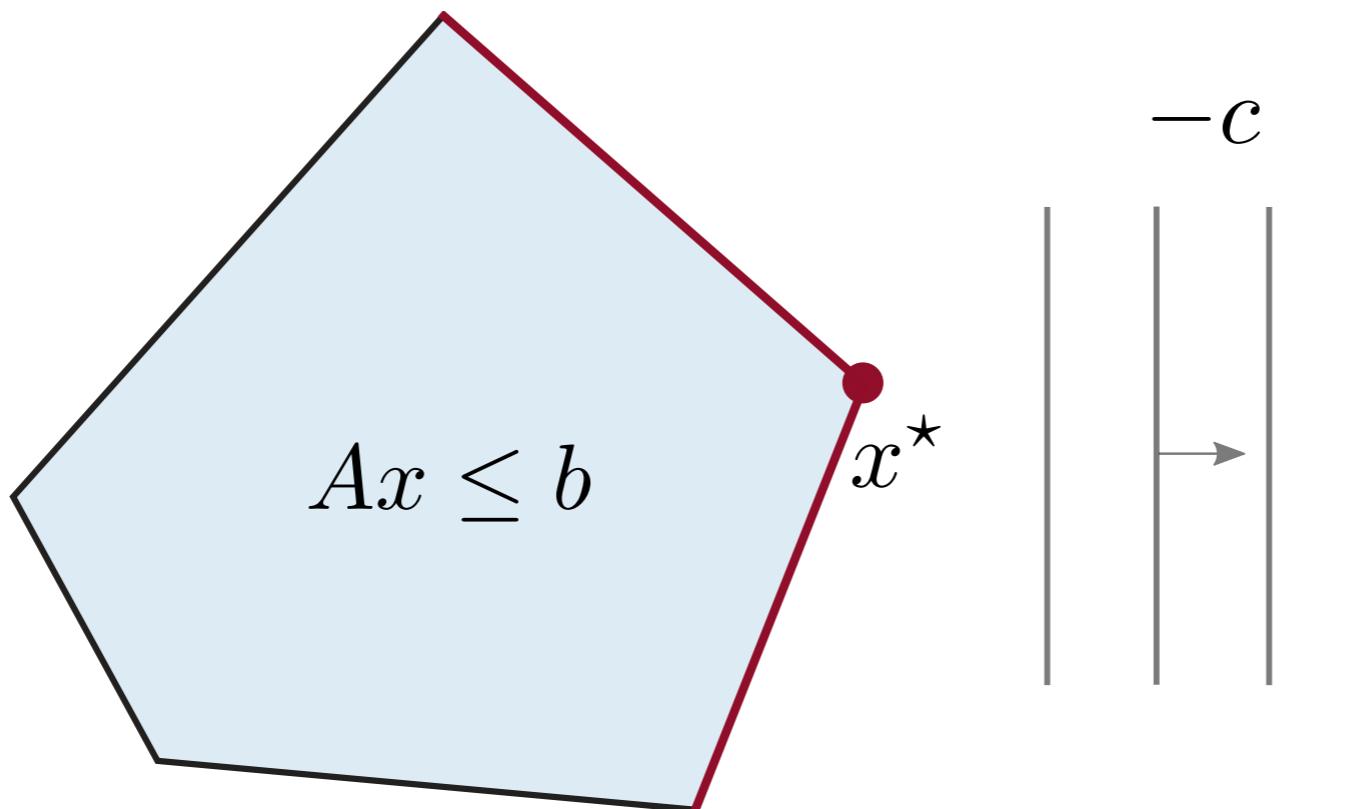
$$\mathcal{T}(\theta) = \{i \mid A_i(\theta)x^* = b_i(\theta)\}$$



$|\mathcal{T}(\theta)| = \# \text{ variables}$ if non-degenerate

Tight constraints

$$\mathcal{T}(\theta) = \{i \mid A_i(\theta)x^* = b_i(\theta)\}$$

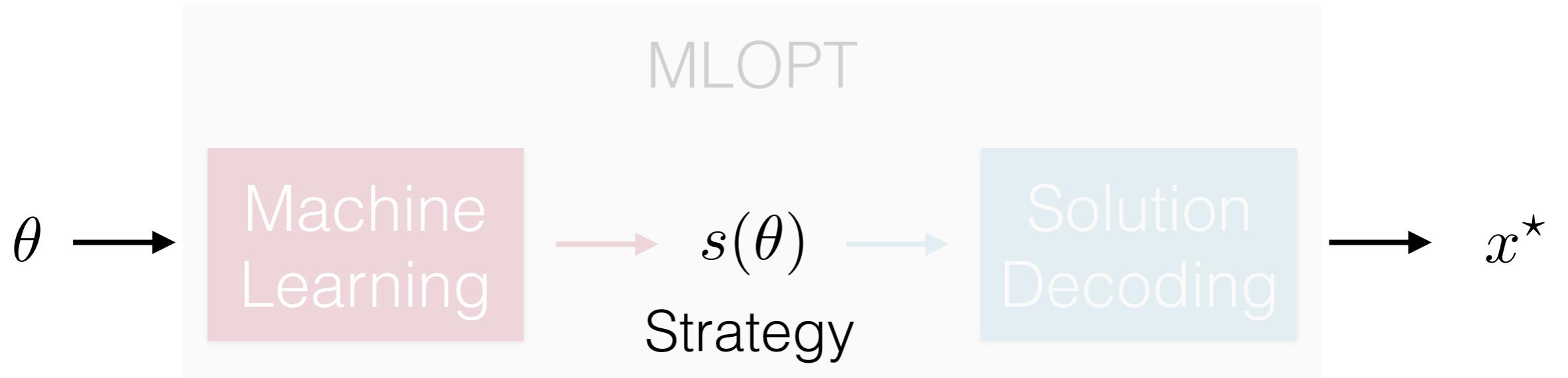


$|\mathcal{T}(\theta)| = \# \text{ variables}$

if non-degenerate
in general

$|\mathcal{T}(\theta)| \ll \# \text{ constraints}$

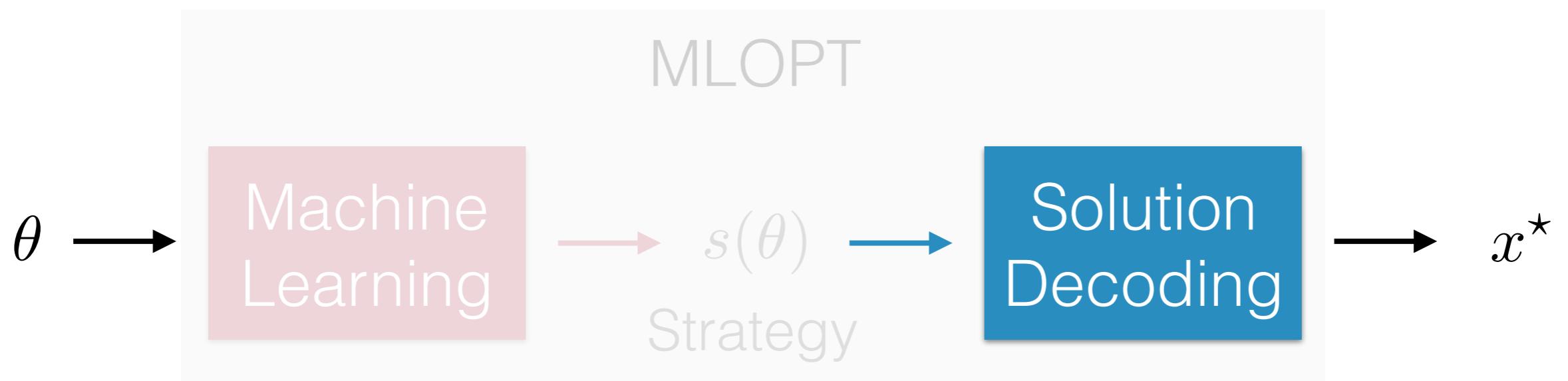
Strategy



*The complete information to efficiently get
the optimal solution*

$$s(\theta) = \mathcal{T}(\theta)$$

Computing the solution from the optimal strategy



Computing the solution from the optimal strategy

$$\begin{array}{ll}\text{minimize} & c(\theta)^T x \\ \text{subject to} & A(\theta)x \leq b(\theta)\end{array}$$

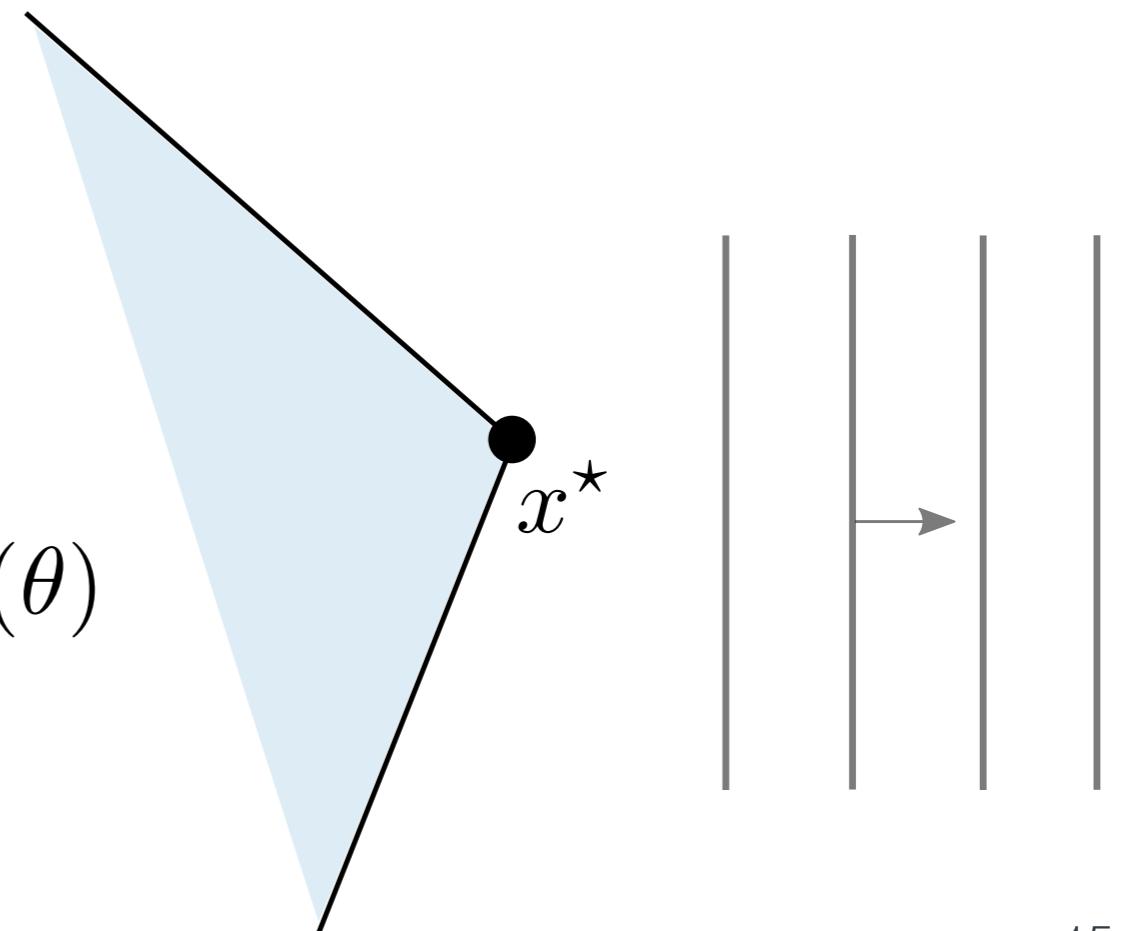
Computing the solution from the optimal strategy

$$\begin{array}{ll}\text{minimize} & c(\theta)^T x \\ \text{subject to} & A(\theta)x \leq b(\theta)\end{array}$$



Strategy $s(\theta) = \mathcal{T}(\theta)$

$$\begin{array}{ll}\text{minimize} & c(\theta)^T x \\ \text{subject to} & A_i(\theta)x = b_i(\theta) \quad \forall i \in \mathcal{T}(\theta)\end{array}$$





Inventory management

minimize $\sum_{t=0}^{T-1} h(x_t) + o(u_t)$

subject to $x_{t+1} = x_t + u_t - d_t$
 $x_0 = x_{\text{init}}$
 $0 \leq u_t \leq M$



Inventory management

minimize $\sum_{t=0}^{T-1} h(x_t) + o(u_t)$

subject to $x_{t+1} = x_t + u_t - d_t$

$x_0 = x_{\text{init}}$

$0 \leq u_t \leq M$

Inventory



Inventory management

minimize $\sum_{t=0}^{T-1} h(x_t) + o(u_t)$

subject to $x_{t+1} = x_t + u_t - d_t$

$x_0 = x_{\text{init}}$

$0 \leq u_t \leq M$

Inventory 

Order 



Inventory management

minimize $\sum_{t=0}^{T-1} h(x_t) + o(u_t)$

subject to $x_{t+1} = x_t + u_t - d_t$

Inventory

$$x_0 = x_{\text{init}}$$
$$0 \leq u_t \leq M$$

Order

Demand

A photograph showing a close-up view of a metal shelving unit in a warehouse. The shelves are filled with numerous cardboard boxes, each with a white shipping label. The boxes are stacked in several layers across the visible height of the shelves.

Inventory management

minimize $\sum_{t=0}^{T-1} h(x_t) + o(u_t)$

subject to $x_{t+1} = x_t + u_t - d_t$

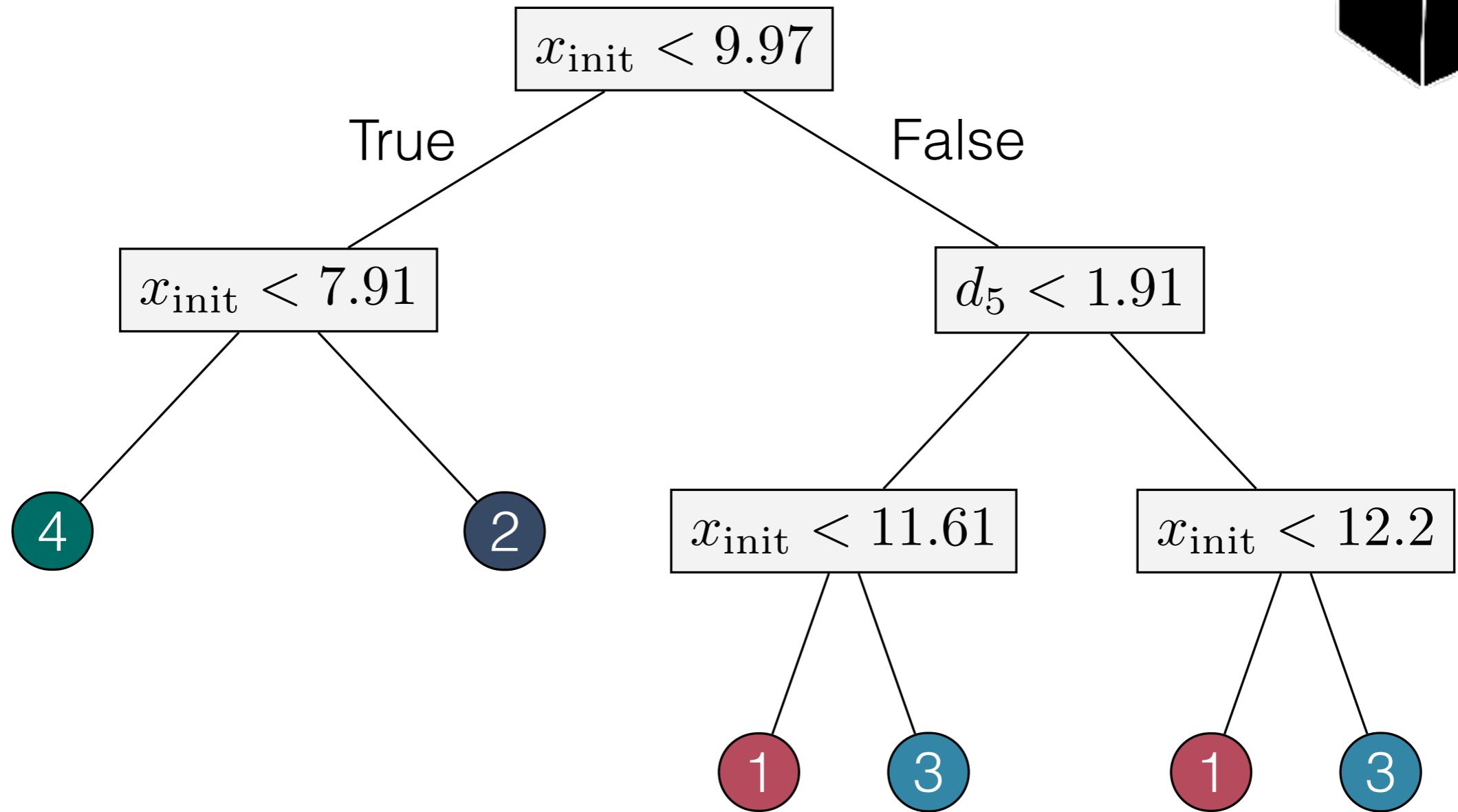
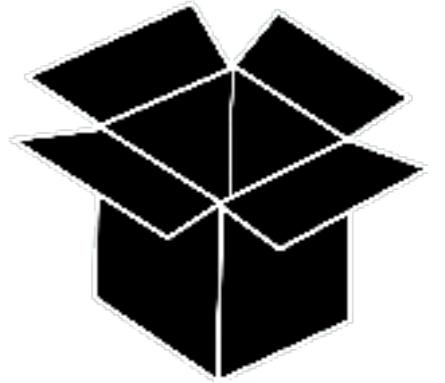
$x_0 = x_{\text{init}}$

$0 \leq u_t \leq M$

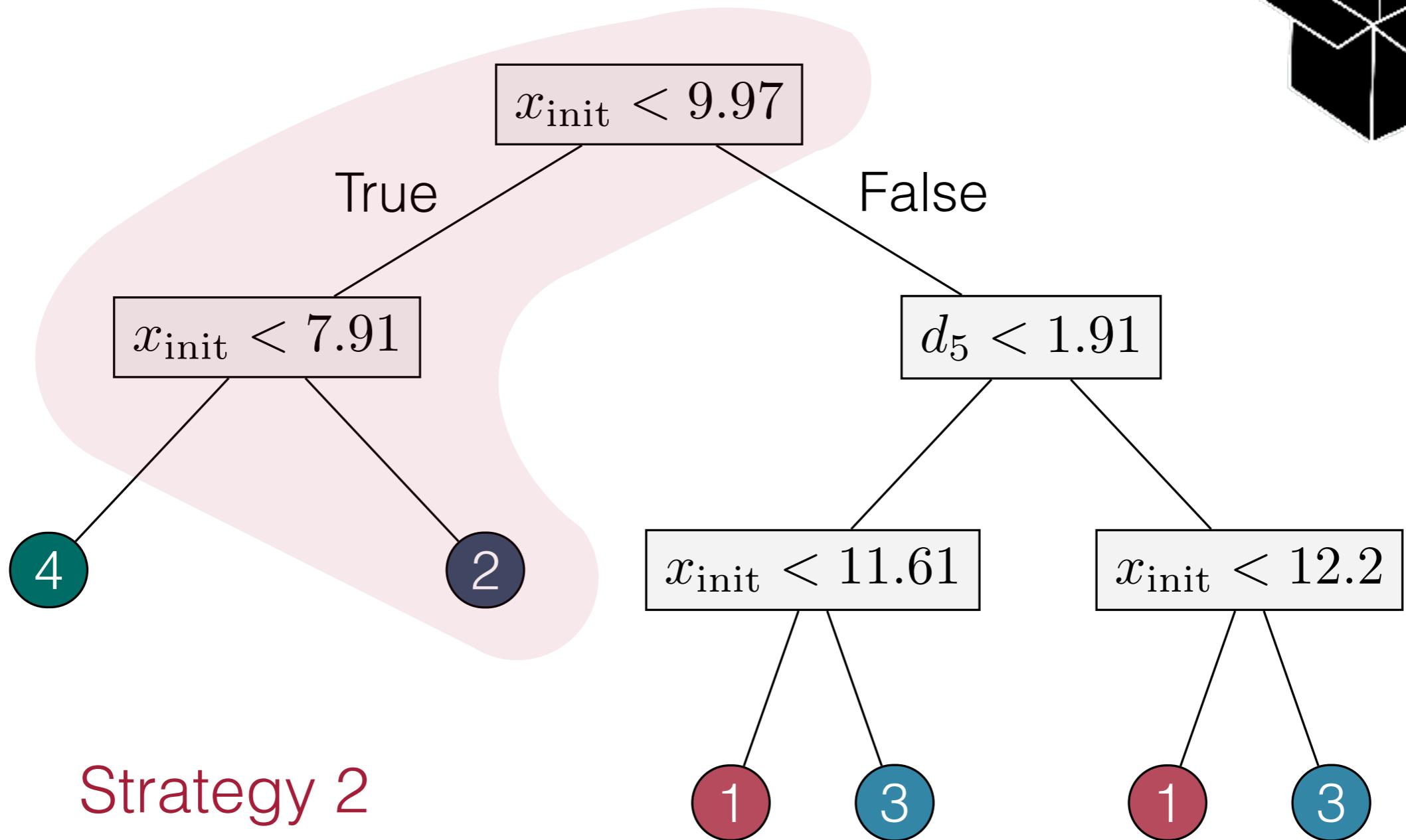
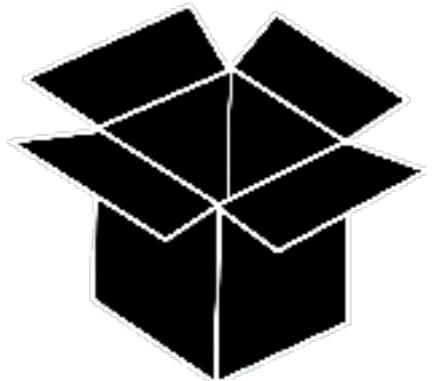
Parameters

A diagram illustrating the components of the inventory management model. A vertical teal arrow points upwards from the word "Parameters" to the initial condition $x_0 = x_{\text{init}}$. A diagonal teal arrow points upwards and to the right from the word "Parameters" to the demand term d_t in the state transition equation.

Strategy selection



Strategy selection



Strategy 2

$$u_t = 0 \quad t \leq 4$$

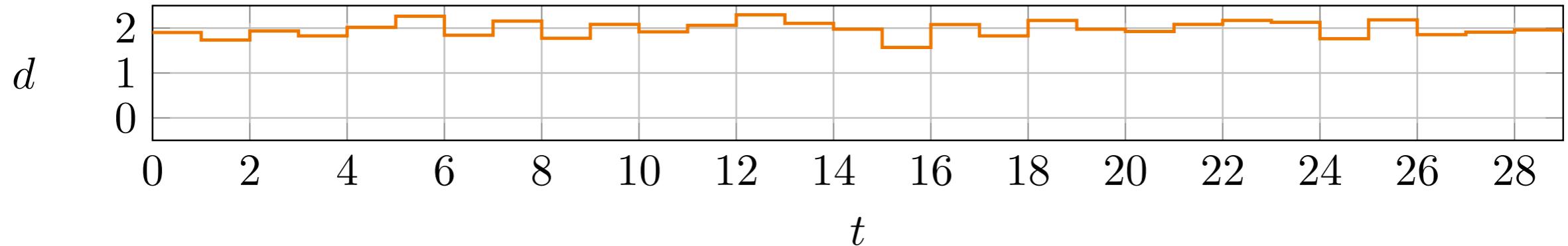
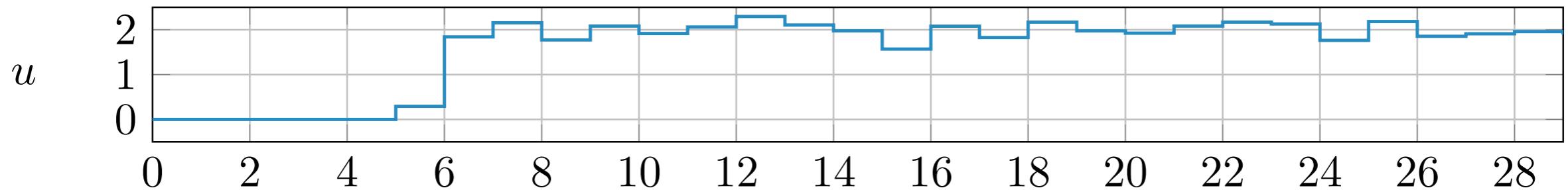
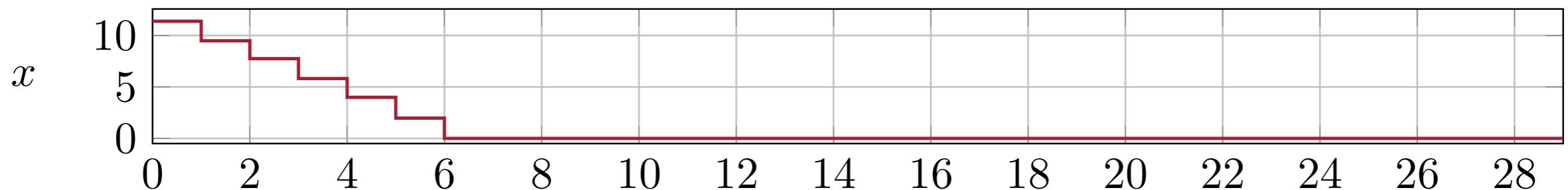
$$0 \leq u_t \leq M \quad t > 4$$

Strategies for inventory

Strategy 2

$$u_t = 0 \quad t \leq 4$$

$$0 \leq u_t \leq M \quad t > 4$$



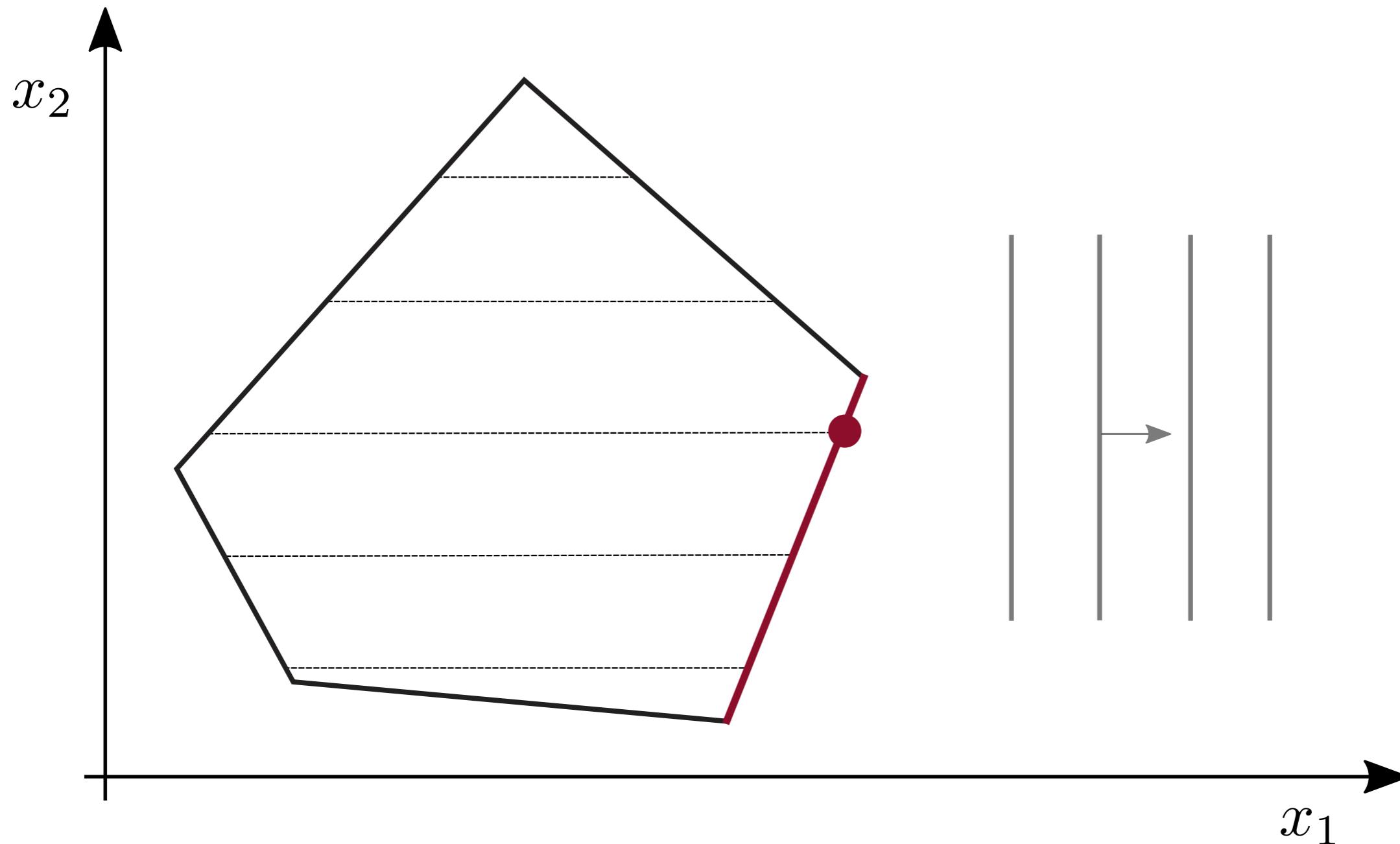
Mixed-integer optimization

$$\begin{array}{ll}\text{minimize} & c(\theta)^T x \\ \text{subject to} & A(\theta)x \leq b(\theta) \\ & x_{\mathcal{I}} \in \mathbf{Z}^d\end{array}$$

Mixed-integer optimization

$$\begin{array}{ll}\text{minimize} & c(\theta)^T x \\ \text{subject to} & A(\theta)x \leq b(\theta) \\ & x_{\mathcal{I}} \in \mathbf{Z}^d \quad \text{Integers}\end{array}$$

Tight constraints are not enough

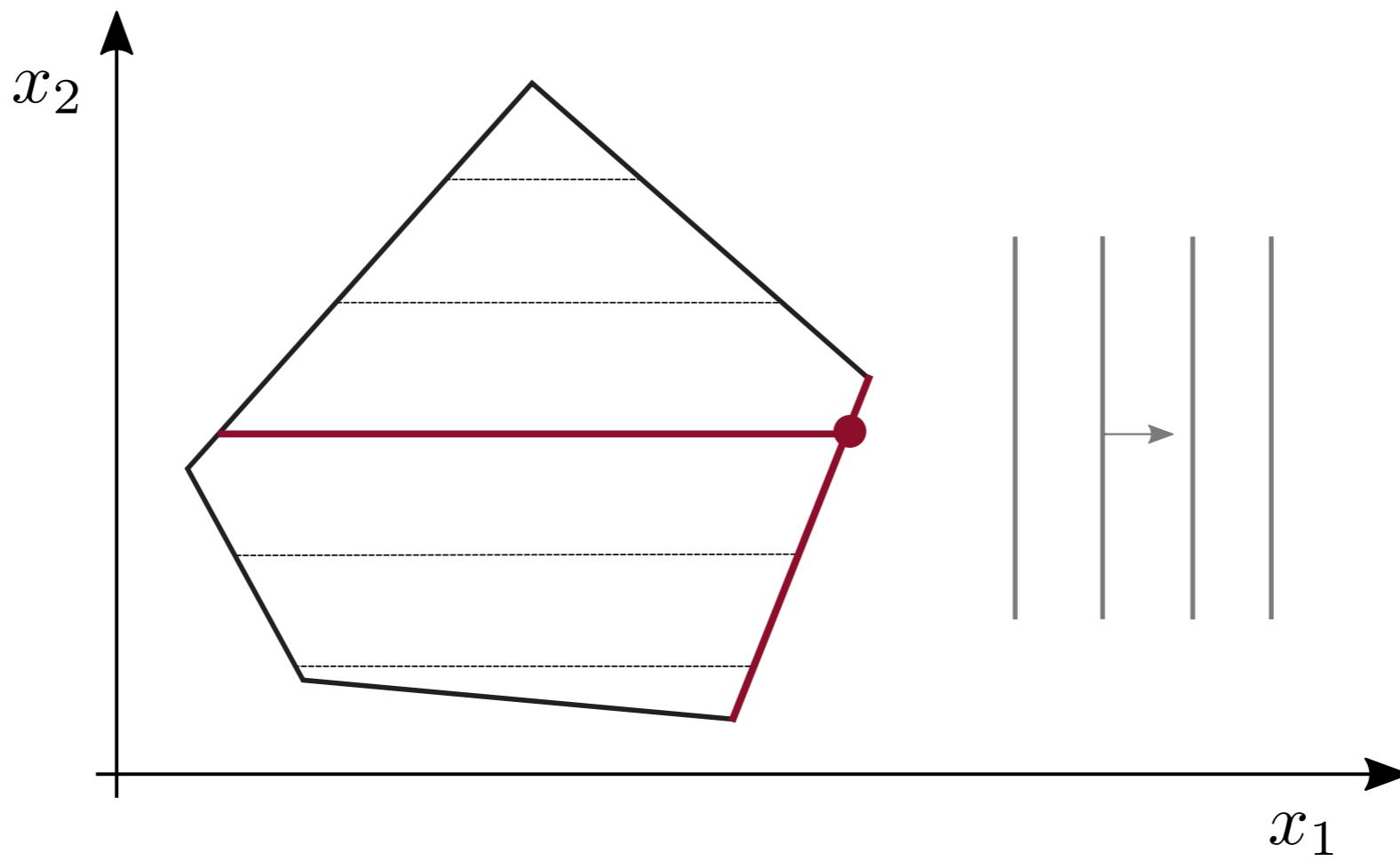


Strategy

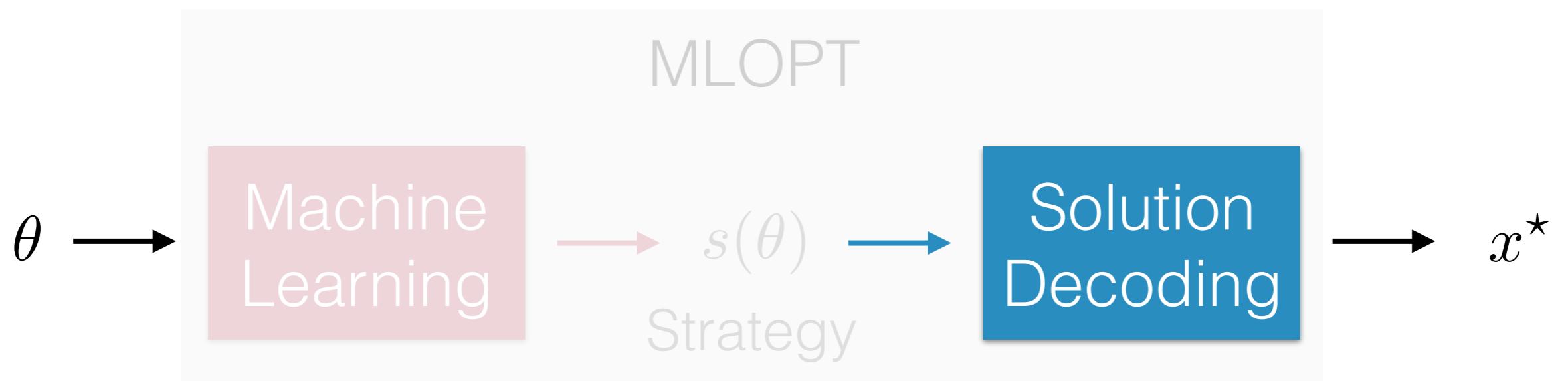
$$s(\theta) = (\mathcal{T}(\theta), x_{\mathcal{I}}^*(\theta))$$



Integer variables



Computing the solution from the optimal strategy



Computing the solution from the optimal strategy

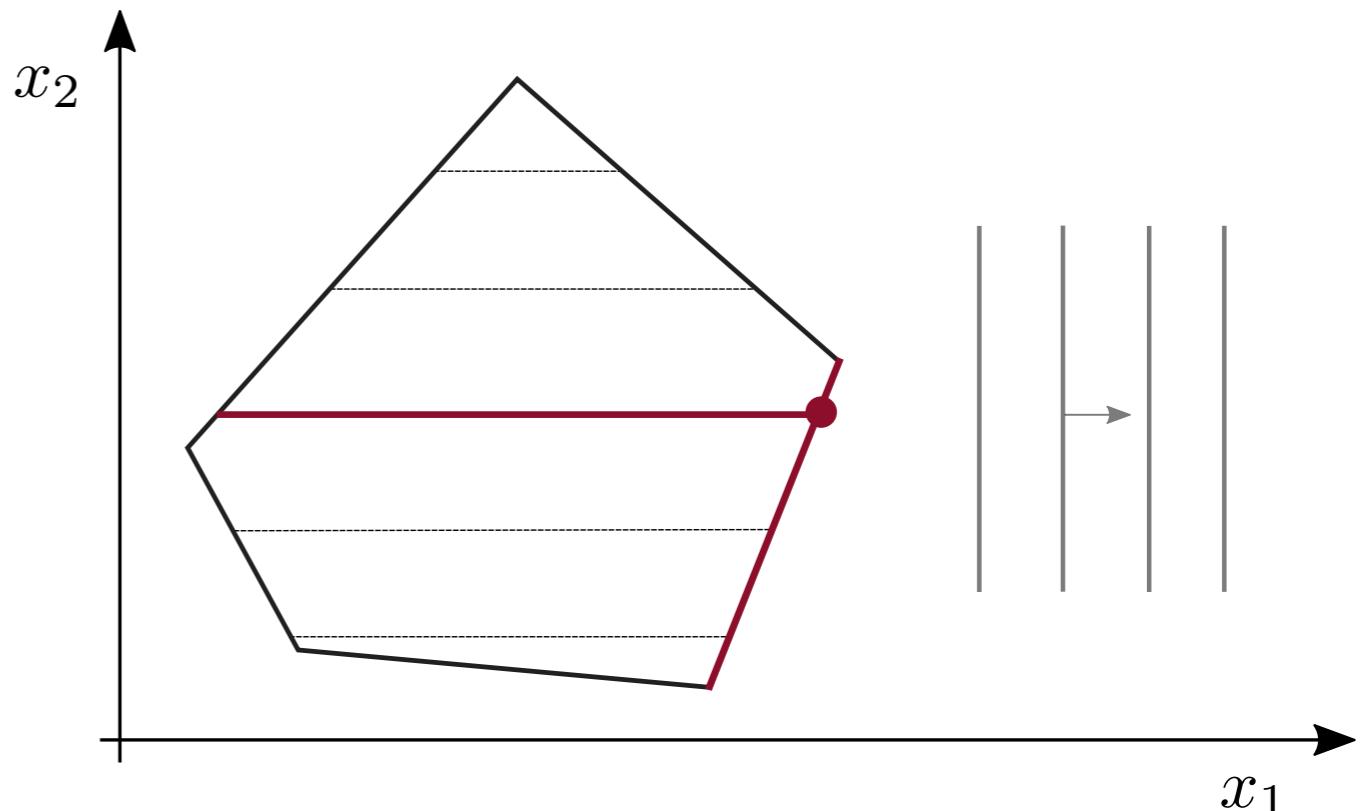
$$\begin{array}{ll}\text{minimize} & c(\theta)^T x \\ \text{subject to} & A(\theta)x \leq b(\theta) \\ & x_{\mathcal{I}} \in \mathbf{Z}^d\end{array}$$

Computing the solution from the optimal strategy

minimize $c(\theta)^T x$
subject to $A(\theta)x \leq b(\theta)$
 $x_{\mathcal{I}} \in \mathbf{Z}^d$

Strategy

$$s(\theta) = (\mathcal{T}(\theta), x_{\mathcal{I}}^*(\theta))$$

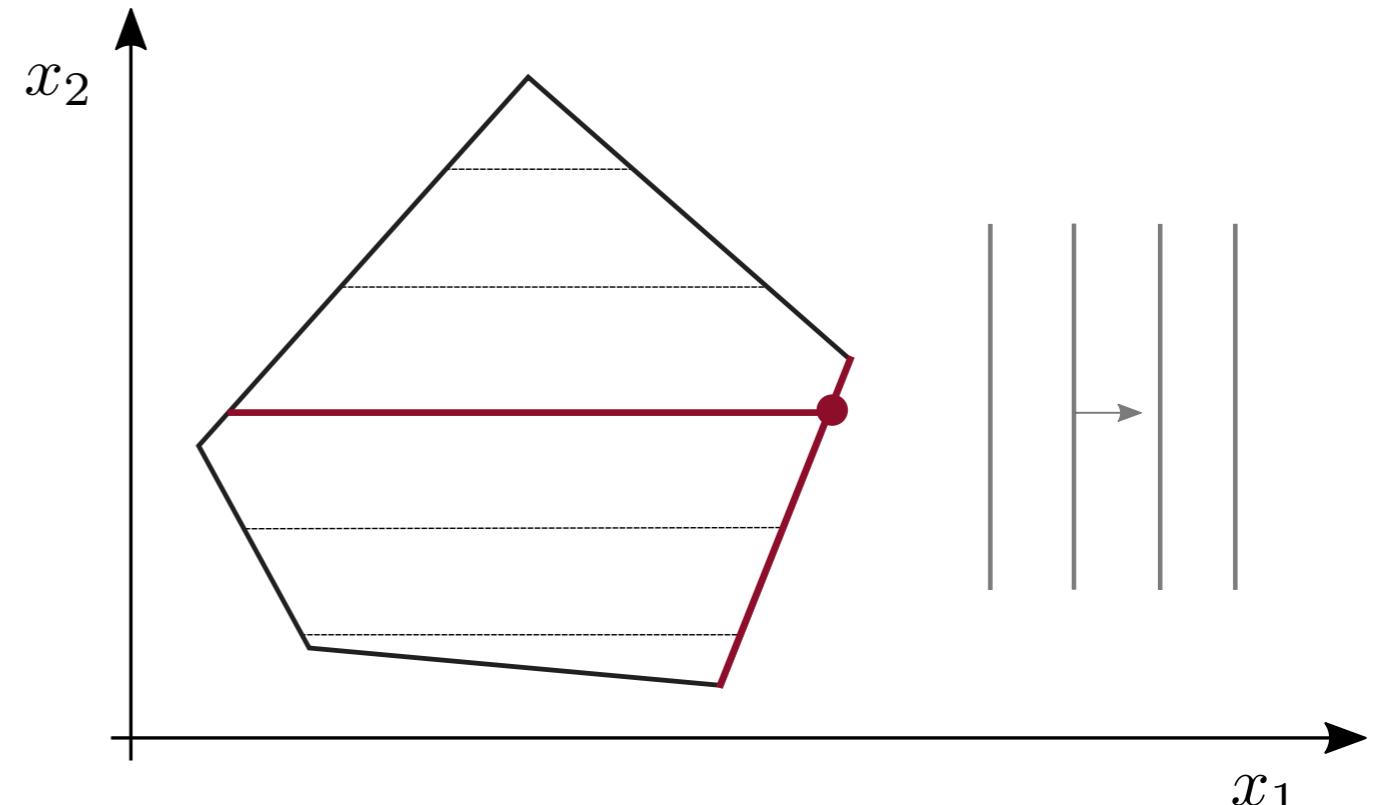


Computing the solution from the optimal strategy

$$\begin{array}{ll}\text{minimize} & c(\theta)^T x \\ \text{subject to} & A(\theta)x \leq b(\theta) \\ & x_{\mathcal{I}} \in \mathbf{Z}^d\end{array}$$

Strategy

$$s(\theta) = (\mathcal{T}(\theta), x_{\mathcal{I}}^*(\theta))$$



$$\begin{array}{ll}\text{minimize} & c(\theta)^T x \\ \text{subject to} & A_i(\theta)x = b_i(\theta) \quad \forall i \in \mathcal{T}(\theta) \\ & x_{\mathcal{I}} = x_{\mathcal{I}}^*(\theta)\end{array}$$

Integers

Knapsack

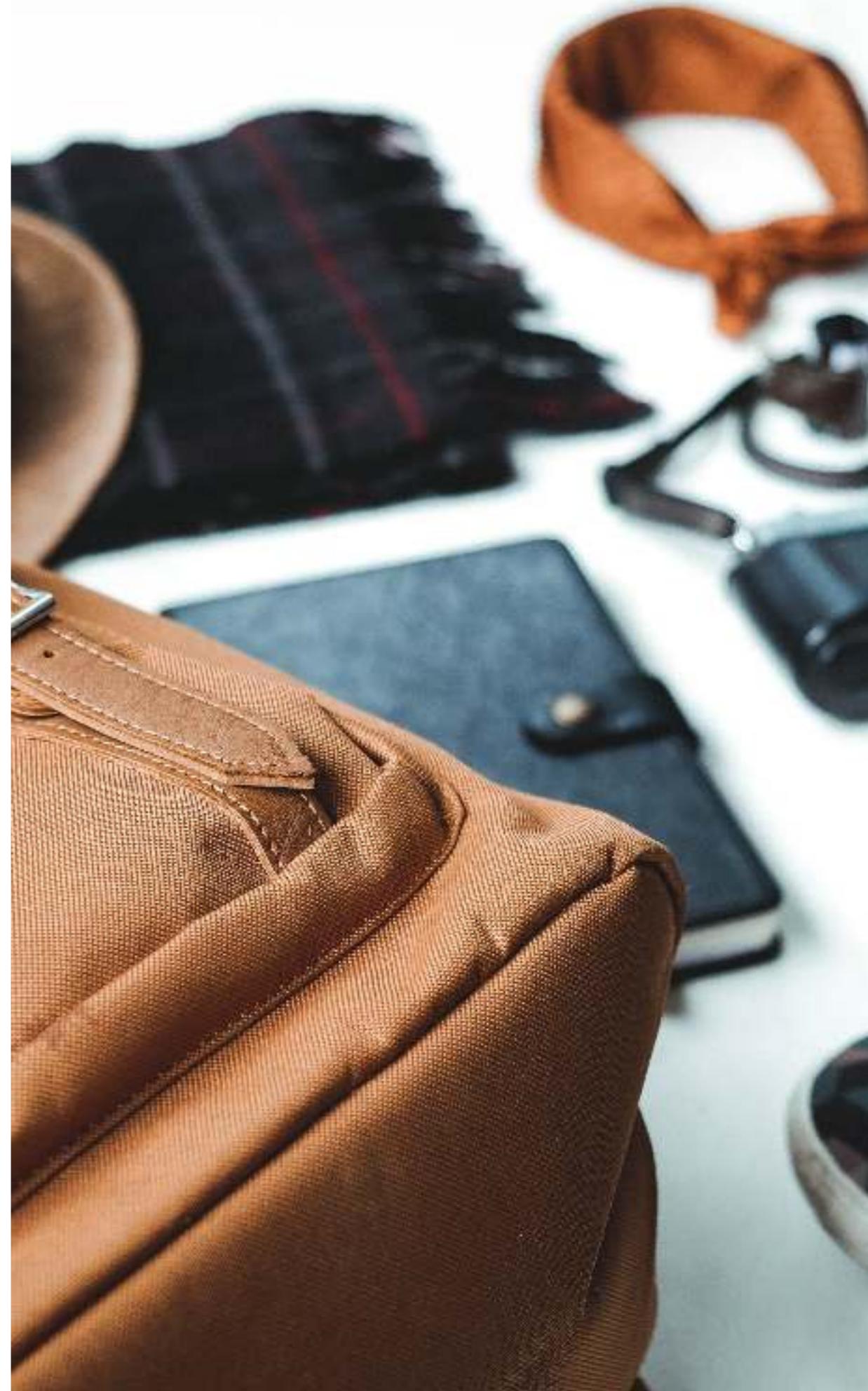
maximize $c^T x$
subject to $a^T x \leq b$
 $0 \leq x \leq u$
 $x \in \mathbf{Z}^n$



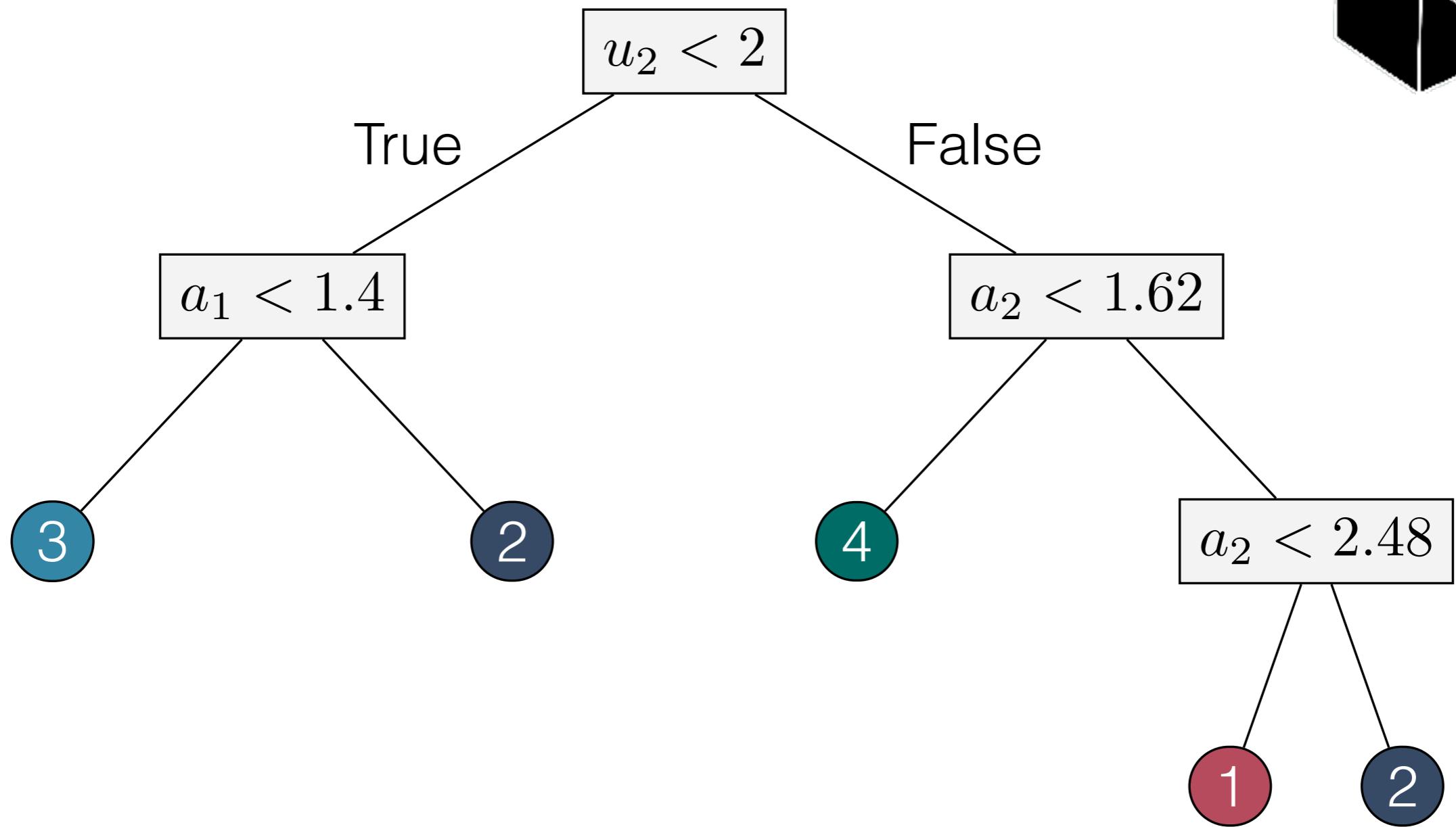
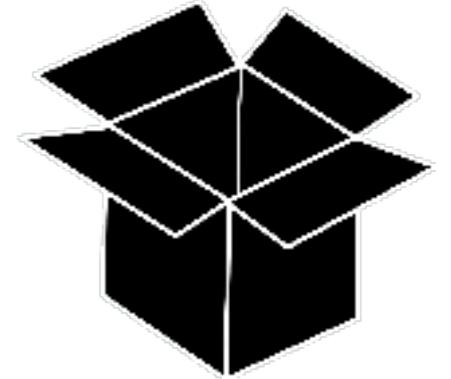
Knapsack

maximize $c^T x$
subject to $a^T x \leq b$
 $0 \leq x \leq u$
 $x \in \mathbf{Z}^n$

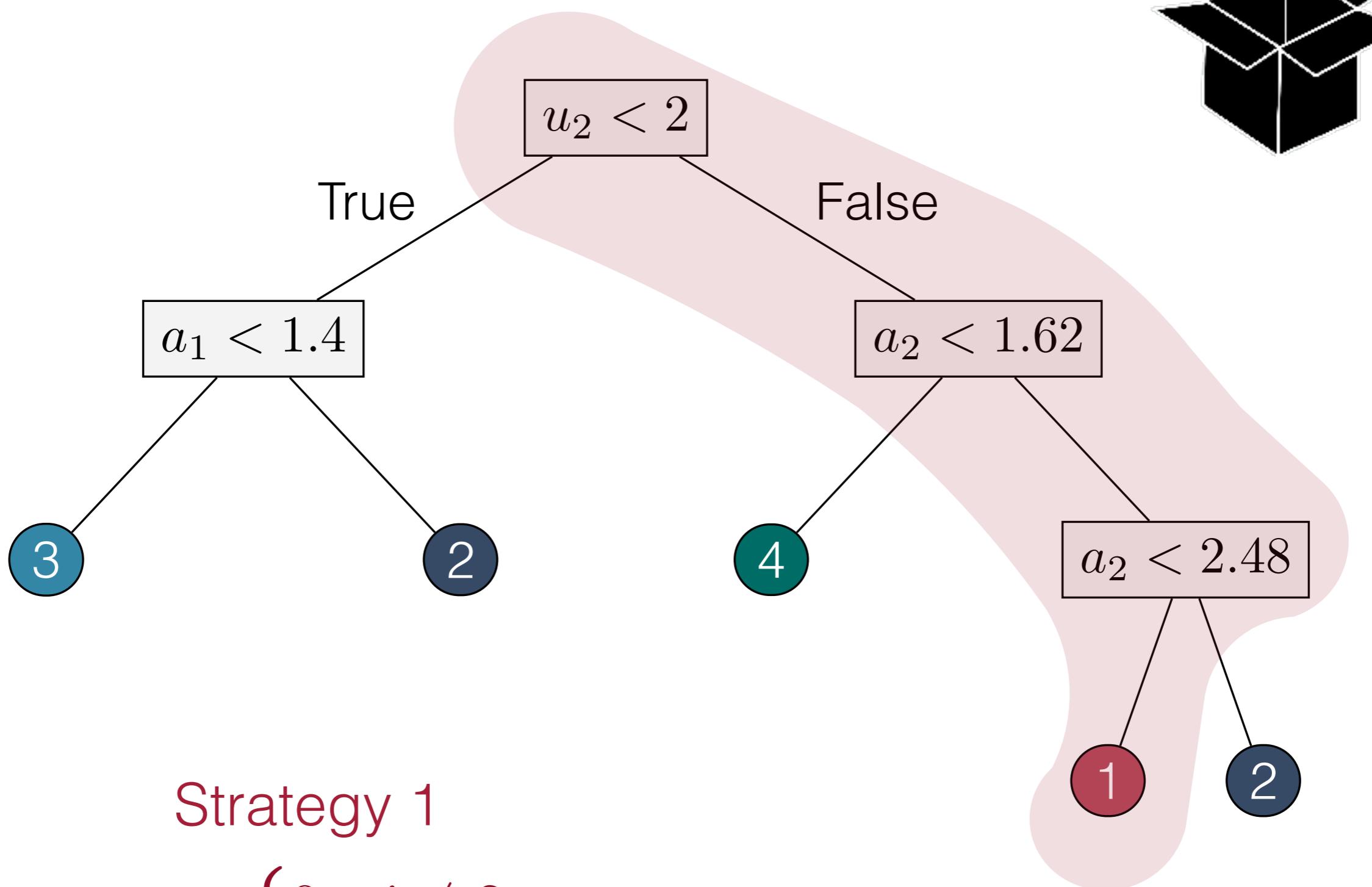
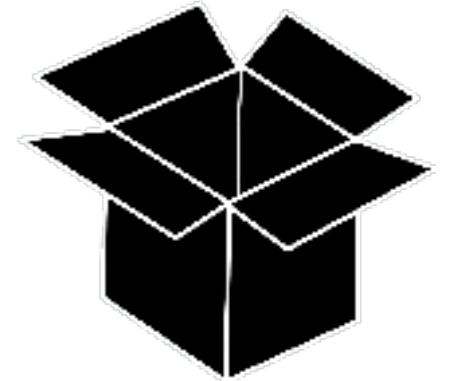
Parameters



Strategy selection



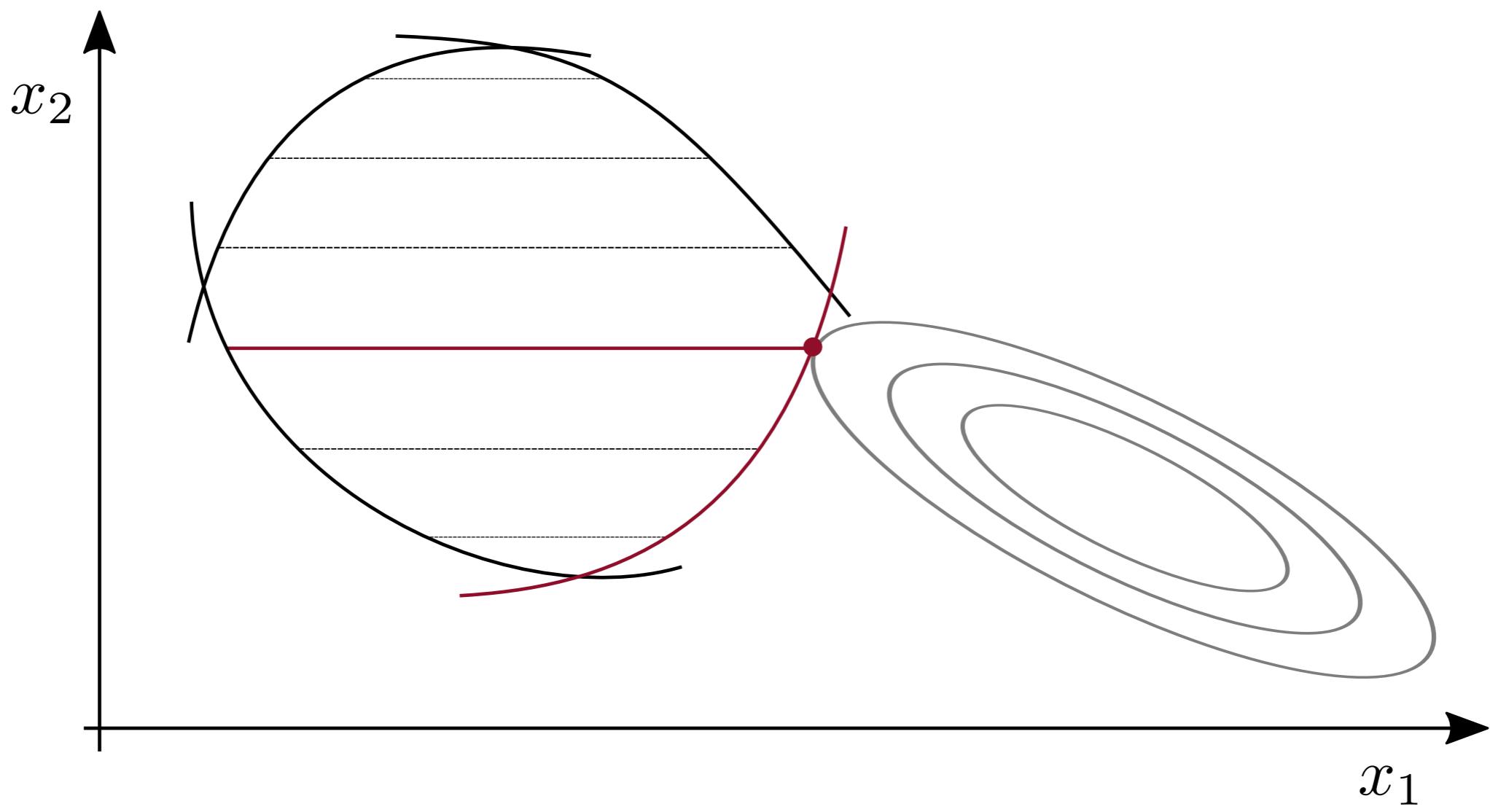
Strategy selection



$$x_i = \begin{cases} 0 & i \neq 2 \\ 2 & i = 2 \end{cases}$$

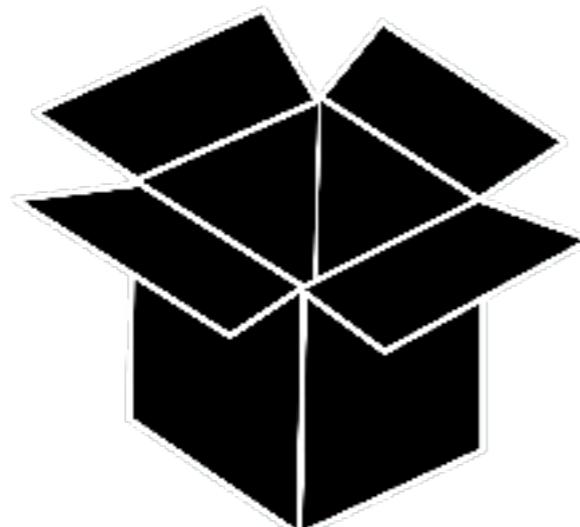
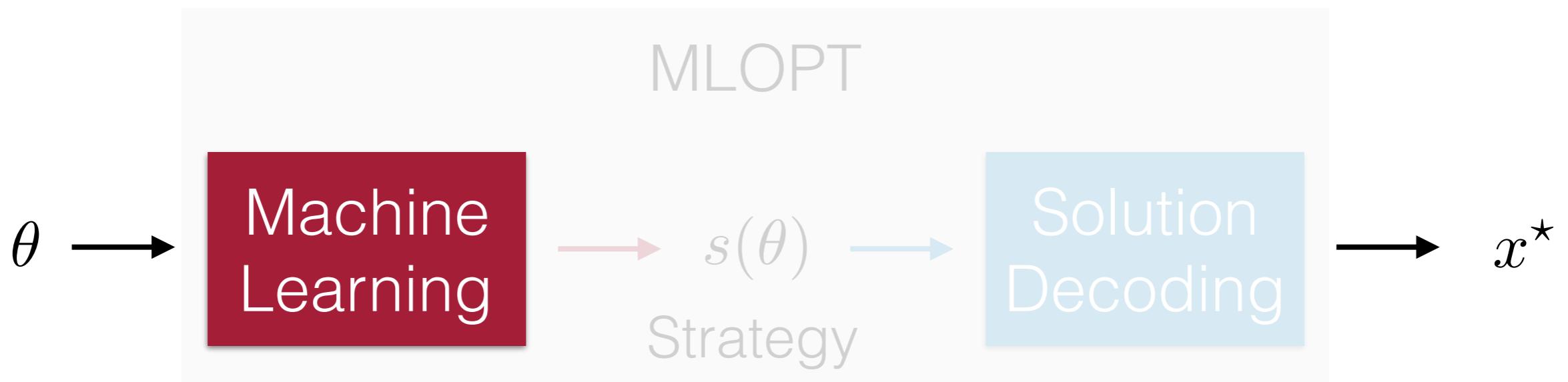
Mixed-integer convex optimization

$$\begin{array}{ll}\text{minimize} & f(\theta, x) \\ \text{subject to} & g(\theta, x) \leq 0 \\ & x_{\mathcal{I}} \in \mathbf{Z}^d\end{array}$$



Learning the strategies

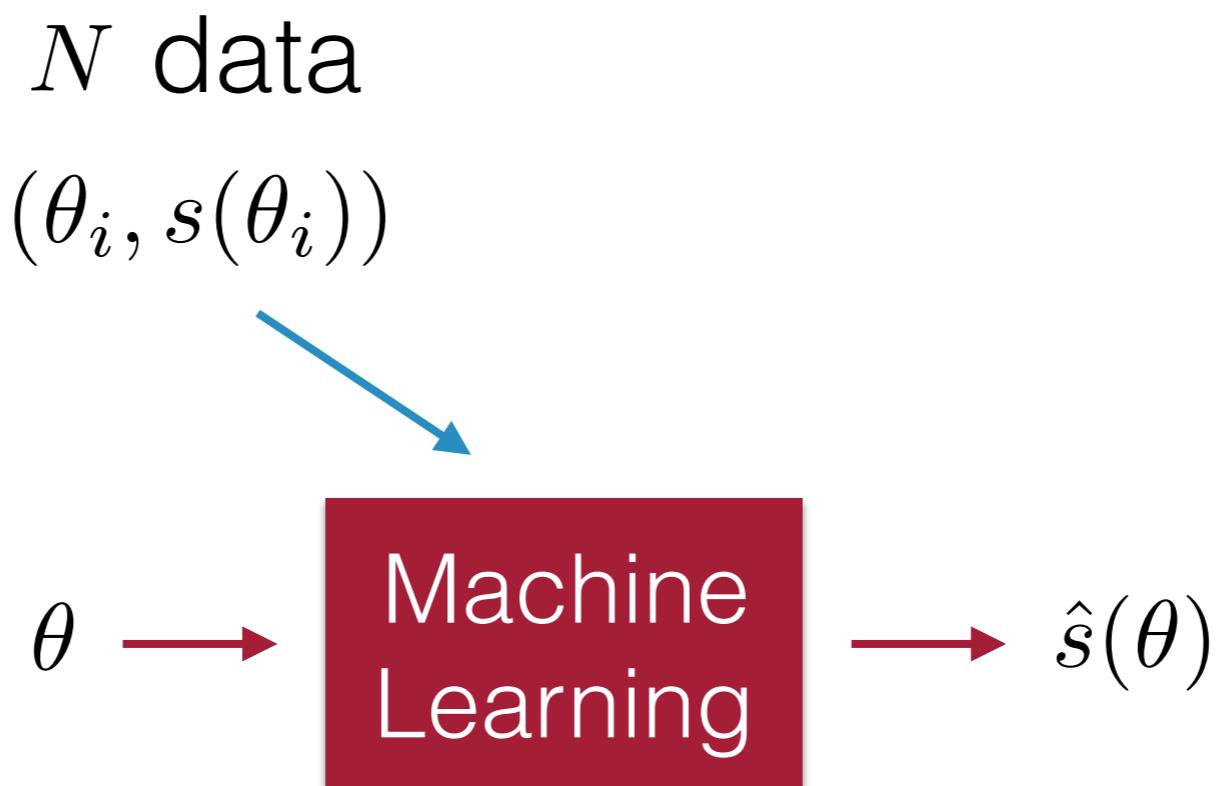
Machine Learning Optimizer



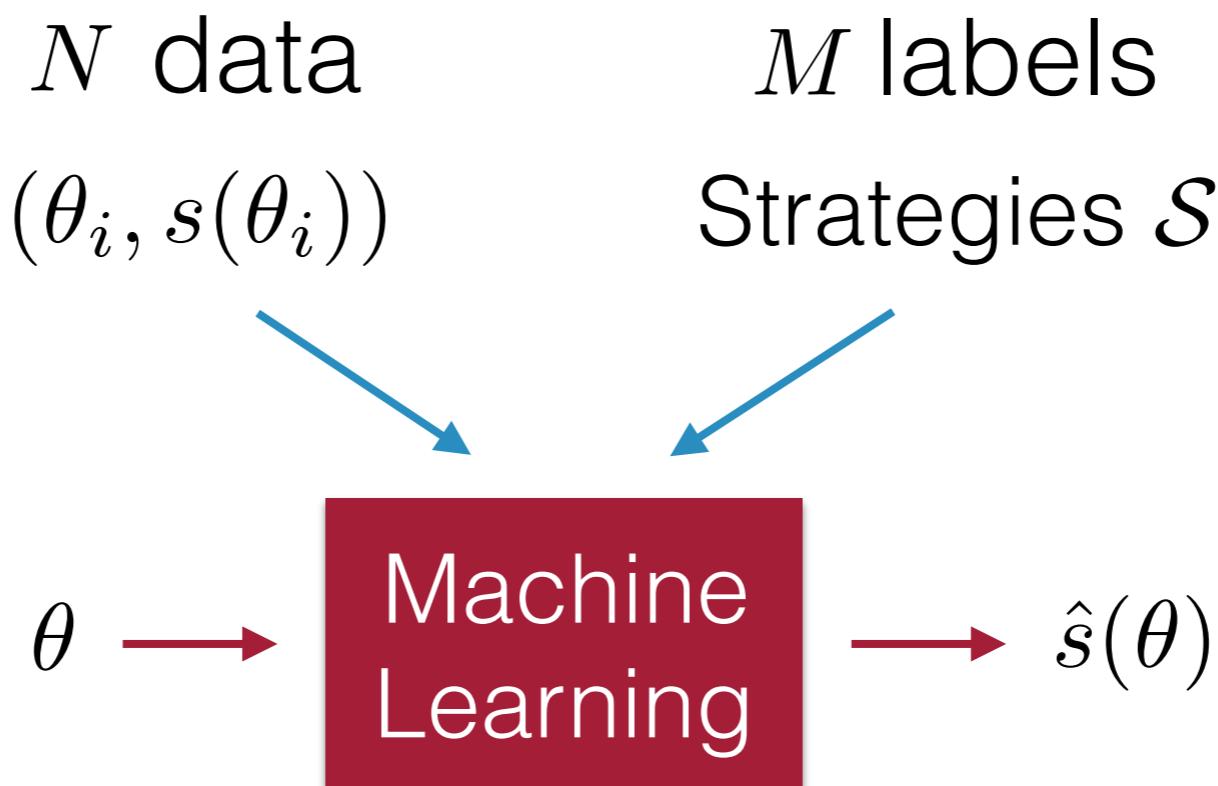
Selecting strategies is a classification problem



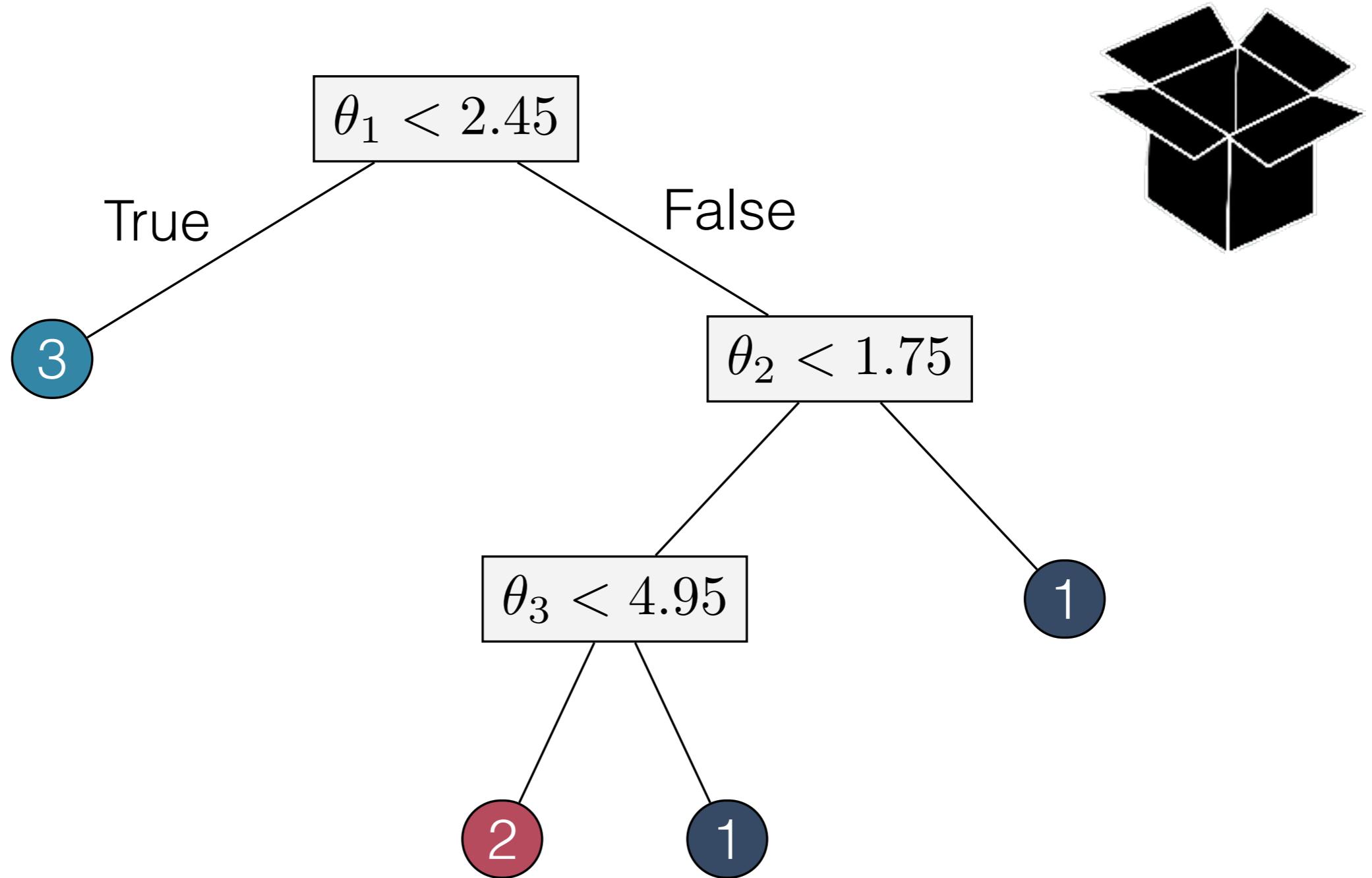
Selecting strategies is a classification problem



Selecting strategies is a classification problem

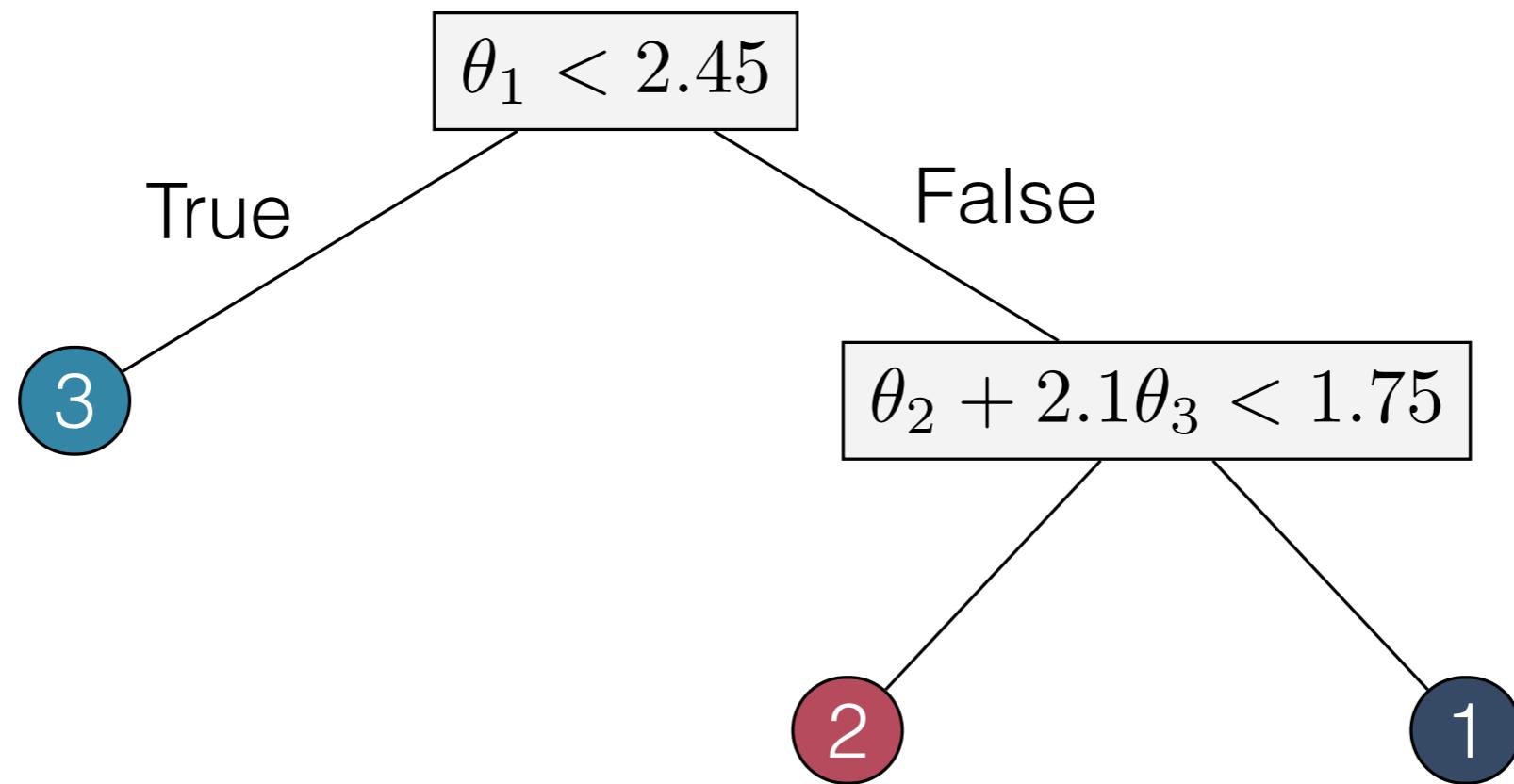
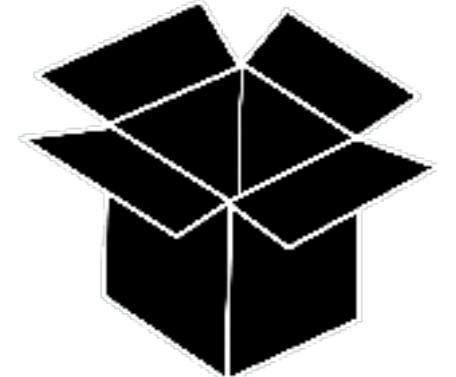


Optimal Classification Trees (OCT)



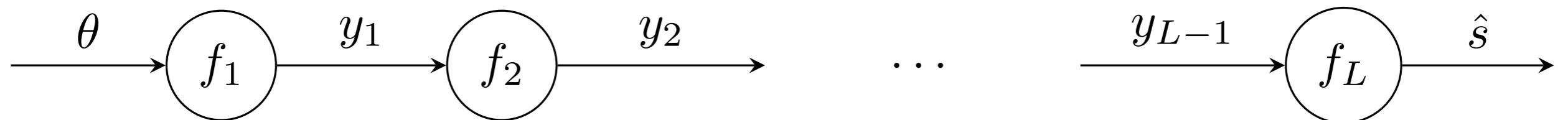
[Bertsimas and Dunn (2017)]

Optimal Classification Trees with Hyperplanes (OCT-H)



[Bertsimas and Dunn (2017)]

Neural Networks (NN)



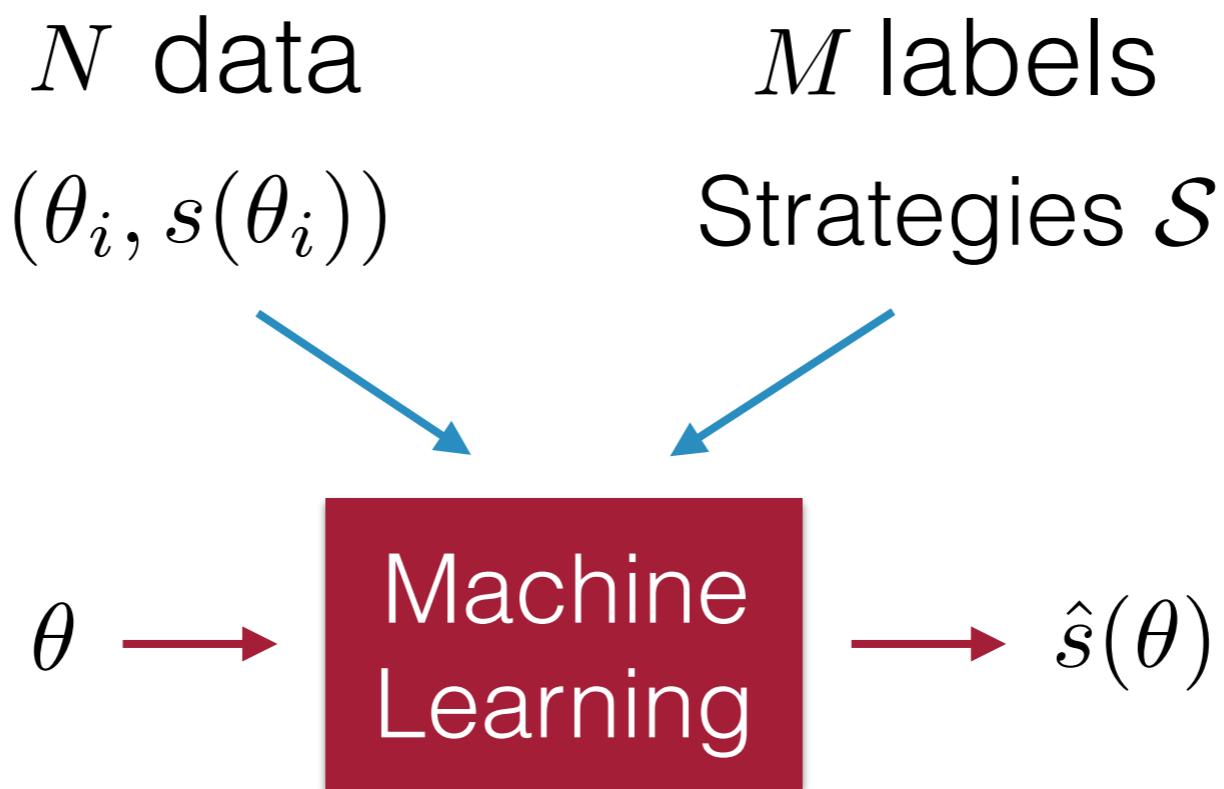
Single layer

$$y_l = f(y_{l-1}) = (W_l y_{l-1} + b_l)_+$$

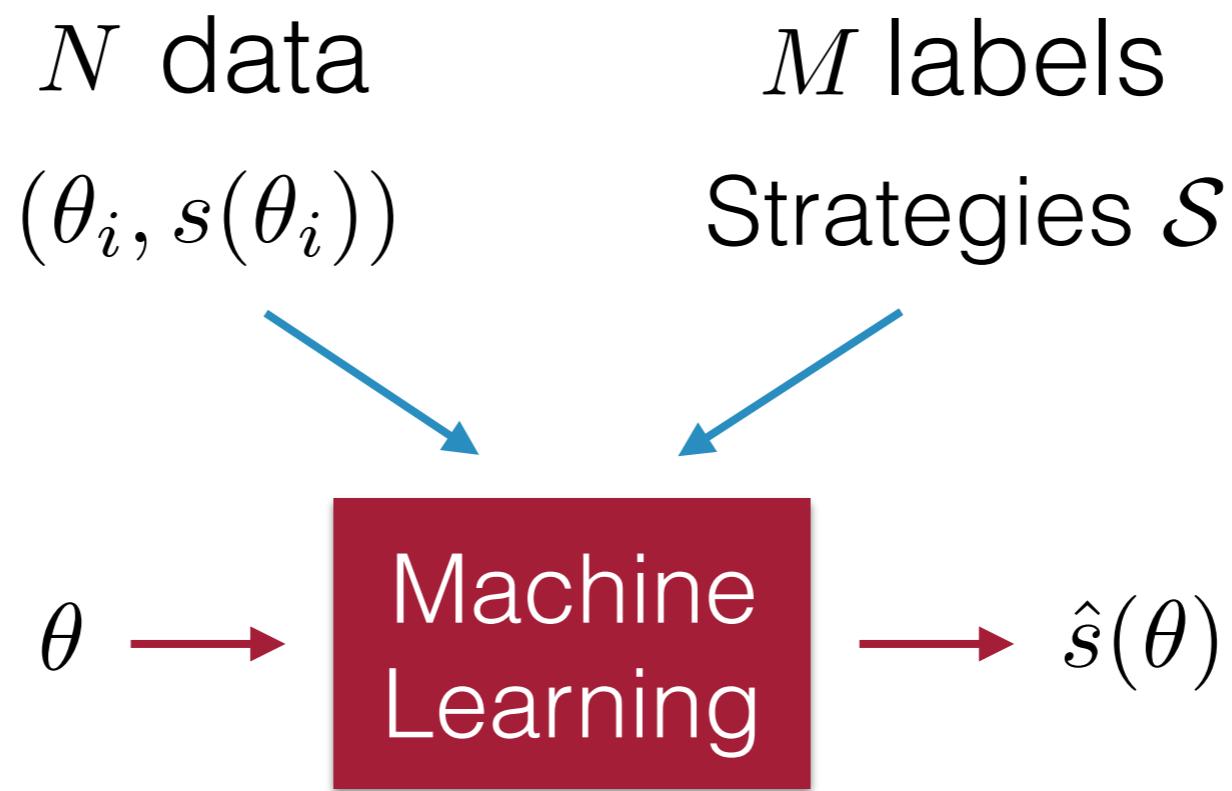
 PyTorch

Strategies exploration

Classification problem



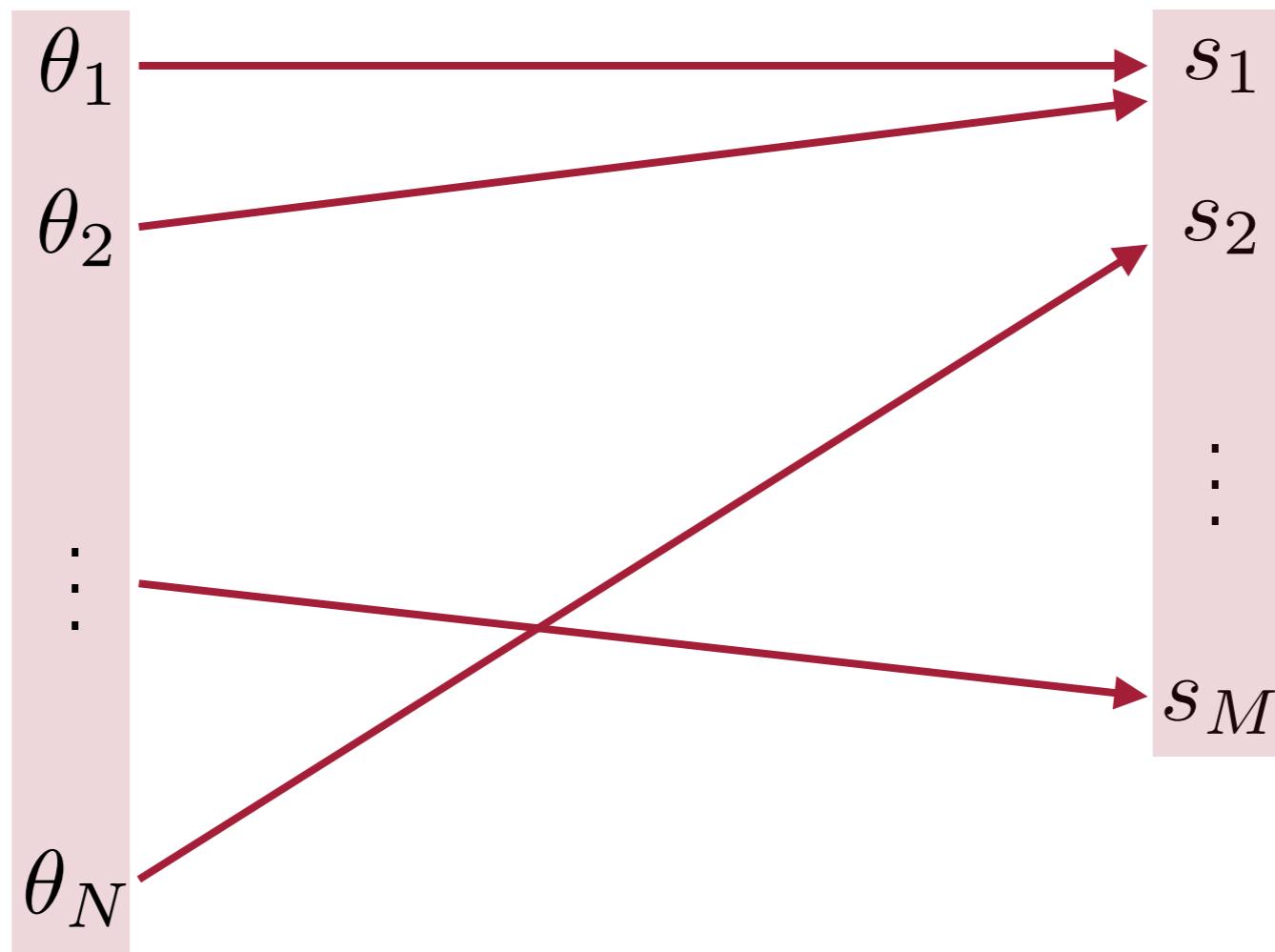
Classification problem



Have we seen enough data?

Will we find new strategies?

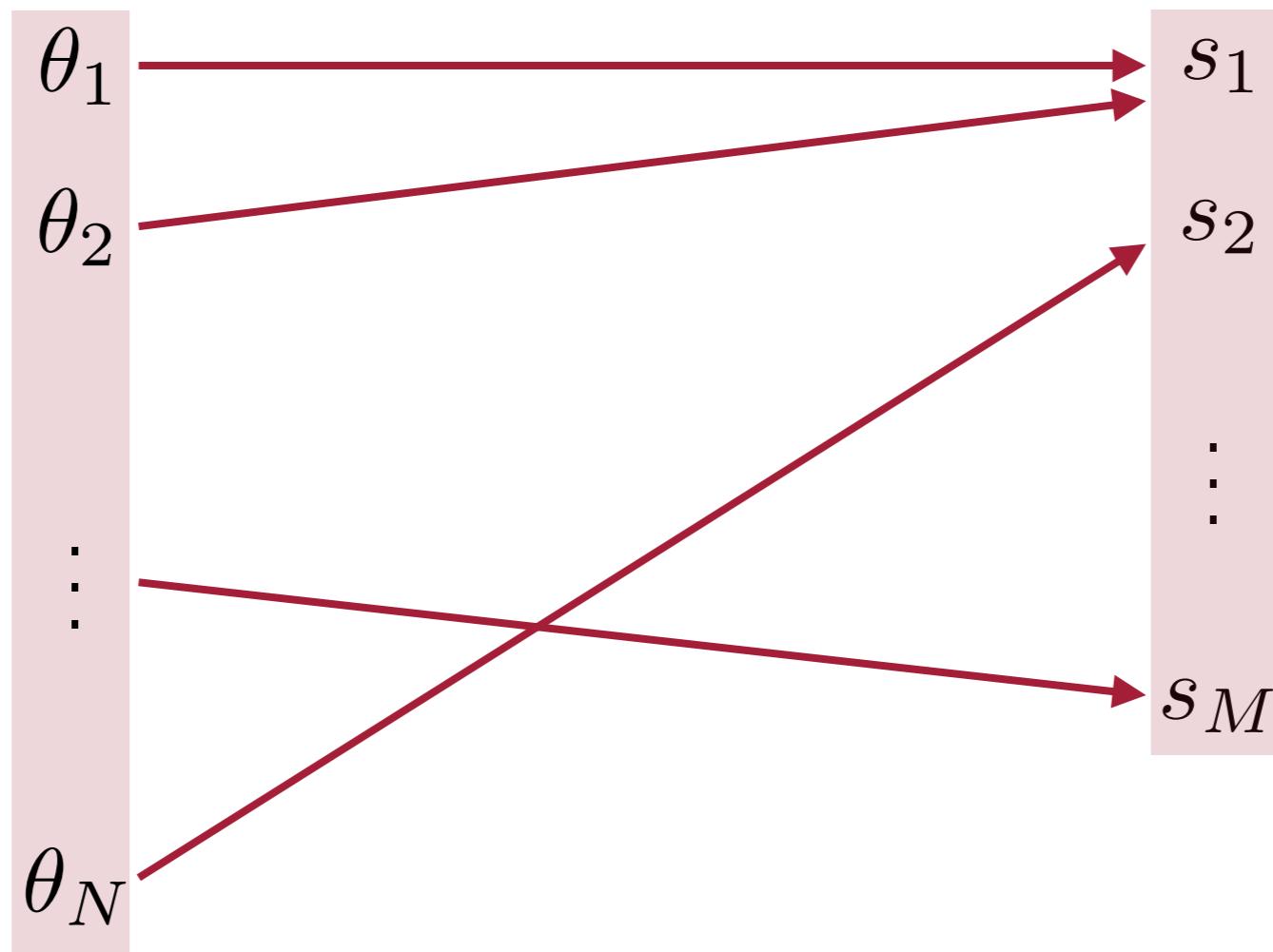
Parameters



Strategies

Will we find new strategies?

Parameters



$\theta_{N+1}?$

Alan Turing already knew that

Decoded the Enigma Machine
in
World War II

Only some words (labels) needed



Good-Turing estimator

s_1	12 times
s_2	45 times
\vdots	\vdots
s_M	2 times

Good-Turing estimator

s_1	12 times
s_2	45 times
\vdots	\vdots
s_M	2 times

Probability of unseen strategies

$$GT = \frac{N_1}{N} \approx \mathbf{P}(\theta_{N+1} \in \mathbf{R}^p \mid s(\theta_{N+1}) \notin \mathcal{S}(\Theta_N))$$

Good-Turing estimator

s_1	12 times
s_2	45 times
\vdots	\vdots
s_M	2 times

strategies
appeared once Probability of unseen strategies

$$GT = \frac{N_1}{N} \approx P(\theta_{N+1} \in \mathbf{R}^p \mid s(\theta_{N+1}) \notin \mathcal{S}(\Theta_N))$$

Good-Turing estimator

s_1	12 times
s_2	45 times
\vdots	\vdots
s_M	2 times

strategies
appeared once Probability of unseen strategies

$$GT = \frac{N_1}{N} \approx P(\theta_{N+1} \in \mathbf{R}^p \mid s(\theta_{N+1}) \notin \mathcal{S}(\Theta_N))$$

samples

Good-Turing estimator

s_1	12 times
s_2	45 times
\vdots	\vdots
s_M	2 times

strategies
appeared once Probability of unseen strategies

$$GT = \frac{N_1}{N} \approx \mathbf{P}(\theta_{N+1} \in \mathbf{R}^p \mid s(\theta_{N+1}) \notin \mathcal{S}(\Theta_N))$$

samples

Bound

$$\mathbf{P}(\theta_{N+1} \in \mathbf{R}^p \mid s(\theta_{N+1}) \notin \mathcal{S}(\Theta_N)) \leq GT + c\sqrt{(1/N) \ln(3/\beta)}$$

Simple sampling scheme

Bound

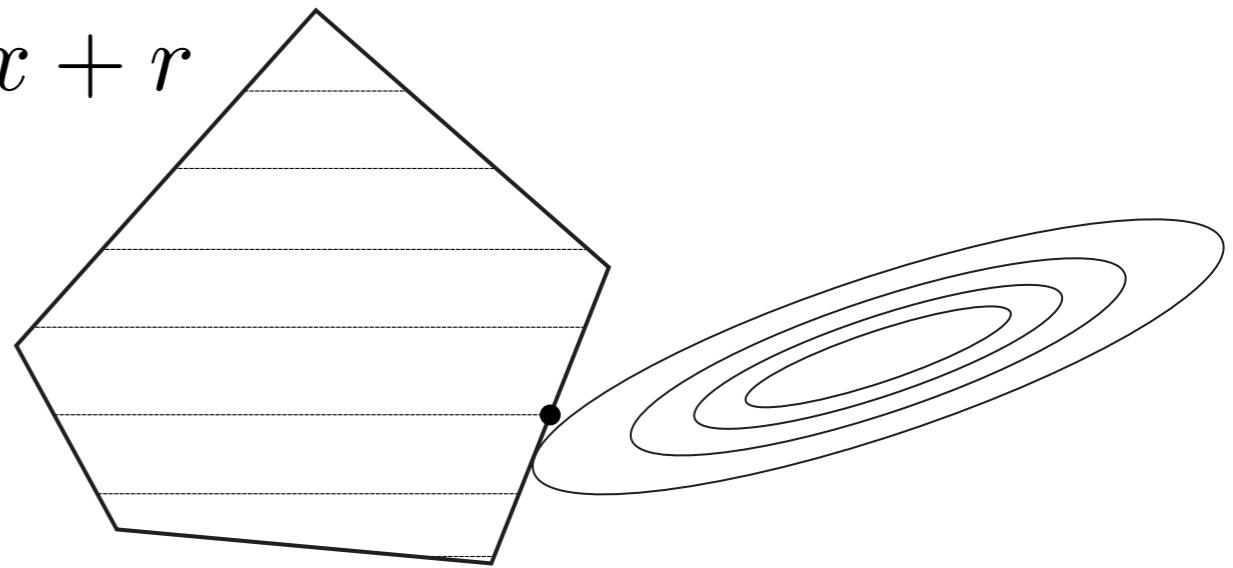
Repeat until $GT + c\sqrt{(1/N) \ln(3/\beta)} \leq \epsilon$

1. sample θ_i
2. compute $s(\theta_i)$
3. update estimator $GT = \frac{N_1}{N}$

Speedups

Mixed-integer quadratic optimization

minmimize $(1/2)x^T Px + q^T x + r$
subject to $Ax \leq b$
 $x_{\mathcal{I}} \in \mathbf{Z}^d$

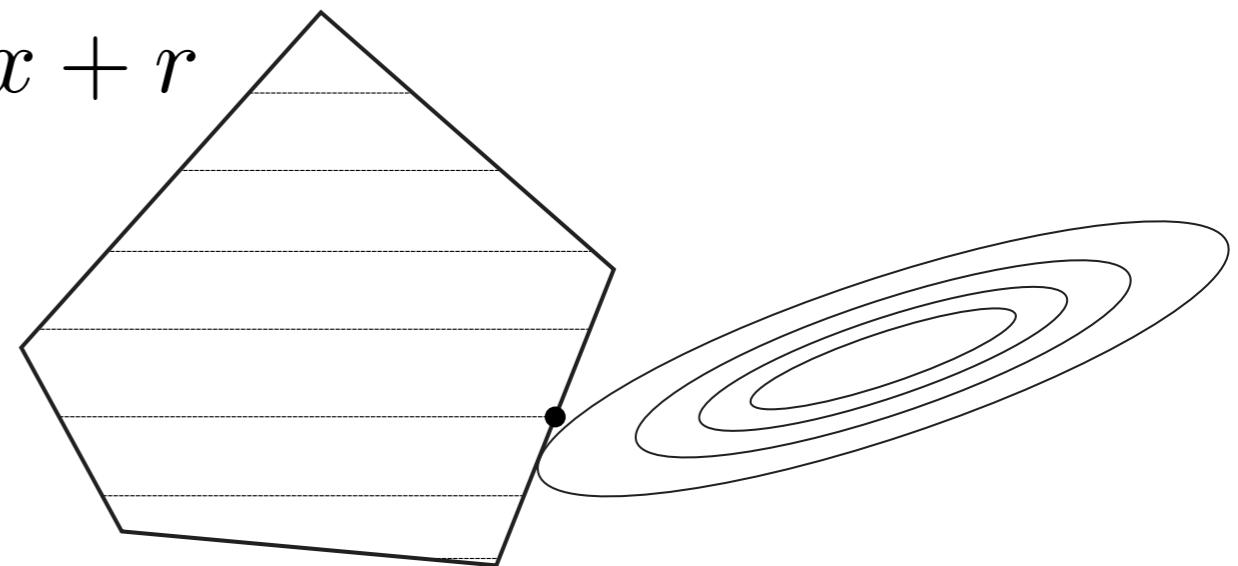


Mixed-integer quadratic optimization

$$\begin{array}{ll}\text{minimize} & (1/2)x^T Px + q^T x + r \\ \text{subject to} & Ax \leq b \\ & x_{\mathcal{I}} \in \mathbf{Z}^d\end{array}$$



Strategy



$$\begin{array}{ll}\text{minimize} & (1/2)x^T Px + q^T x + r \\ \text{subject to} & A_{\mathcal{T}(\theta)}x = b_{\mathcal{T}(\theta)} \\ & x_{\mathcal{I}} = x_{\mathcal{I}}^*(\theta)\end{array}$$

Equality
Constrained
QP

Quick solution

KKT System

$$\begin{bmatrix} P & A_{\mathcal{T}(\theta)}^T & I_{\mathcal{I}}^T \\ A_{\mathcal{T}(\theta)} & 0 & \\ I_{\mathcal{I}} & & \end{bmatrix} \begin{bmatrix} x \\ \nu \end{bmatrix} = \begin{bmatrix} -q \\ b_{\mathcal{T}(\theta)} \\ x_{\mathcal{I}}^{\star}(\theta) \end{bmatrix}$$

Quick solution

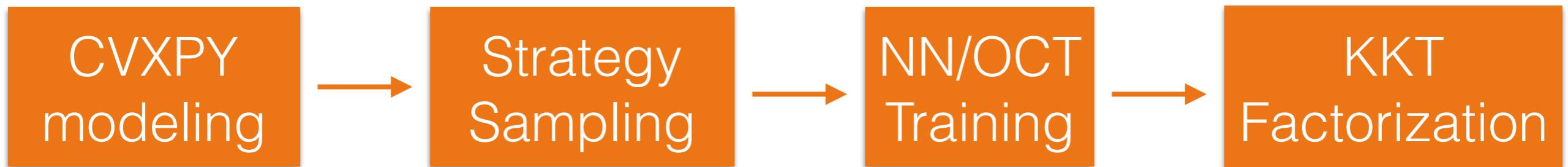
KKT System

$$\begin{bmatrix} P & A_{\mathcal{T}(\theta)}^T & I_{\mathcal{I}}^T \\ A_{\mathcal{T}(\theta)} & 0 & \\ I_{\mathcal{I}} & & \end{bmatrix} \begin{bmatrix} x \\ \nu \end{bmatrix} = \begin{bmatrix} -q \\ b_{\mathcal{T}(\theta)} \\ x_{\mathcal{I}}^*(\theta) \end{bmatrix}$$

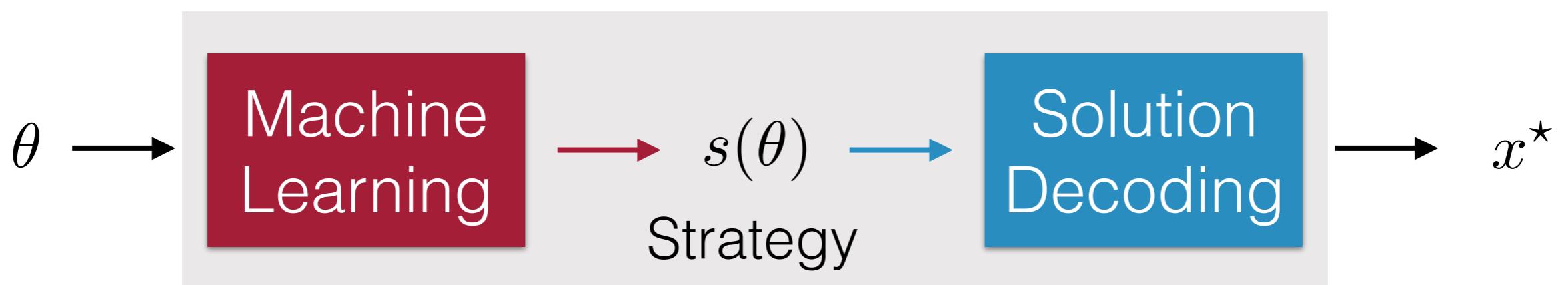


MLOPT: Machine Learning Optimizer

Offline Learning

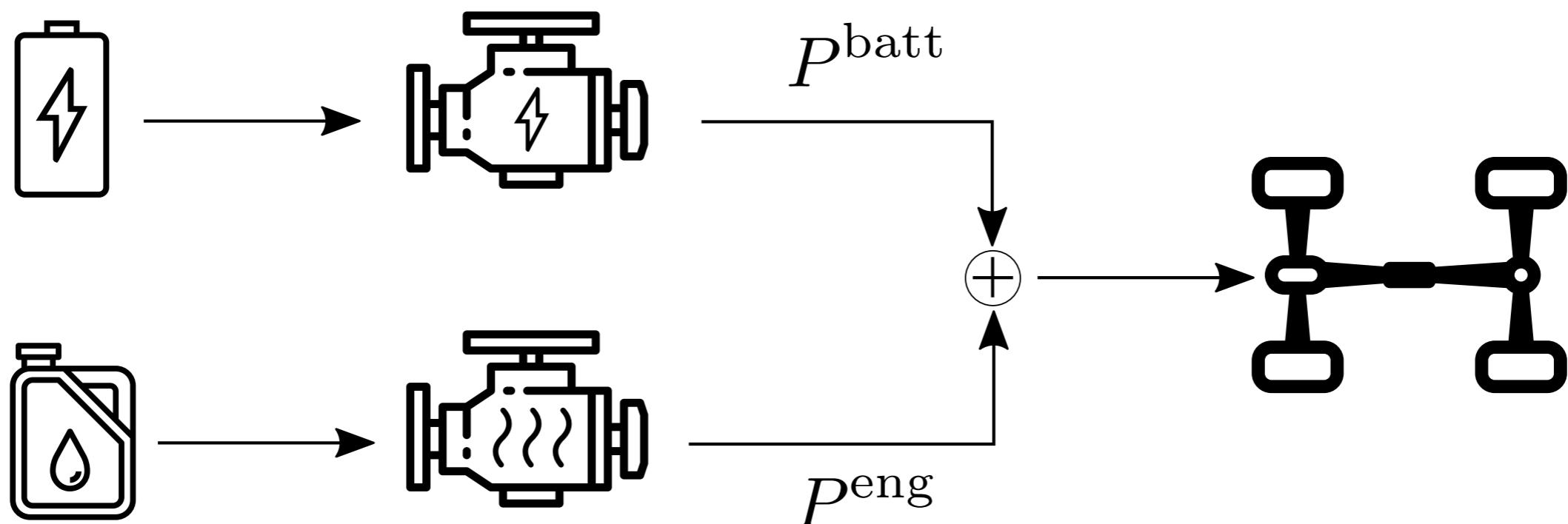


Online Optimization

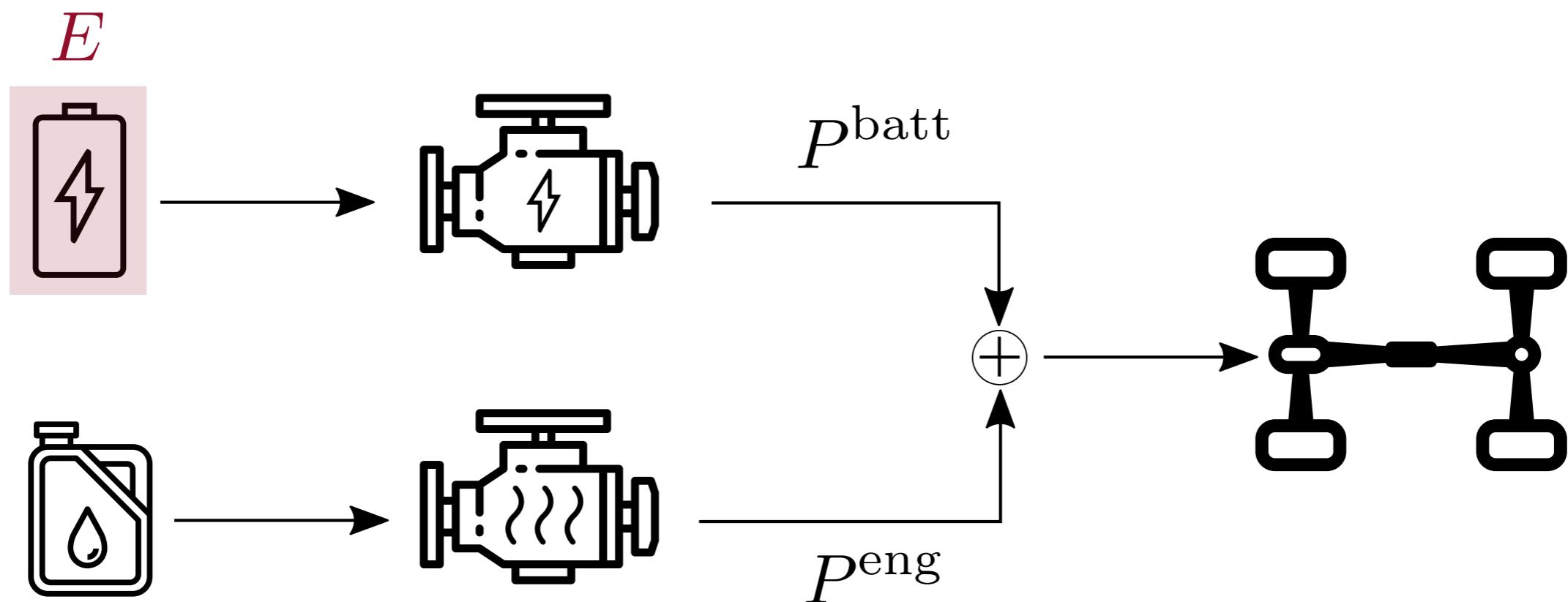


Benchmarks

Hybrid-vehicle control



Hybrid-vehicle control



Hybrid-vehicle control

$$\begin{aligned} & \text{maximize} && \sum_{t=0}^{T-1} f(P_t^{\text{eng}}, z_t) + \delta(z_t - z_{t-1})_+ \\ & \text{subject to} && P_t^{\text{batt}} + P_t^{\text{eng}} \geq P_t^{\text{des}} \end{aligned}$$

$$\begin{aligned} E_{t+1} &= E_t - \tau P_t^{\text{batt}} \\ E_0 &= E_{\text{init}} \end{aligned}$$

$$z_t \in \{0, 1\}$$

Hybrid-vehicle control

maximize
$$\sum_{t=0}^{T-1} f(P_t^{\text{eng}}, z_t) + \delta(z_t - z_{t-1}) +$$

subject to
$$P_t^{\text{batt}} + P_t^{\text{eng}} \geq P_t^{\text{des}}$$
 Desired Power

$$E_{t+1} = E_t - \tau P_t^{\text{batt}}$$
$$E_0 = E_{\text{init}}$$

$$z_t \in \{0, 1\}$$

Hybrid-vehicle control

maximize
$$\sum_{t=0}^{T-1} f(P_t^{\text{eng}}, z_t) + \delta(z_t - z_{t-1})_+$$

subject to $P_t^{\text{batt}} + P_t^{\text{eng}} \geq P_t^{\text{des}}$ Desired Power

$$\begin{aligned} E_{t+1} &= E_t - \tau P_t^{\text{batt}} \\ E_0 &= E_{\text{init}} \end{aligned}$$
 Battery Dynamics

$$z_t \in \{0, 1\}$$

Hybrid-vehicle control

maximize
$$\sum_{t=0}^{T-1} f(P_t^{\text{eng}}, z_t) + \delta(z_t - z_{t-1})_+$$

subject to $P_t^{\text{batt}} + P_t^{\text{eng}} \geq P_t^{\text{des}}$ Desired Power

$$\begin{aligned} E_{t+1} &= E_t - \tau P_t^{\text{batt}} \\ E_0 &= E_{\text{init}} \end{aligned}$$
 Battery Dynamics

$$z_t \in \{0, 1\}$$
 Engine Switch

Hybrid-vehicle control

maximize $\sum_{t=0}^{T-1} f(P_t^{\text{eng}}, z_t) + \delta(z_t - z_{t-1}) +$

subject to $P_t^{\text{batt}} + P_t^{\text{eng}} \geq P_t^{\text{des}}$ Desired Power

$$E_{t+1} = E_t - \tau P_t^{\text{batt}}$$

$$E_0 = E_{\text{init}}$$

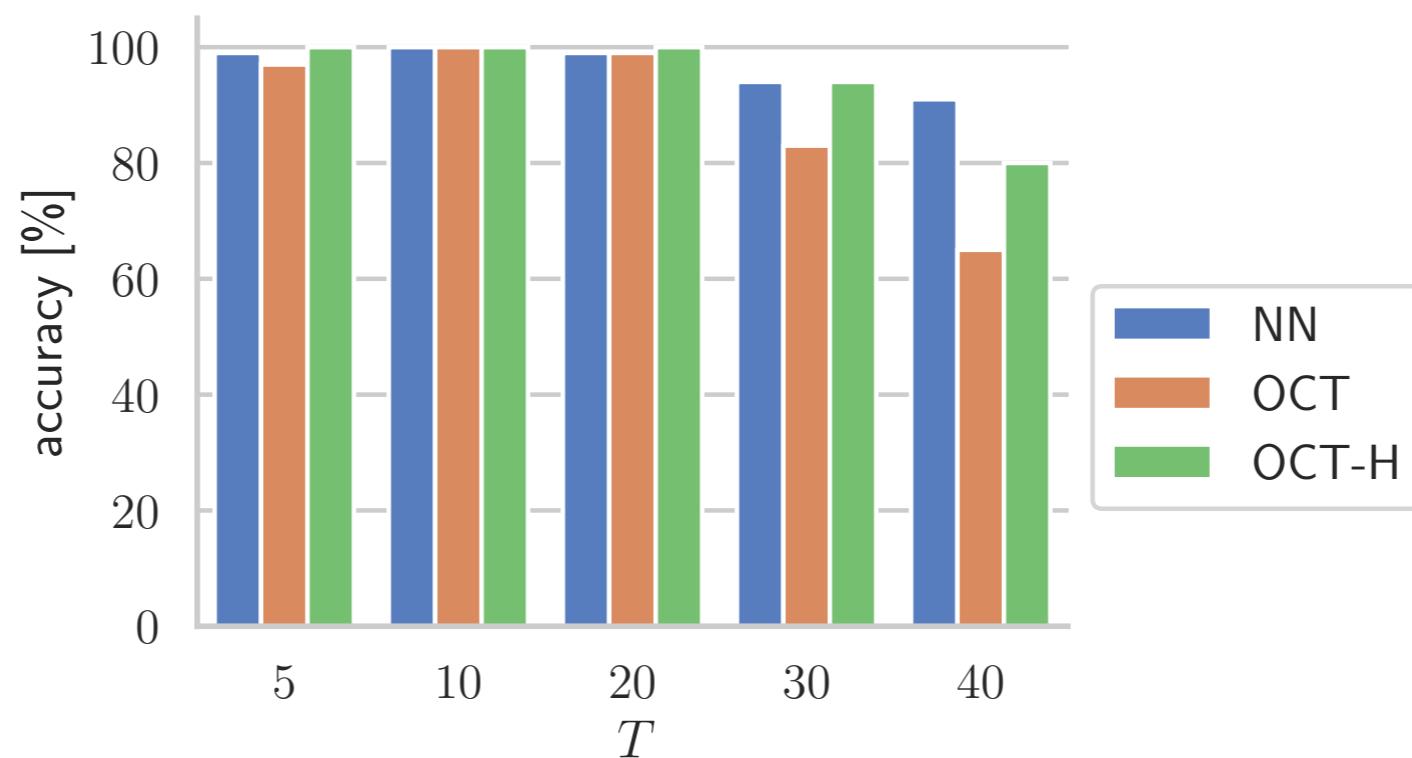
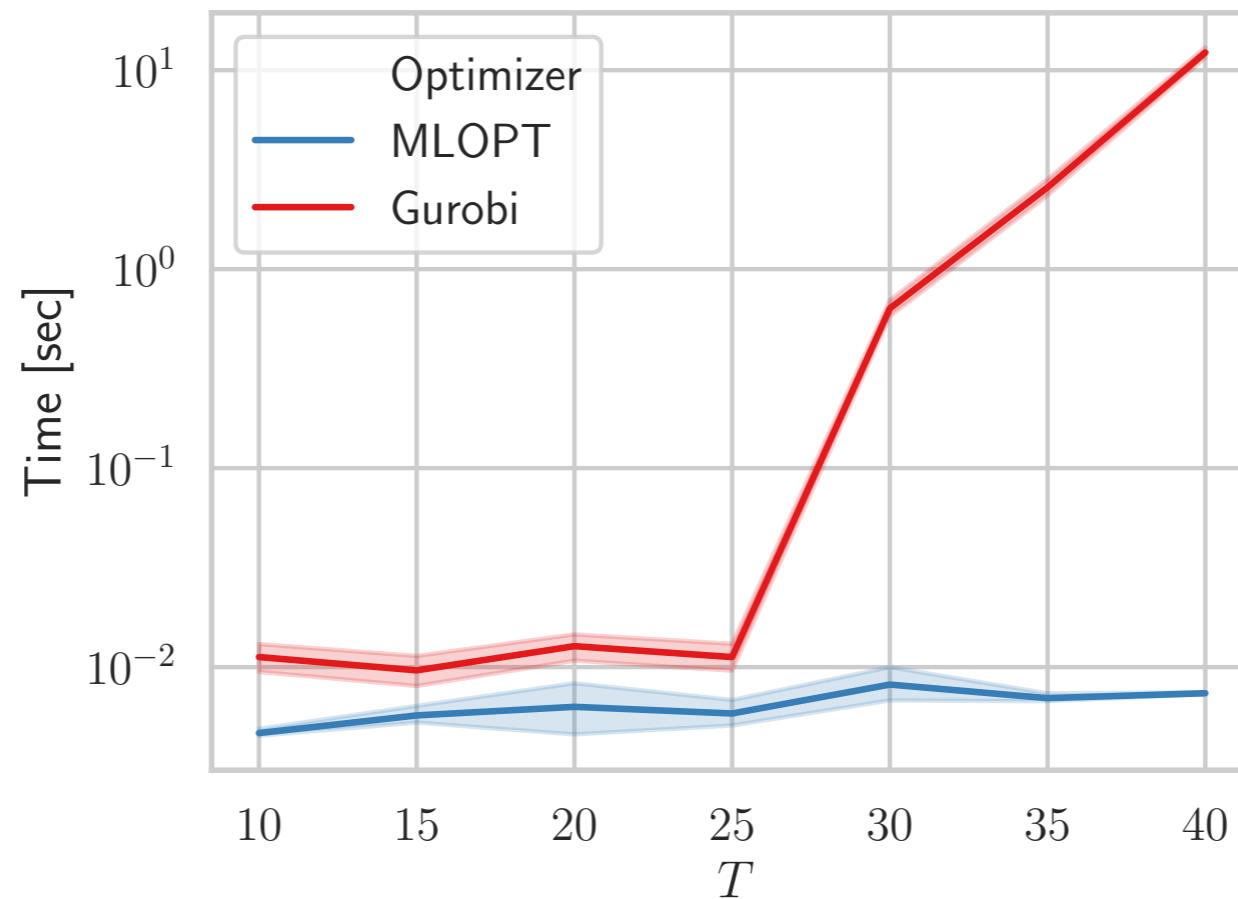
Battery
Dynamics

$$z_t \in \{0, 1\}$$

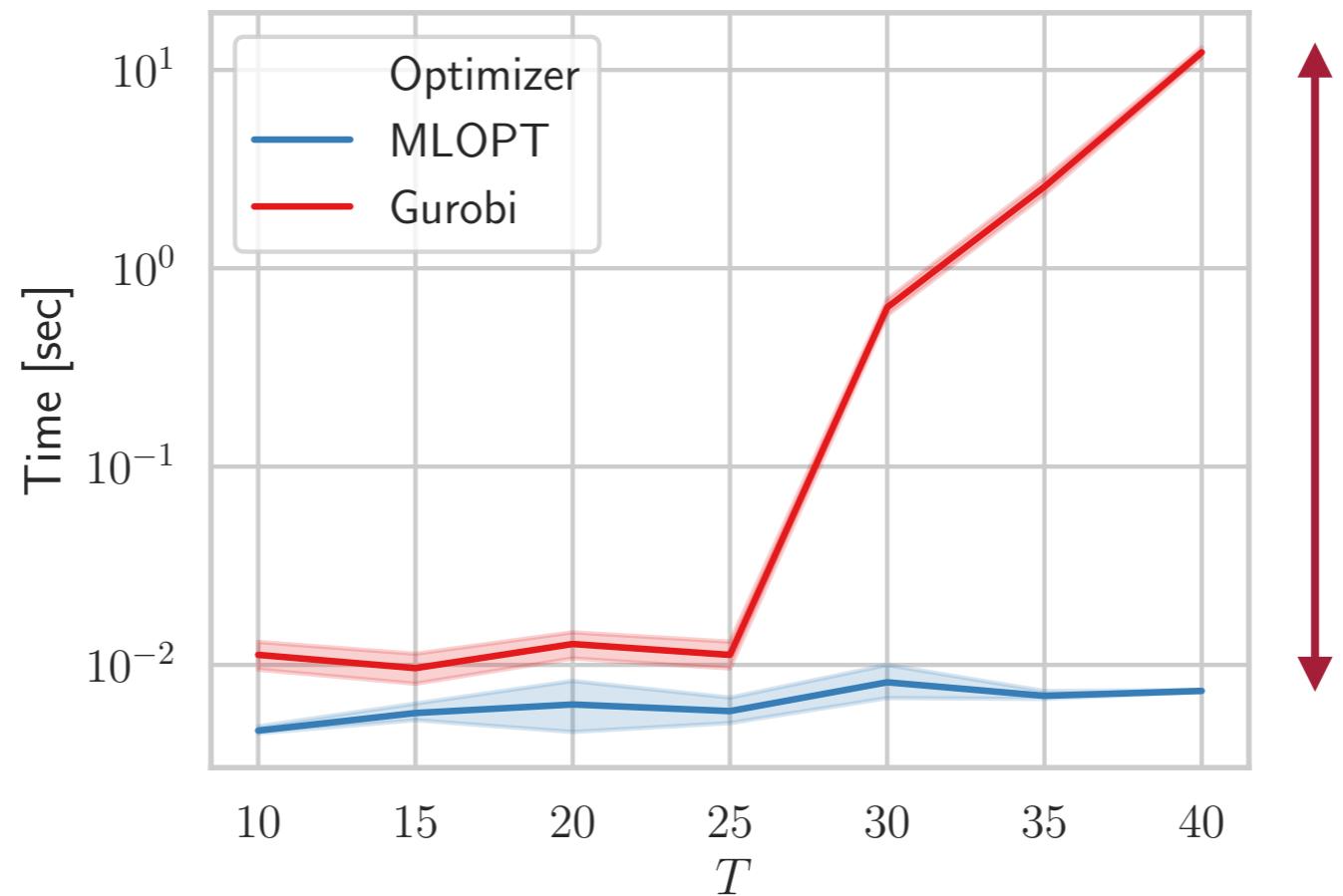
Engine Switch

Parameters

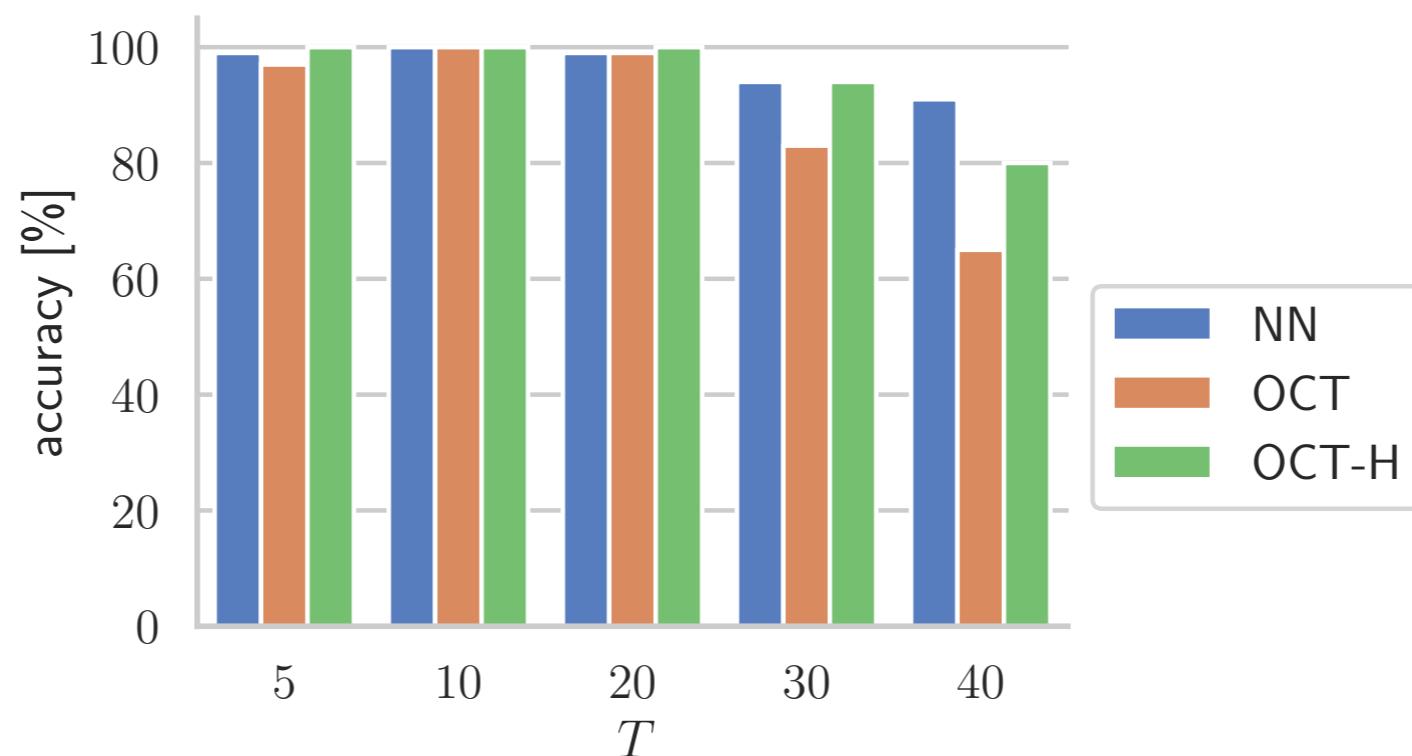
MPC results: high speed and accuracy



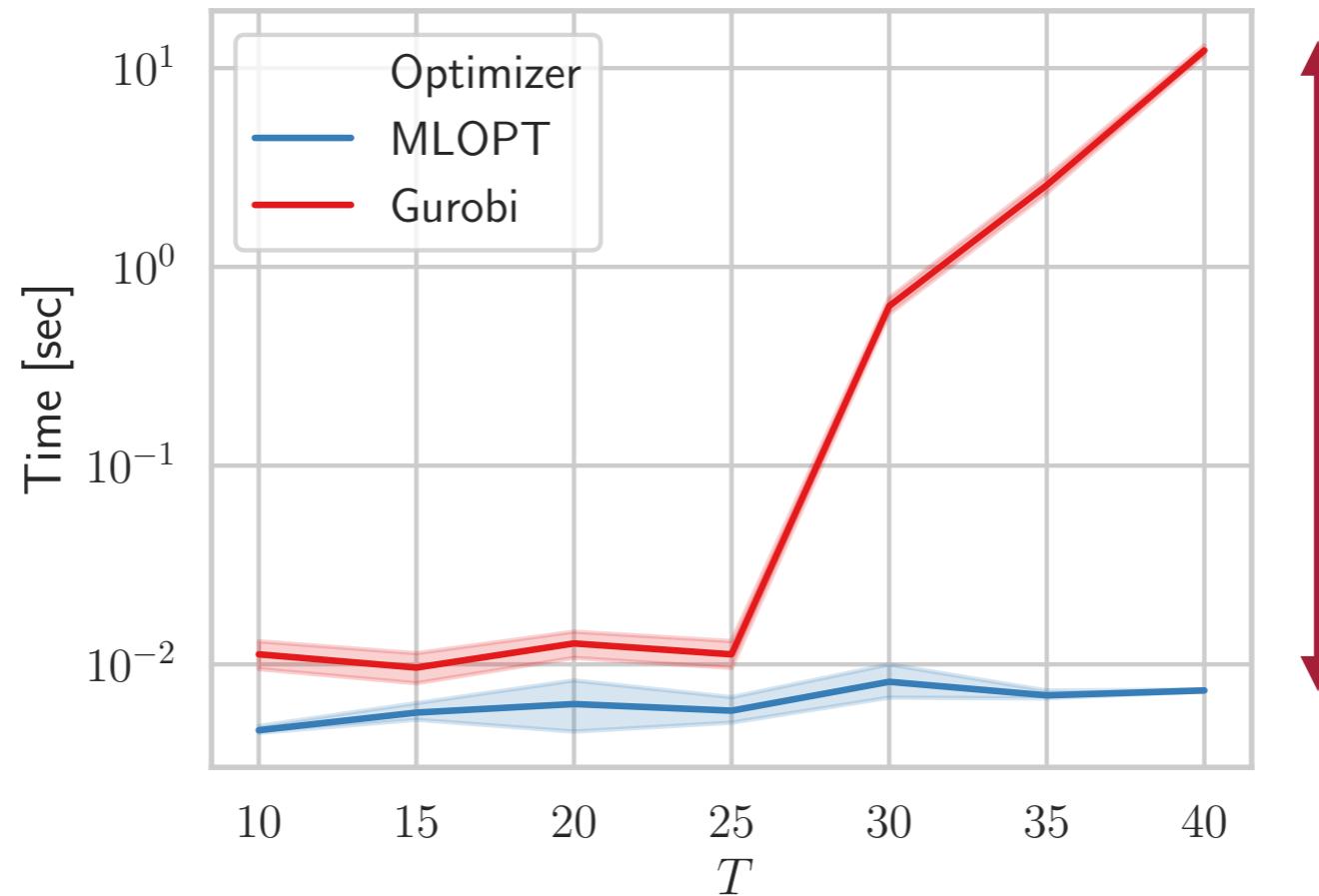
MPC results: high speed and accuracy



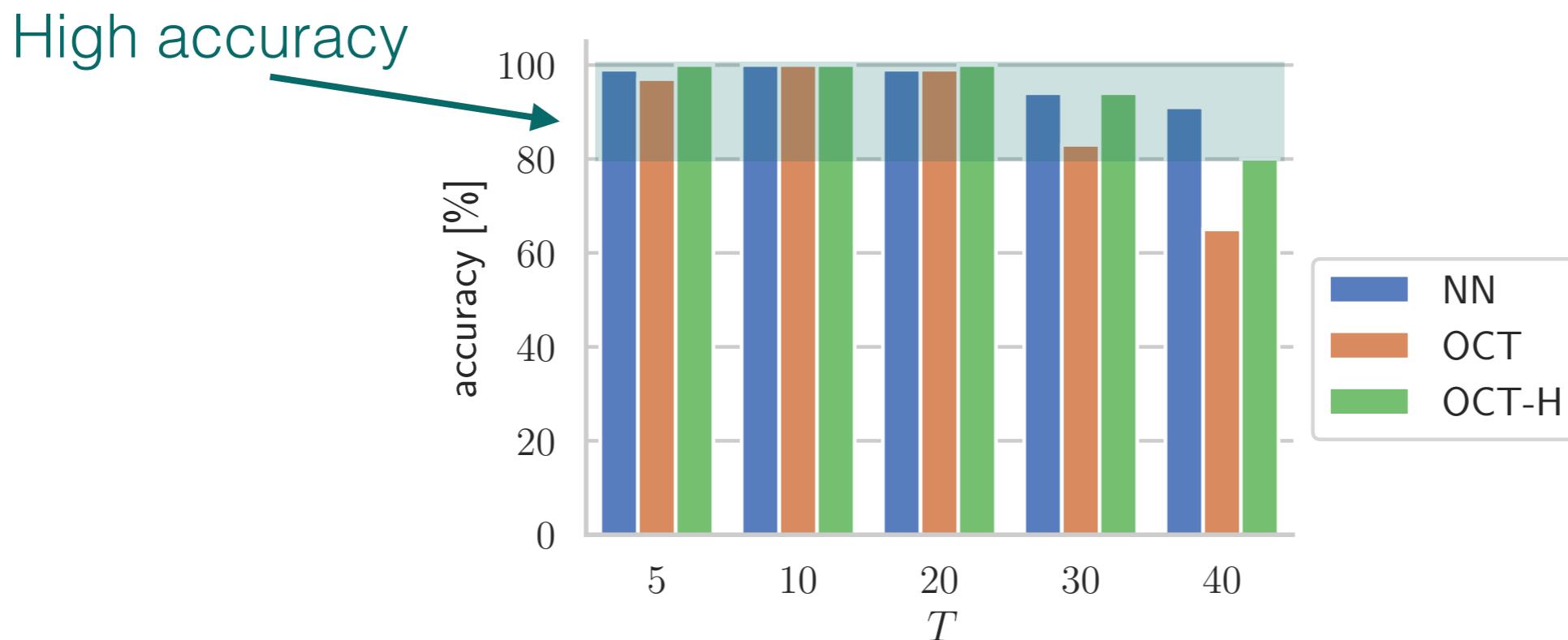
1000x
faster



MPC results: high speed and accuracy



1000x
faster

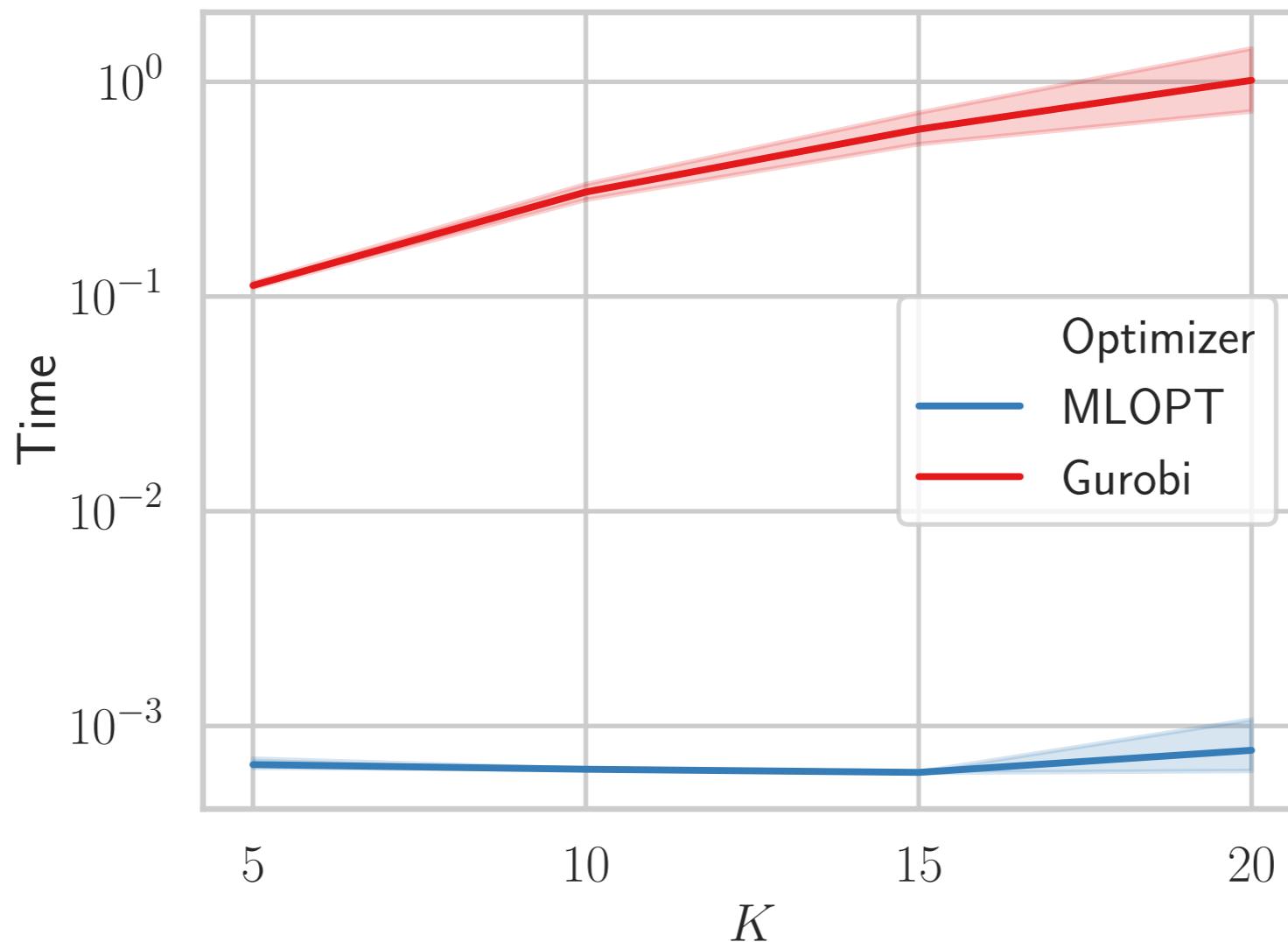


Sparse portfolio optimization

$$\begin{array}{ll}\text{maximize} & r^T w - \gamma w^T \Sigma w \\ \text{subject to} & 1^T w = 1 \\ & \text{card}(w) \leq c\end{array}$$

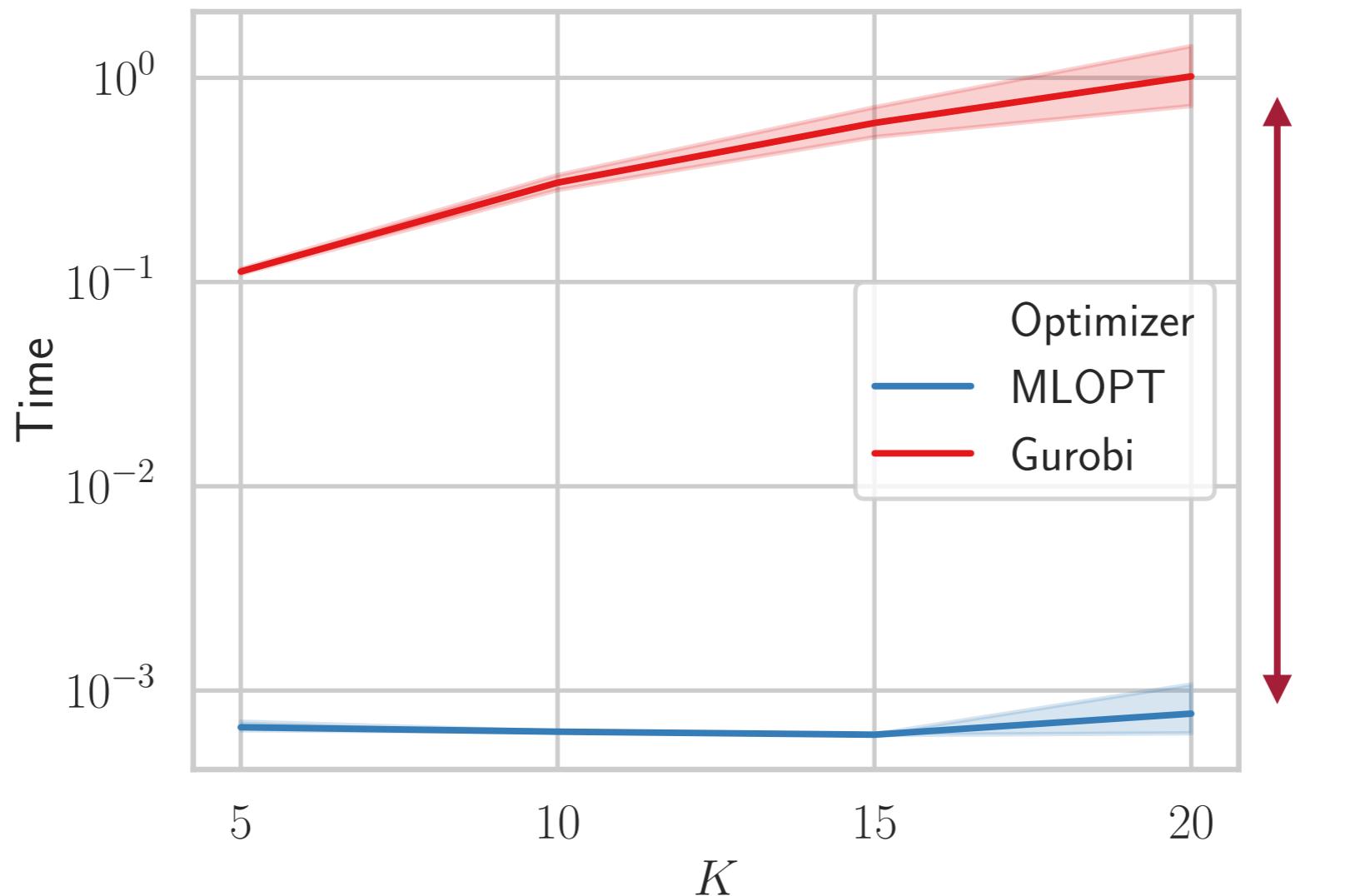


S&P100 backtesting: high speed and accuracy



K	M	acc [%]	suboptimality	infeasibility
5	1027	99.11	9.88×10^{-3}	2.25×10^{-9}
10	1083	99.21	7.01×10^{-3}	8.19×10^{-7}
15	1120	99.30	1.92×10^{-2}	2.68×10^{-7}
20	1209	99.50	3.54×10^{-3}	4.12×10^{-7}

S&P100 backtesting: high speed and accuracy

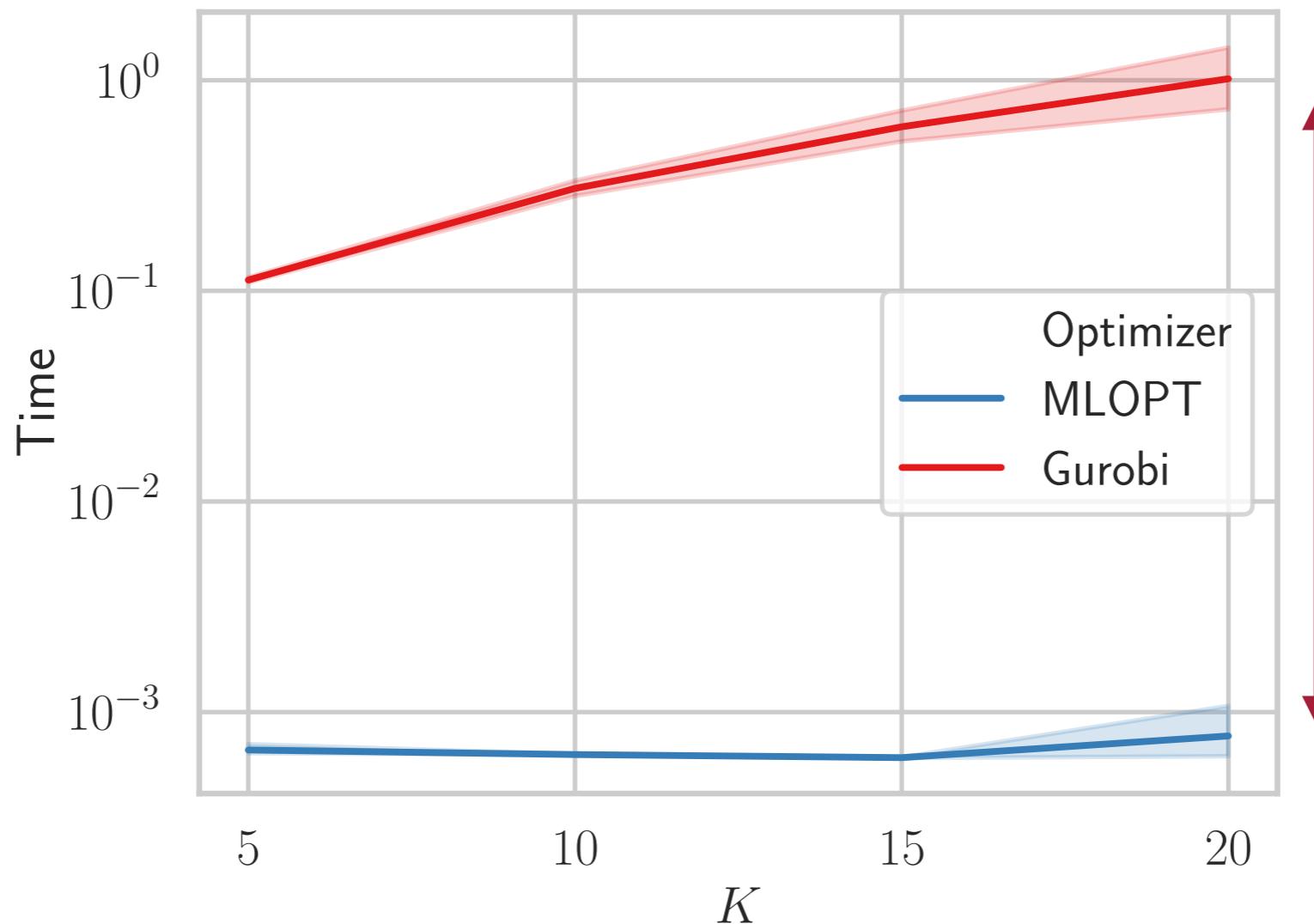


1000x
faster



K	M	acc [%]	suboptimality	infeasibility
5	1027	99.11	9.88×10^{-3}	2.25×10^{-9}
10	1083	99.21	7.01×10^{-3}	8.19×10^{-7}
15	1120	99.30	1.92×10^{-2}	2.68×10^{-7}
20	1209	99.50	3.54×10^{-3}	4.12×10^{-7}

S&P100 backtesting: high speed and accuracy



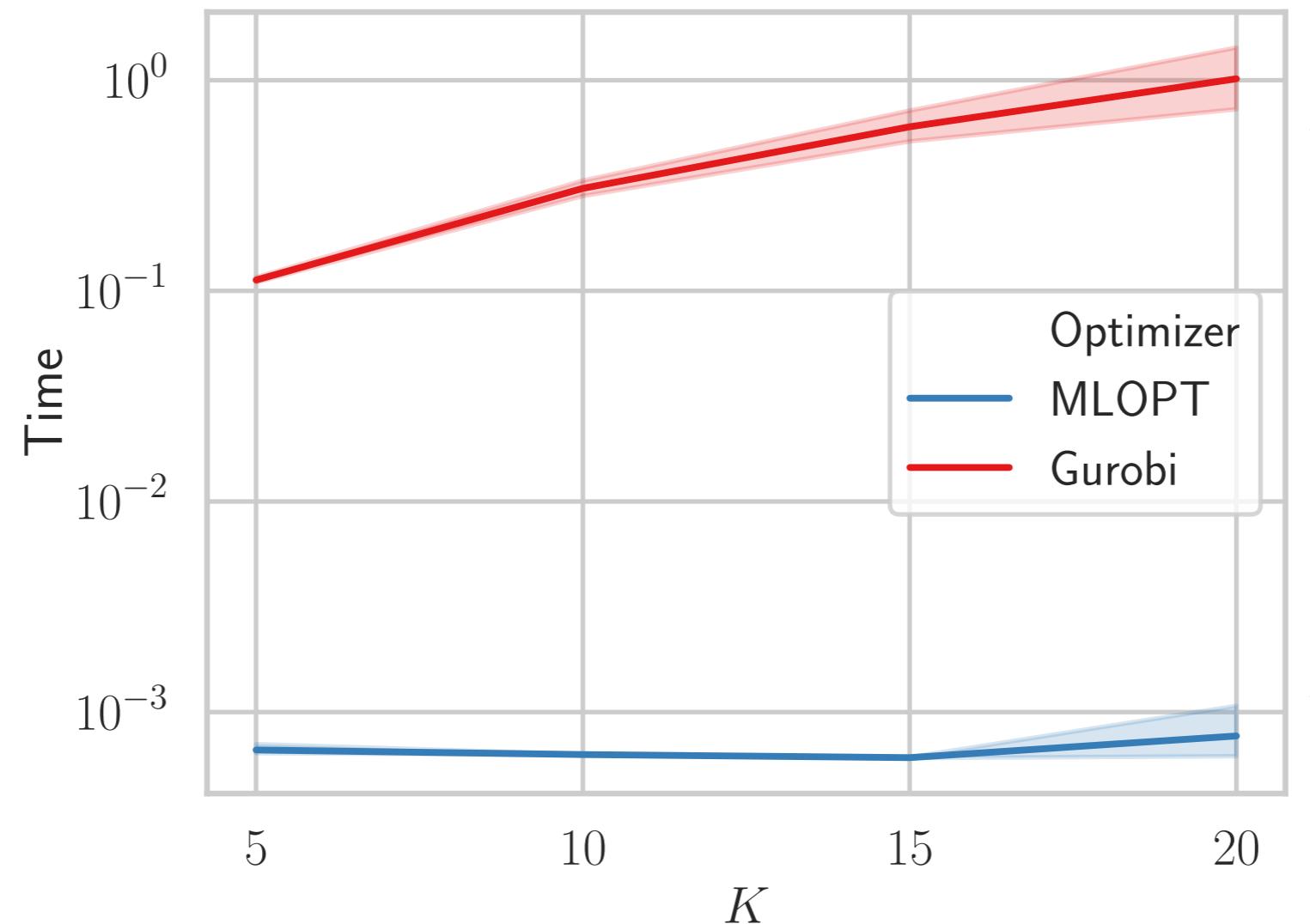
1000x
faster



Low
suboptimality
and
infeasibility

K	M	acc [%]	suboptimality	infeasibility
5	1027	99.11	9.88×10^{-3}	2.25×10^{-9}
10	1083	99.21	7.01×10^{-3}	8.19×10^{-7}
15	1120	99.30	1.92×10^{-2}	2.68×10^{-7}
20	1209	99.50	3.54×10^{-3}	4.12×10^{-7}

S&P100 backtesting: high speed and accuracy



High accuracy

K	M	acc [%]	suboptimality	infeasibility
5	1027	99.11	9.88×10^{-3}	2.25×10^{-9}
10	1083	99.21	7.01×10^{-3}	8.19×10^{-7}
15	1120	99.30	1.92×10^{-2}	2.68×10^{-7}
20	1209	99.50	3.54×10^{-3}	4.12×10^{-7}

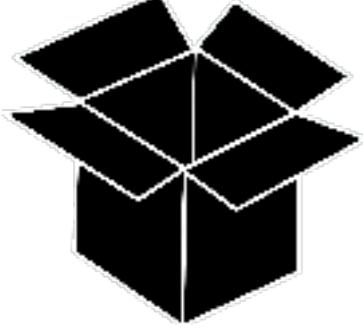
1000x faster



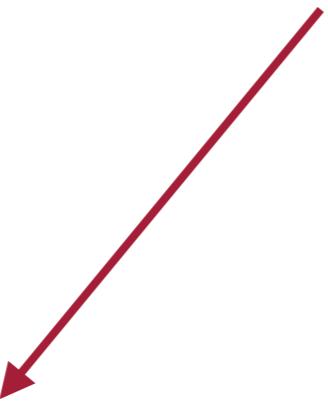
Low suboptimality and infeasibility

Conclusions

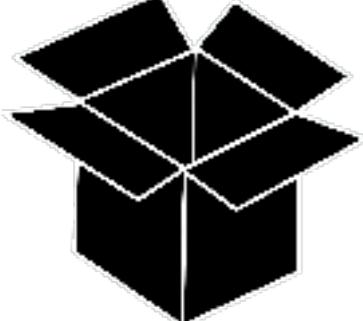
Optimization
links data
to decisions



Optimization
links data
to decisions



Understand
using ML



Optimization
links data
to decisions

Understand
using ML

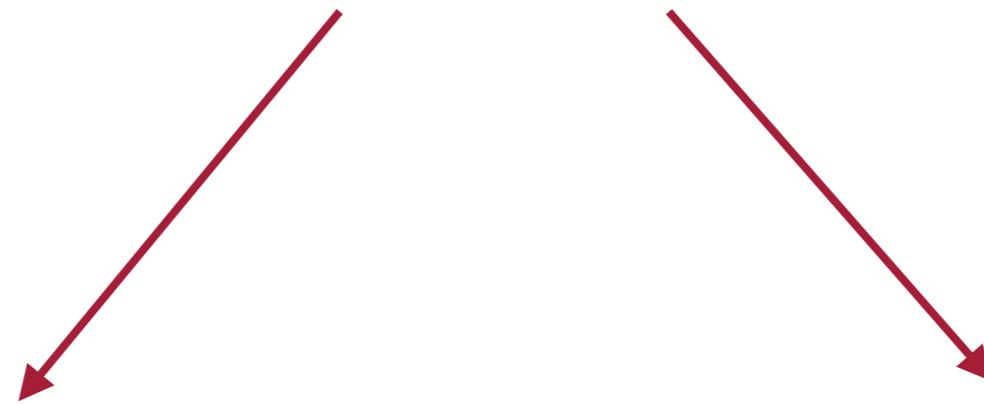
Solve
really fast



Optimization
links data
to decisions



Understand
using ML



Solve
really fast



Bertsimas, D., Stellato, B., *The Voice of Optimization*, arXiv:1812.09991



stellato@mit.edu

Bertsimas, D., Stellato, B., *Online Mixed-Integer Optimization in Milliseconds*, arXiv:1907.02206



@b_stellato

Backup

Sampling scheme

Algorithm 1 Strategies exploration

```
1: given  $\epsilon, \beta, \Theta = \emptyset, \mathcal{S} = \emptyset, u = \infty$ 
2: for  $k = 1, \dots, \text{do}$ 
3:   Sample  $\theta_k$  and compute  $s(\theta_k)$             $\triangleright$  Sample parameter and strategy.
4:    $\Theta \leftarrow \Theta \cup \{\theta_k\}$                    $\triangleright$  Update set of samples.
5:   if  $s(\theta_k) \notin \mathcal{S}$  then
6:      $\mathcal{S} \leftarrow \mathcal{S} \cup \{s(\theta_k)\}$        $\triangleright$  Update strategy set if new strategy found
7:   end if
8:   if  $G + c\sqrt{(1/k) \ln(3/\beta)} \leq \epsilon$  then     $\triangleright$  Break if bound less than  $\epsilon$ 
9:     break
10:  end if
11: end for
12: return  $k, \Theta, \mathcal{S}$ 
```
