A Machine Learning Approach to Optimization

Bartolomeo Stellato

MIT MANAGEMENT SLOAN SCHOOL → PRINCETON UNIVERSITY

ORFE Operations Research & Financial Engineering

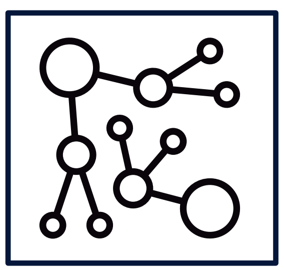
MADDD seminar
UC Davis
June 9, 2020



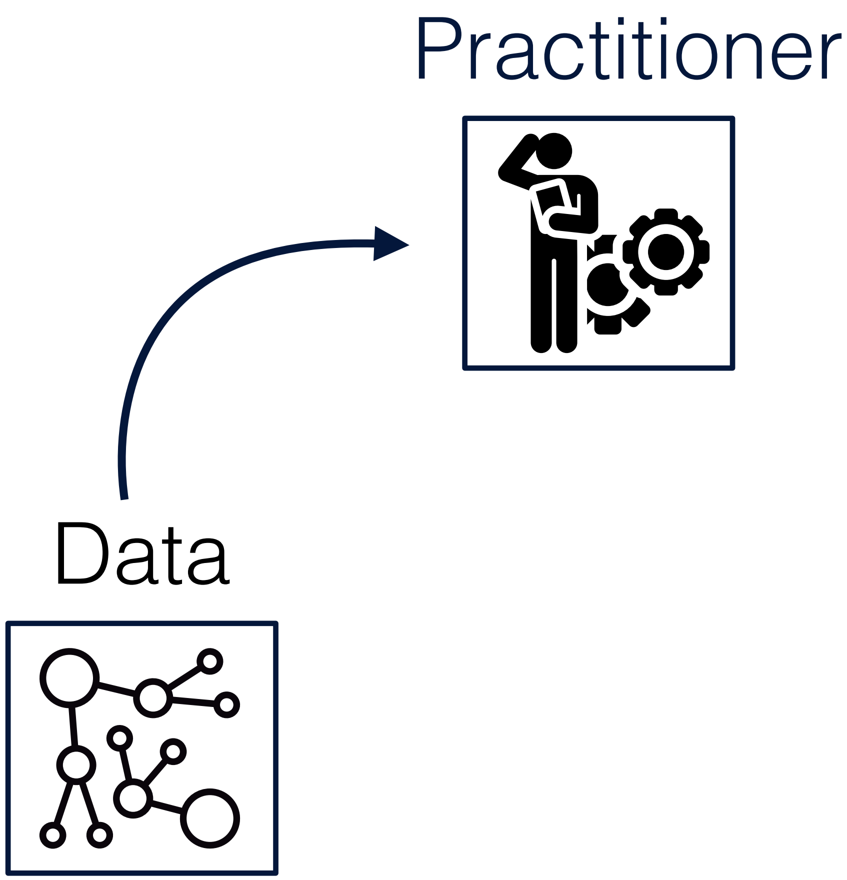
TIMES
DOW JONES

The decision-making workflow

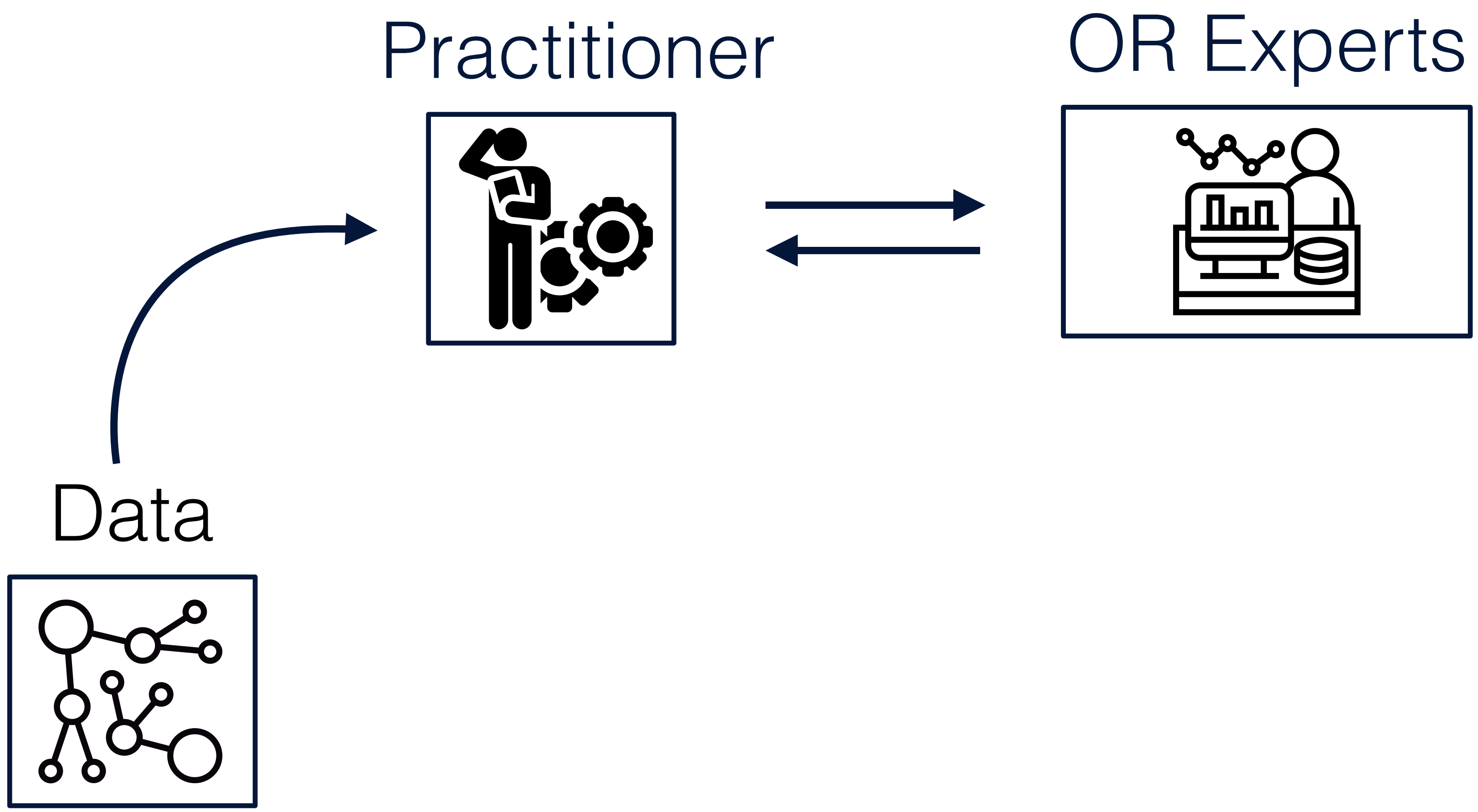
Data



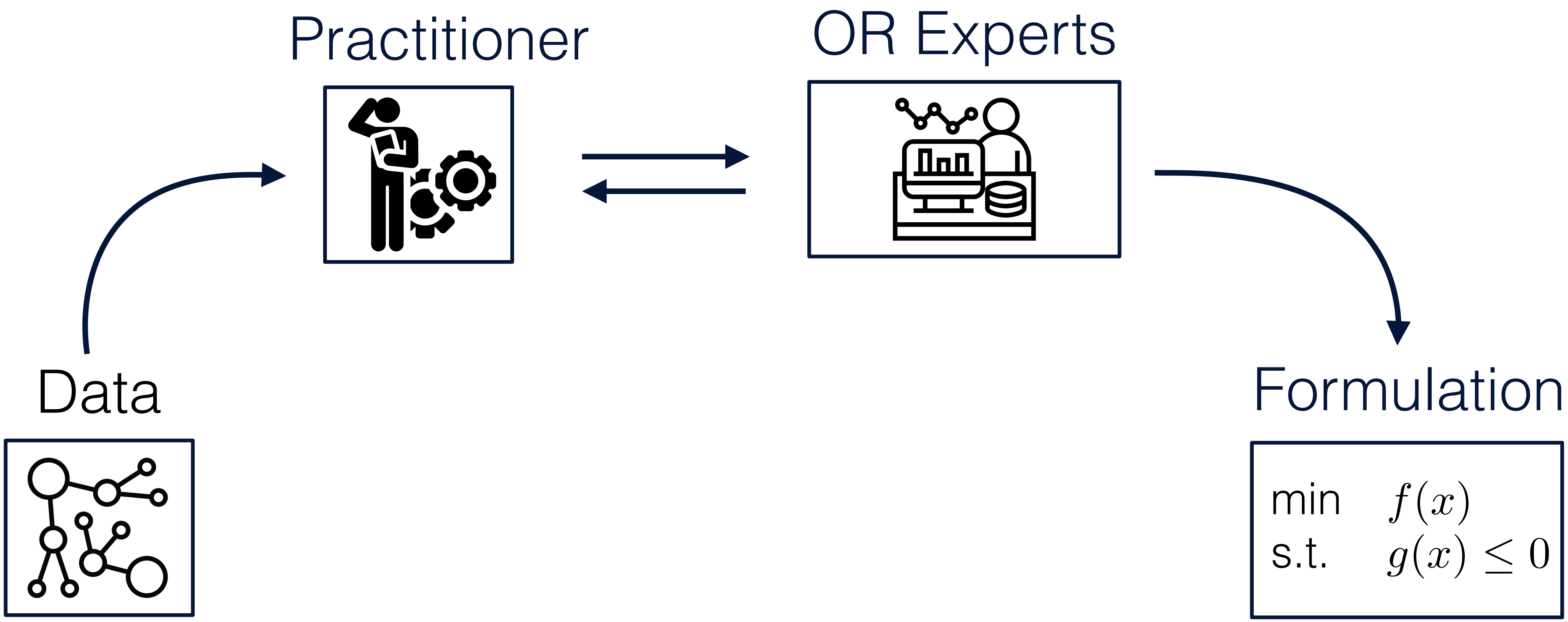
The decision-making workflow



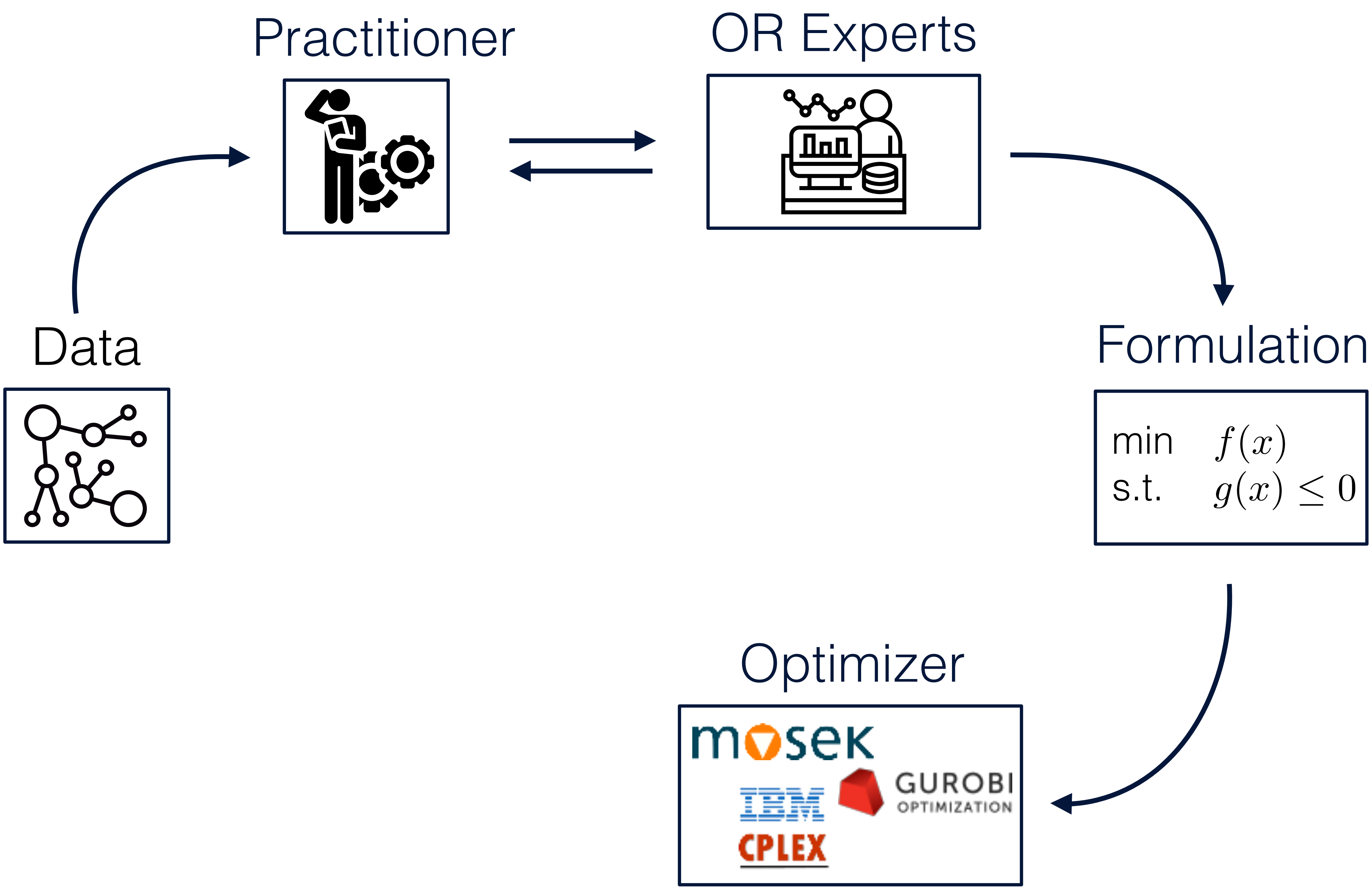
The decision-making workflow



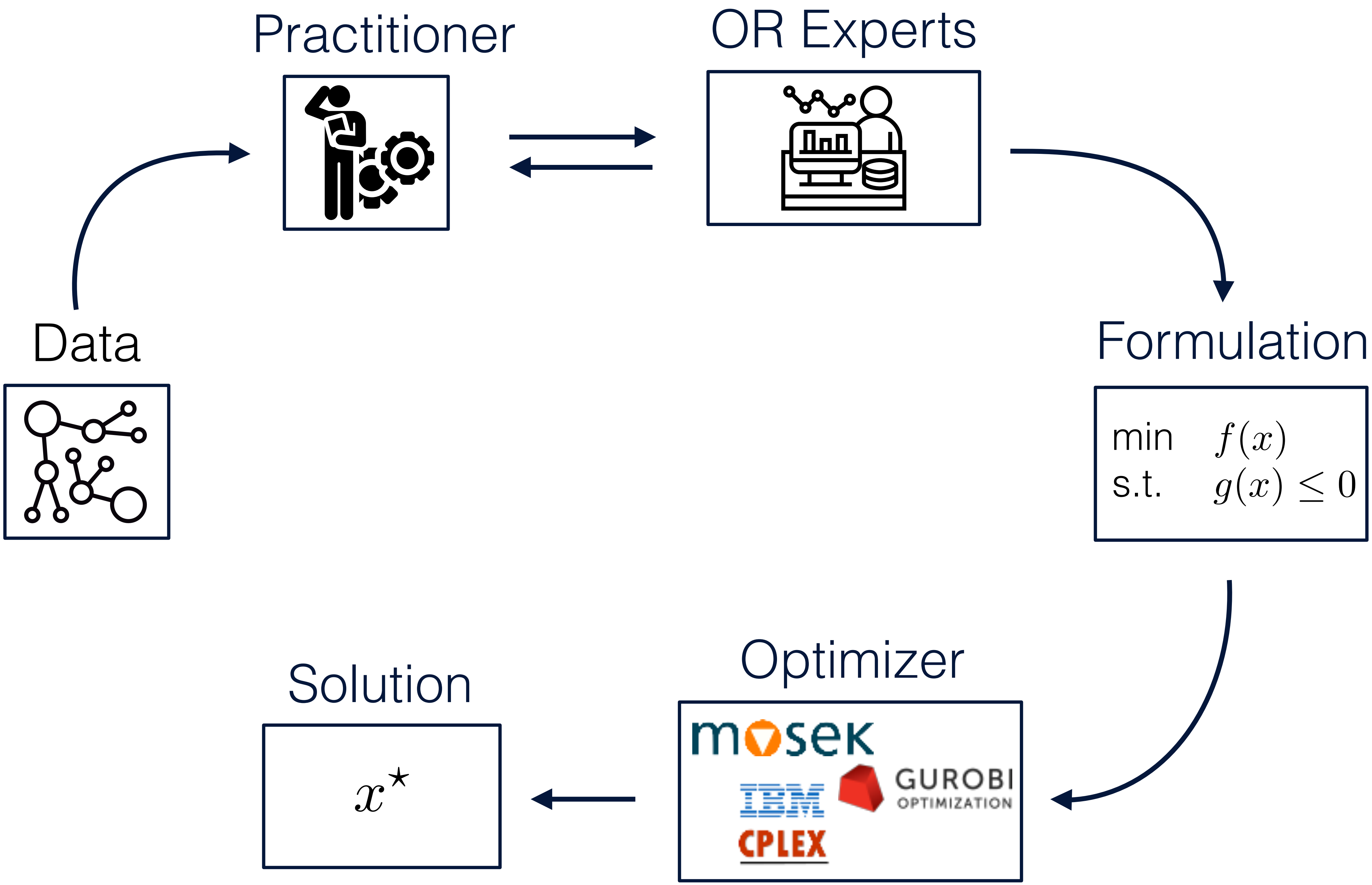
The decision-making workflow



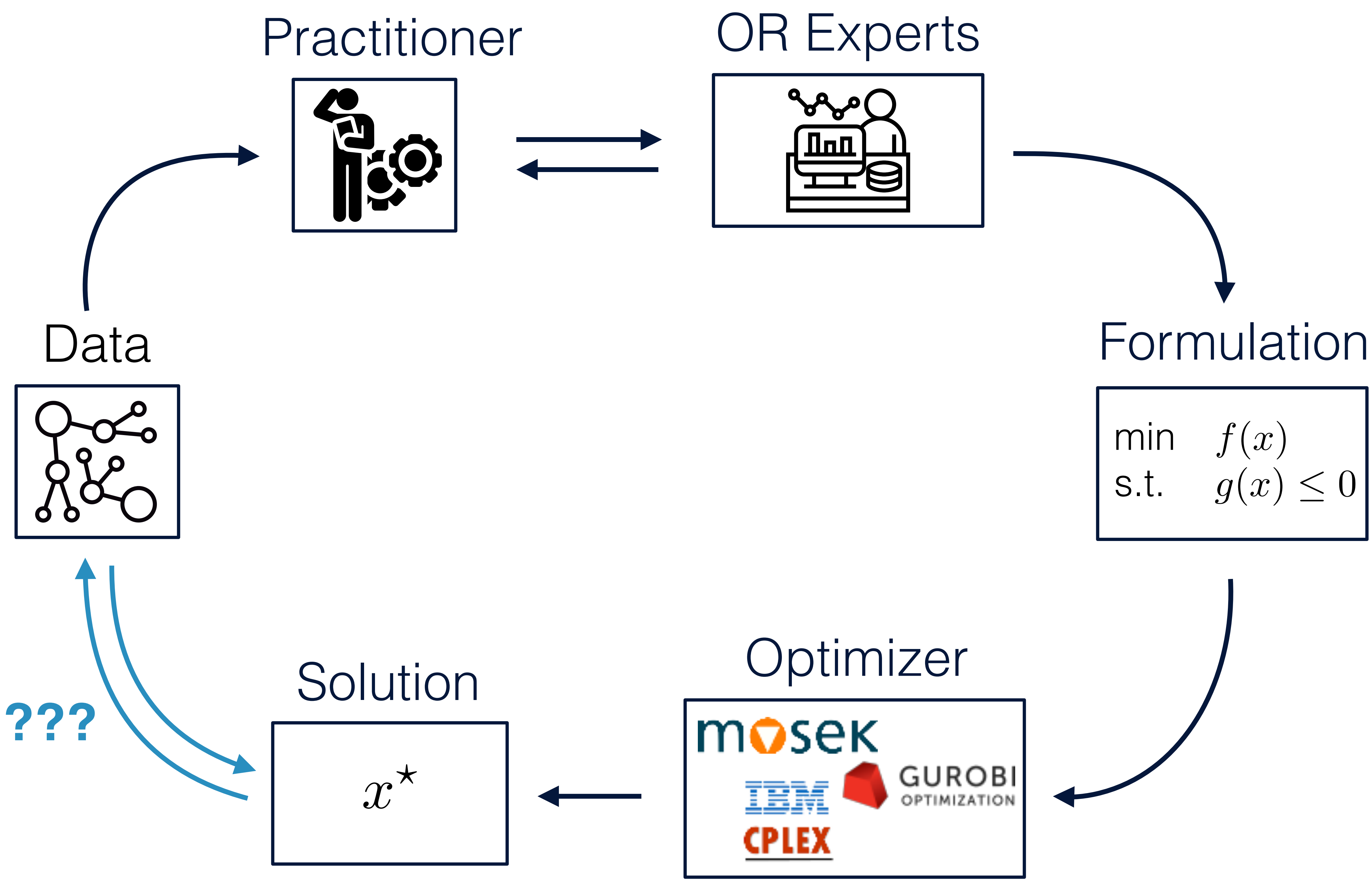
The decision-making workflow



The decision-making workflow



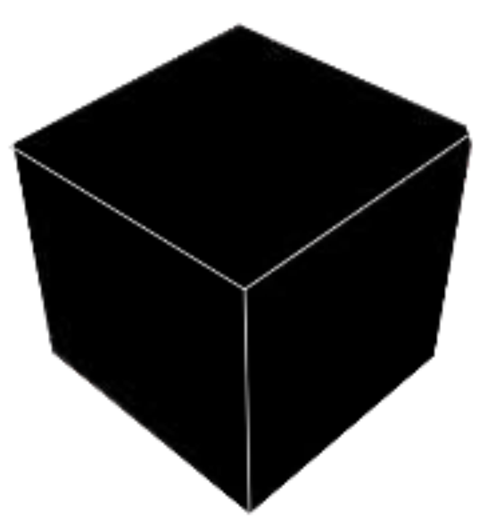
The decision-making workflow



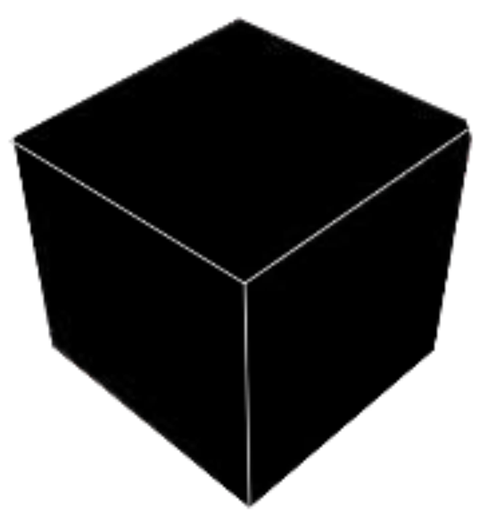
Limitations of traditional optimization

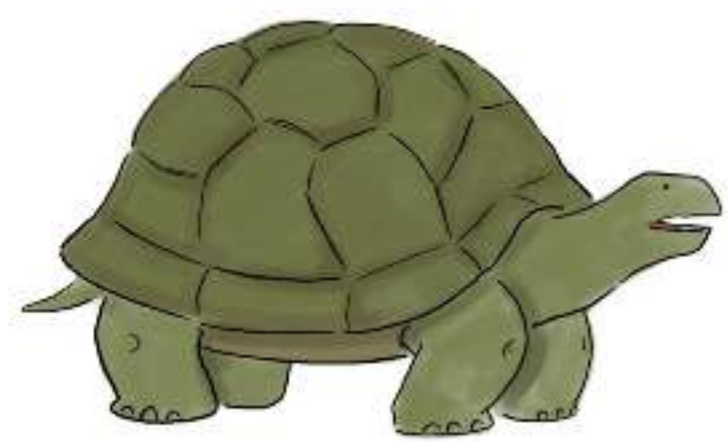
Limitations of traditional optimization

$\theta \longrightarrow$ **Optimization** $\longrightarrow x^\star$



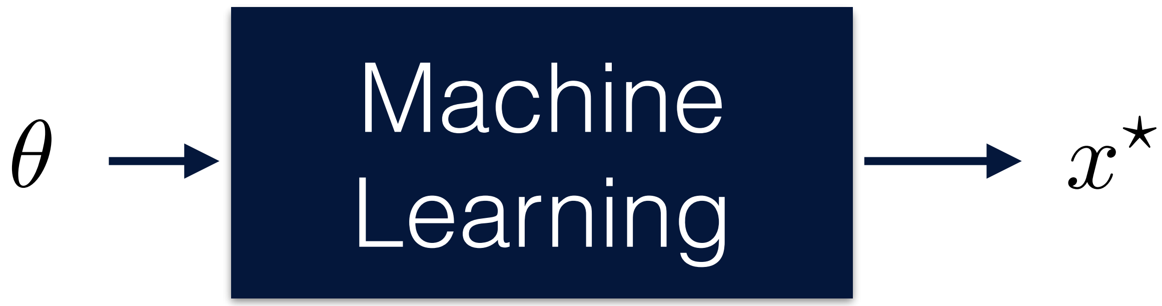
Limitations of traditional optimization





Existing methods

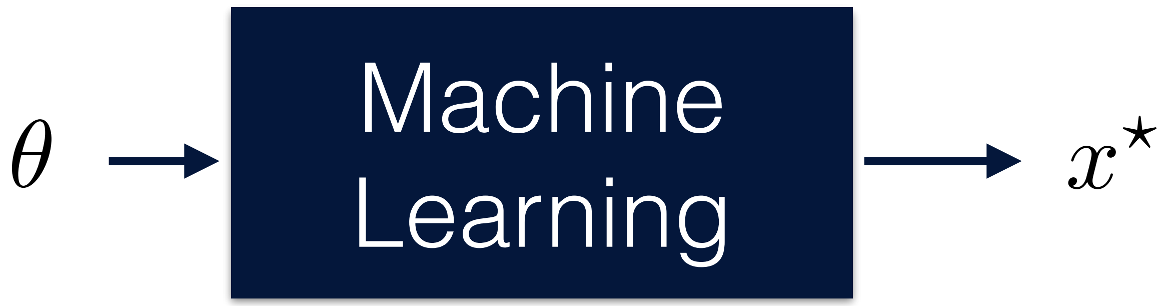
Machine learning for optimization



Comb. Opt: [Smith (1999)]
TSP: [Vinyals et al. (2017)]

Existing methods

Machine learning for optimization



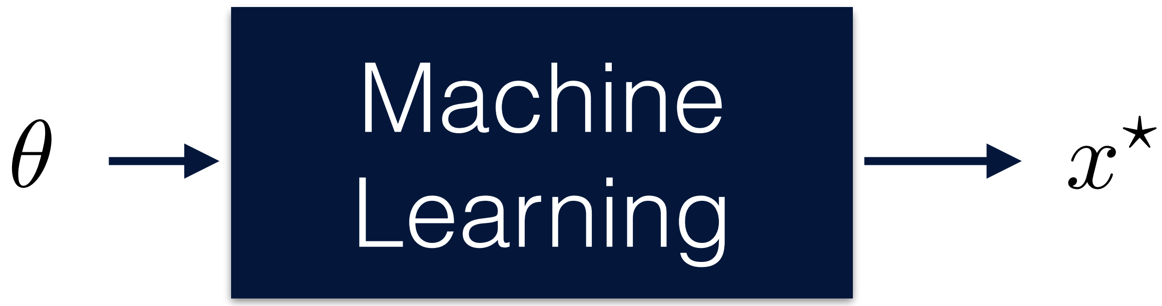
Comb. Opt: [Smith (1999)]
TSP: [Vinyals et al. (2017)]

Very small problems

Imprecise

Existing methods

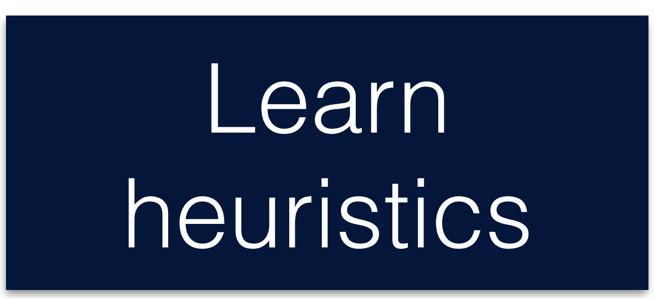
Machine learning for optimization



Very small problems

Imprecise

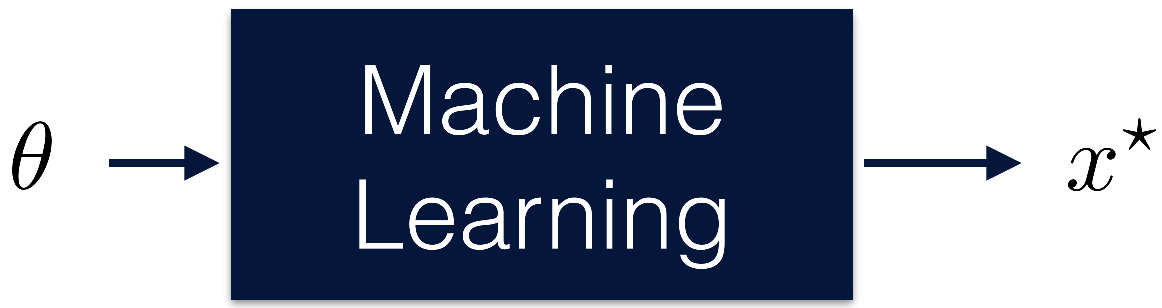
Comb. Opt: [Smith (1999)]
TSP: [Vinyals et al. (2017)]



B&B: [Khalil et al. (2016)]
Decompositions [Bonami et al. (2018)]
TSP: [Khalil et al. (2017)]
Cuts: [Tang et a. (2020)]
Many more…

Existing methods

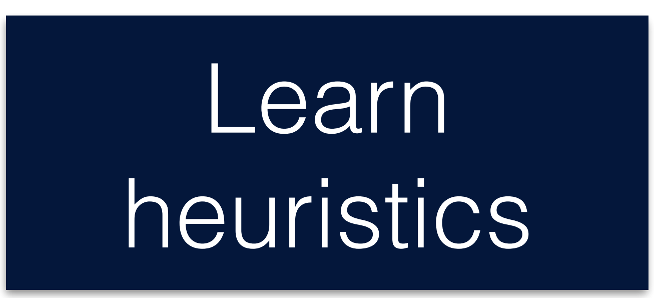
Machine learning for optimization



Very small problems

Imprecise

Comb. Opt: [Smith (1999)]
TSP: [Vinyals et al. (2017)]

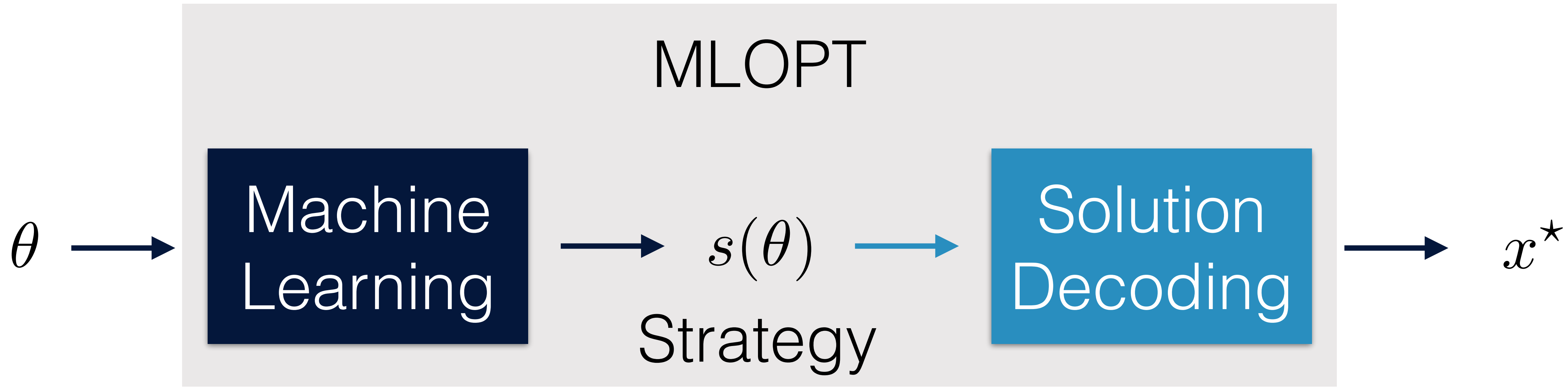


Still hard to solve

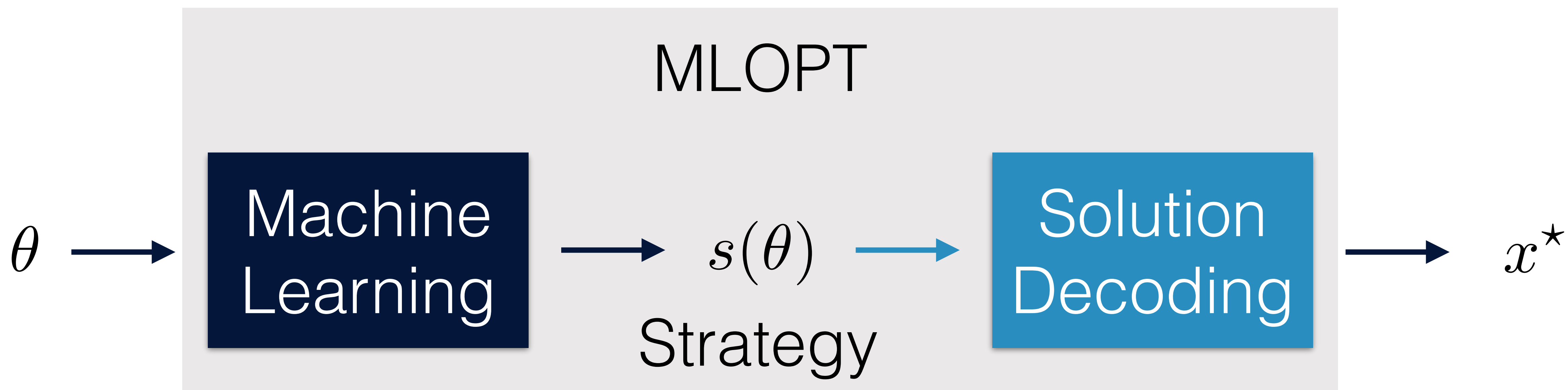
No parametric problems

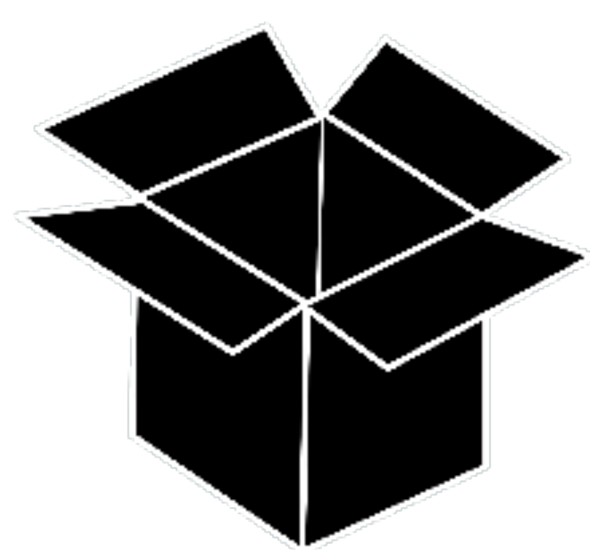
B&B: [Khalil et al. (2016)]
Decompositions [Bonami et al. (2018)]
TSP: [Khalil et al. (2017)]
Cuts: [Tang et a. (2020)]
Many more…

Machine learning optimizer

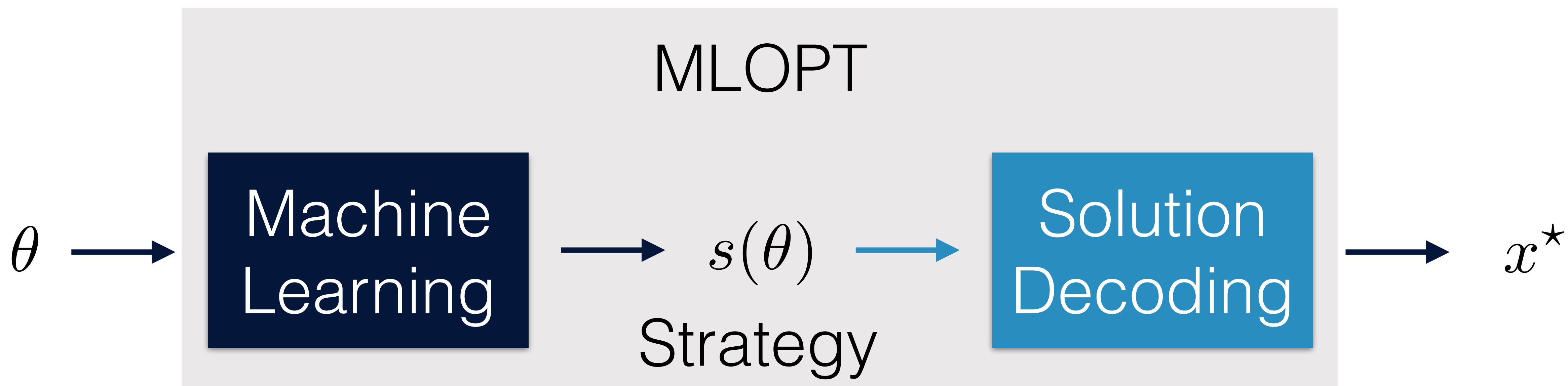


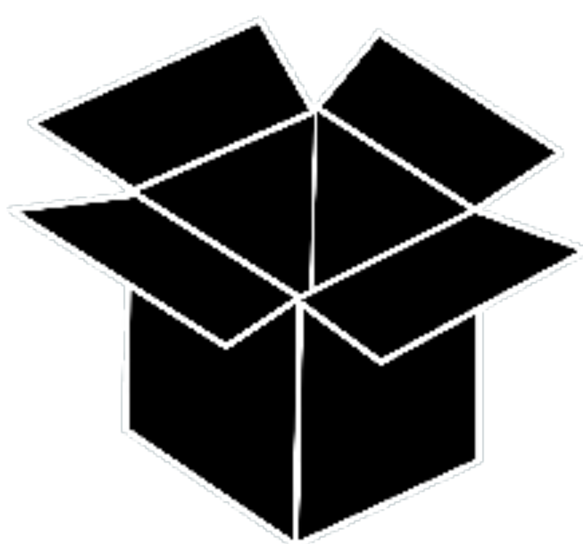
Machine learning optimizer





Machine learning optimizer





Machine learning optimizer

Optimal Strategies

Strategy Prediction

Strategy Exploration

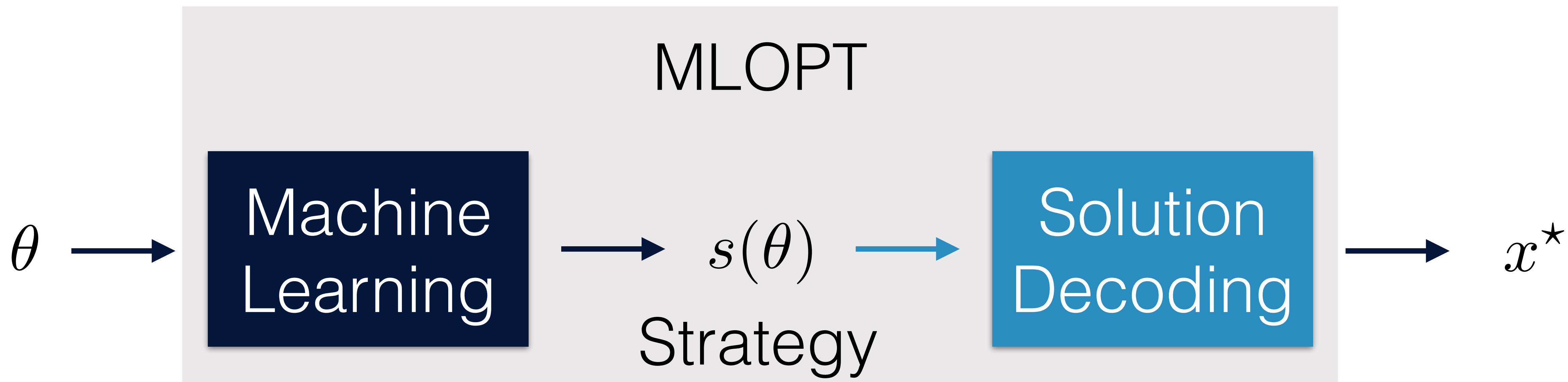
Speedups

Examples

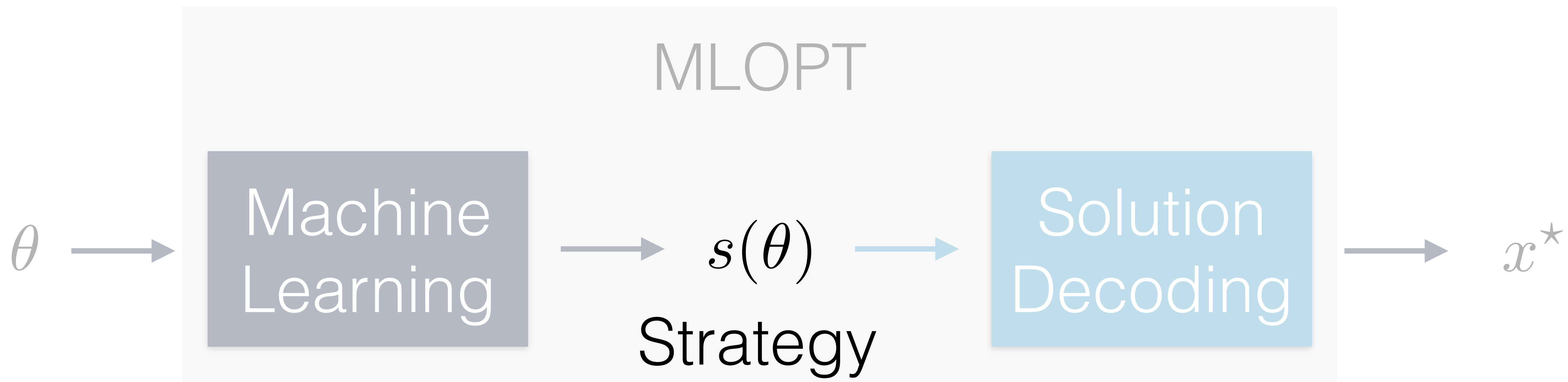
Machine learning optimizer

- Optimal Strategies
- Strategy Prediction
- Strategy Exploration
- Speedups
- Examples

Machine learning optimizer



Machine learning optimizer

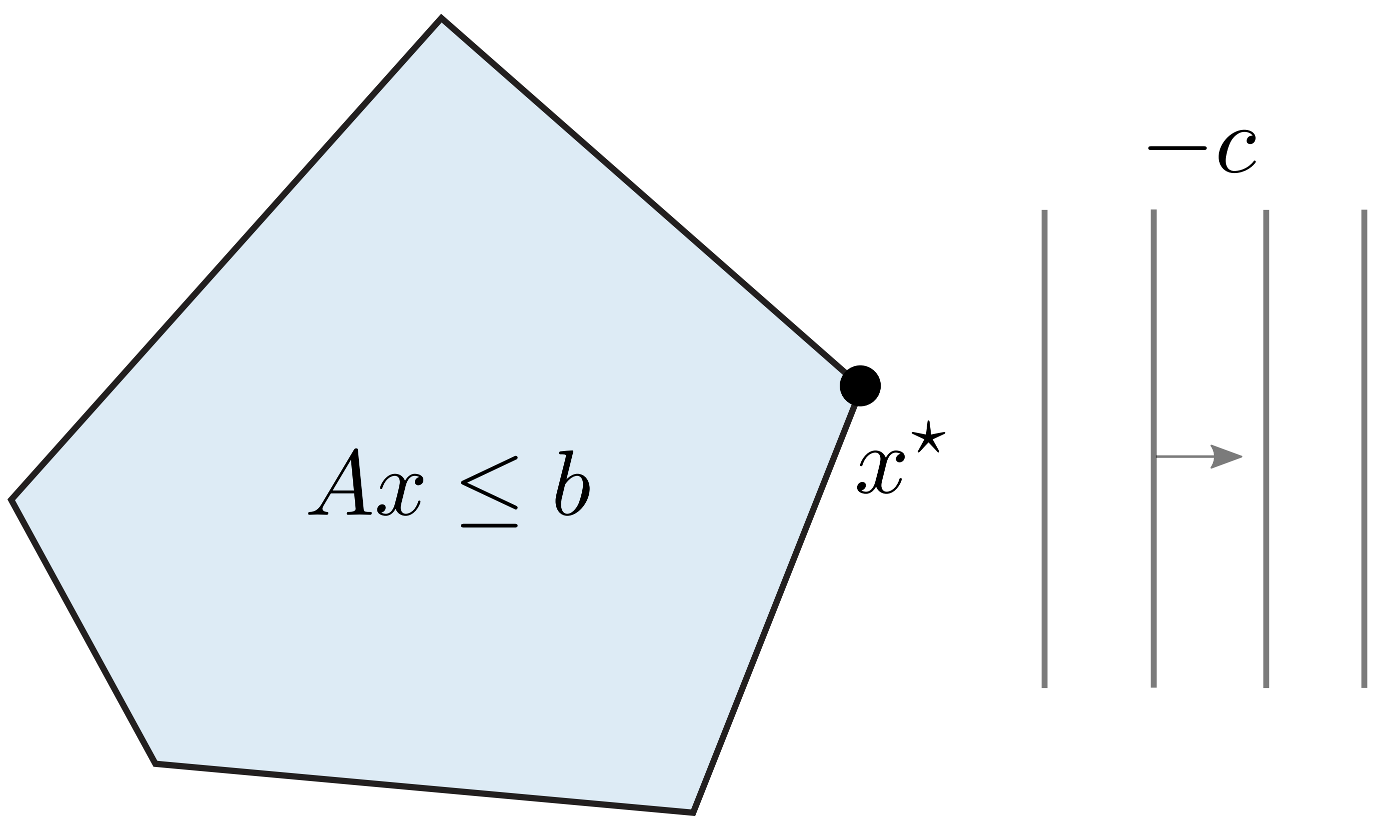


The complete information to efficiently get the optimal solution

$$s(\theta)$$

Linear optimization

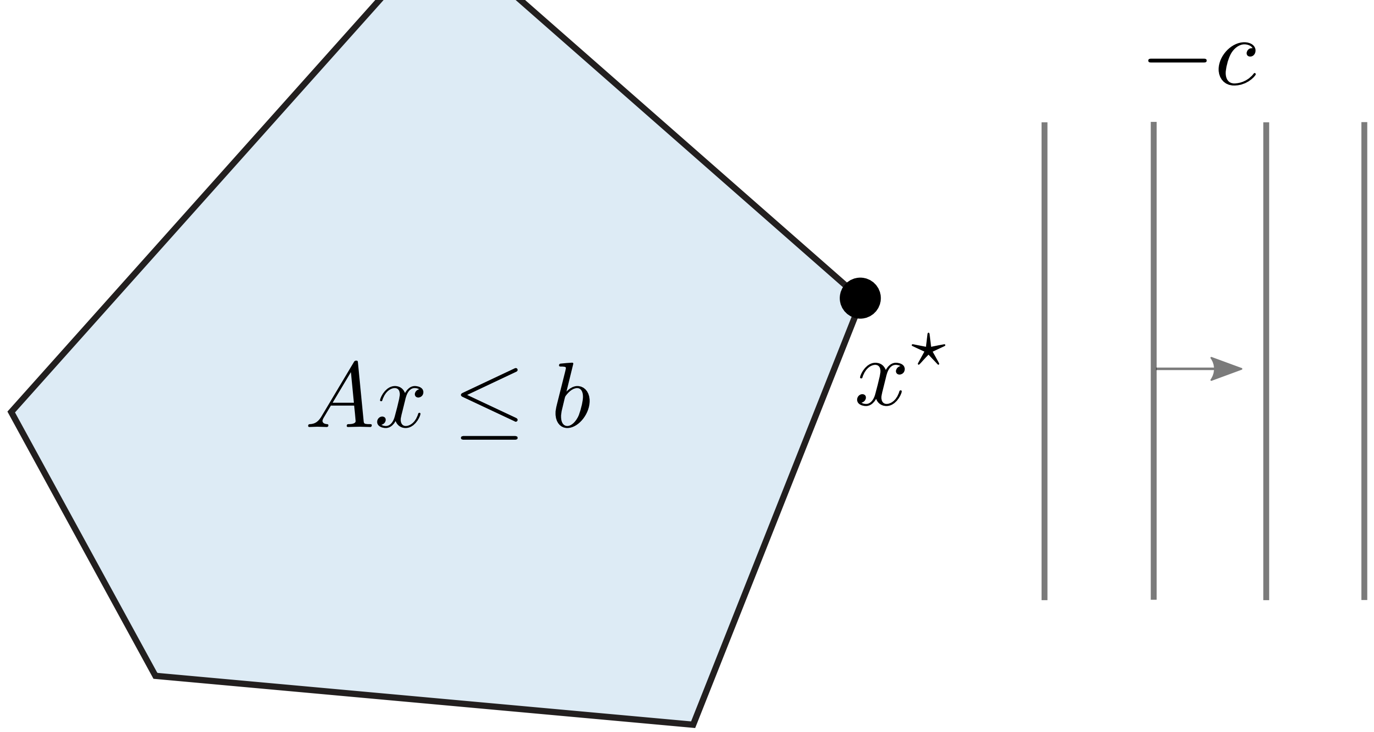
$$\begin{array}{ll} \text{minimize} & c(\theta)^T x \\ \text{subject to} & A(\theta)x \leq b(\theta) \end{array}$$



Linear optimization

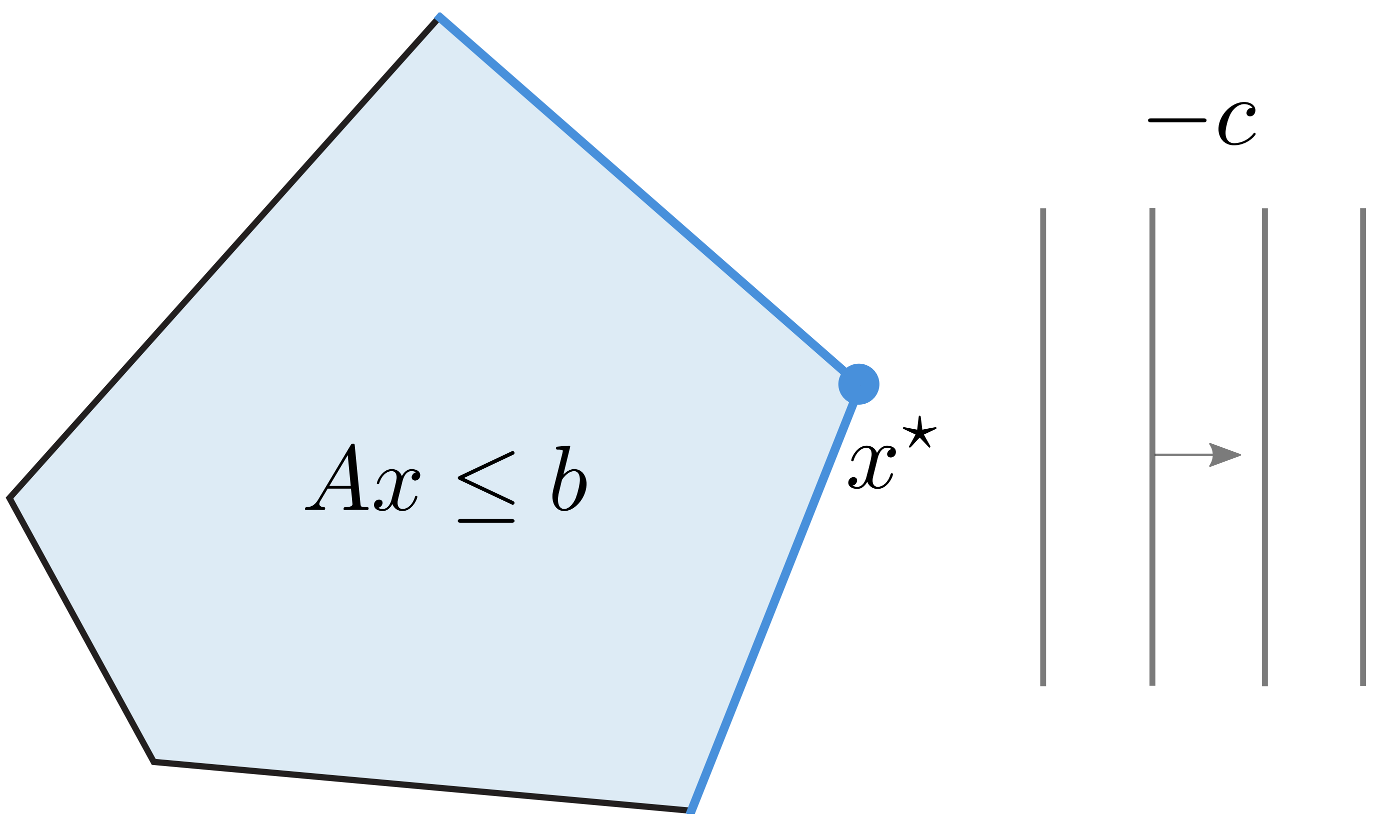
$$\begin{array}{ll} \text{minimize} & c(\theta)^T x \\ \text{subject to} & A(\theta)x \leq b(\theta) \end{array}$$

Parameters



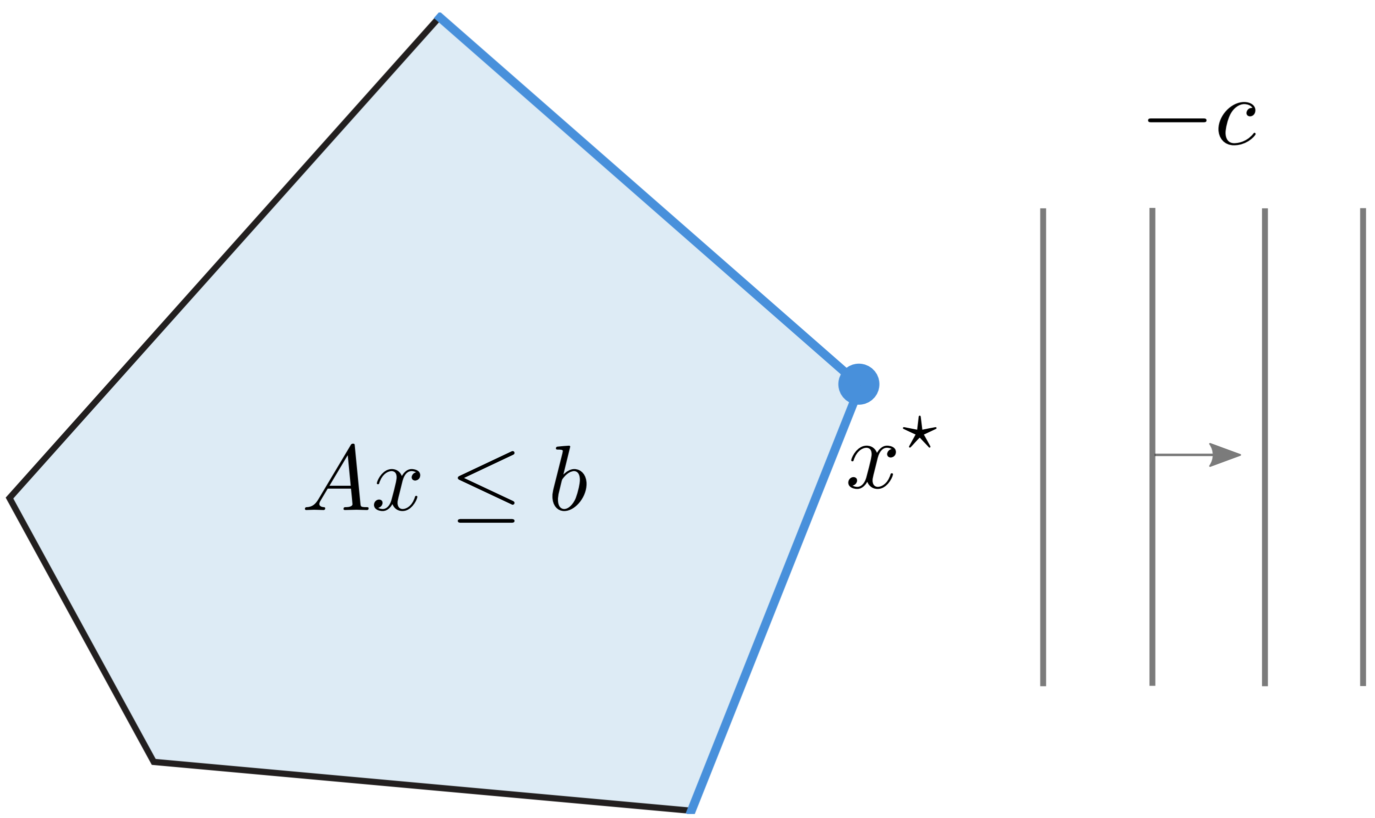
Tight constraints

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



Tight constraints

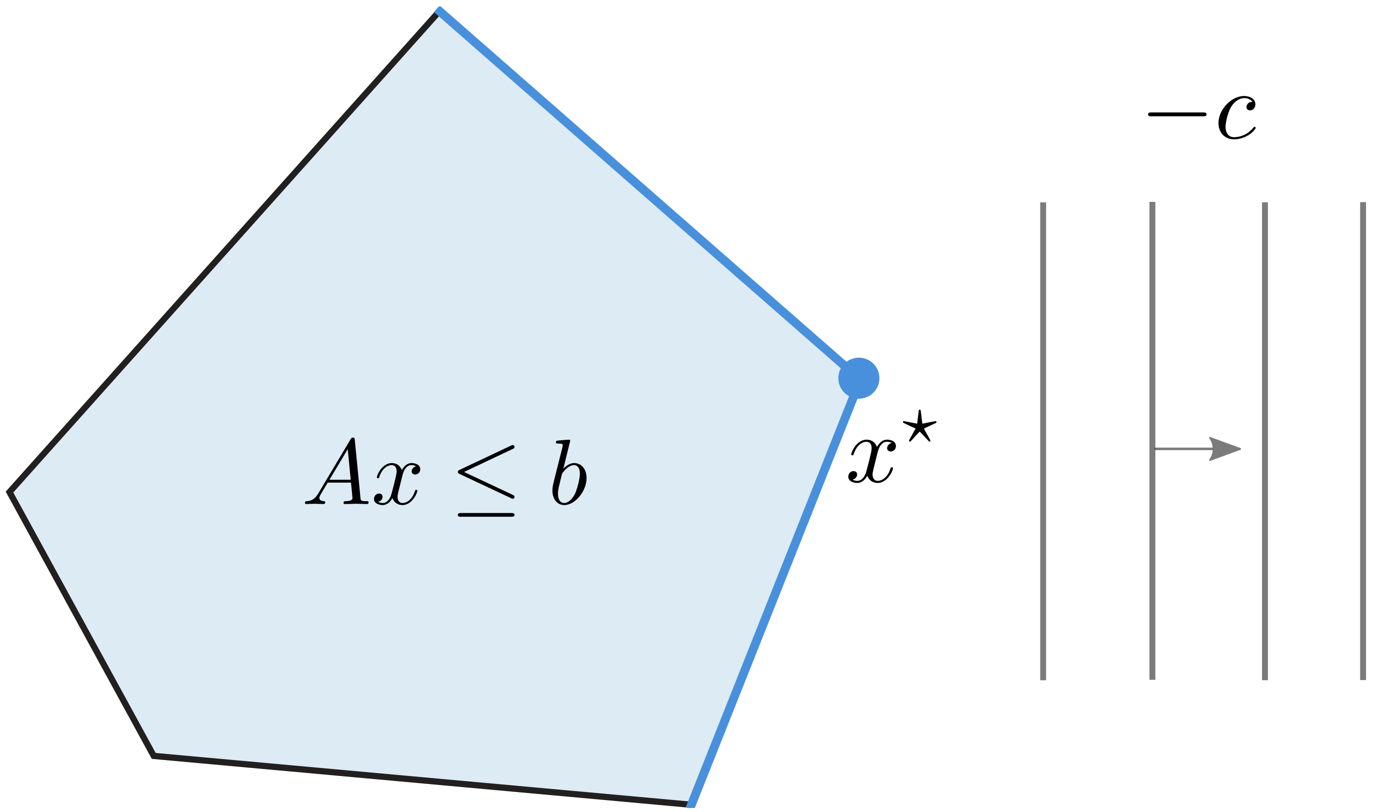
$$\mathcal{T}(\theta) = \{i \mid A_i(\theta)x^\star = b_i(\theta)\}$$



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

Tight constraints

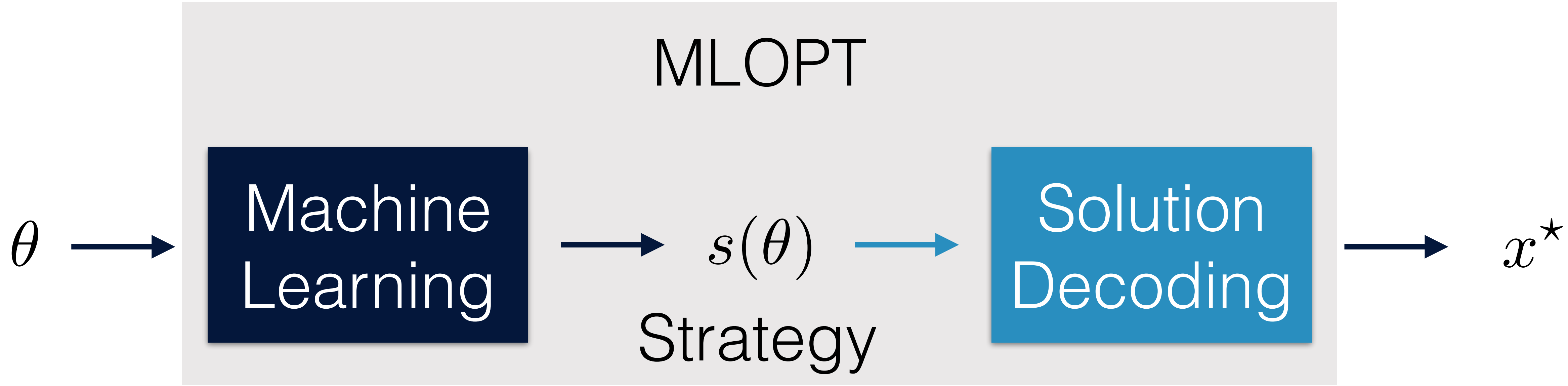
$$\mathcal{T}(\theta) = \{i \mid A_i(\theta)x^\star = b_i(\theta)\}$$



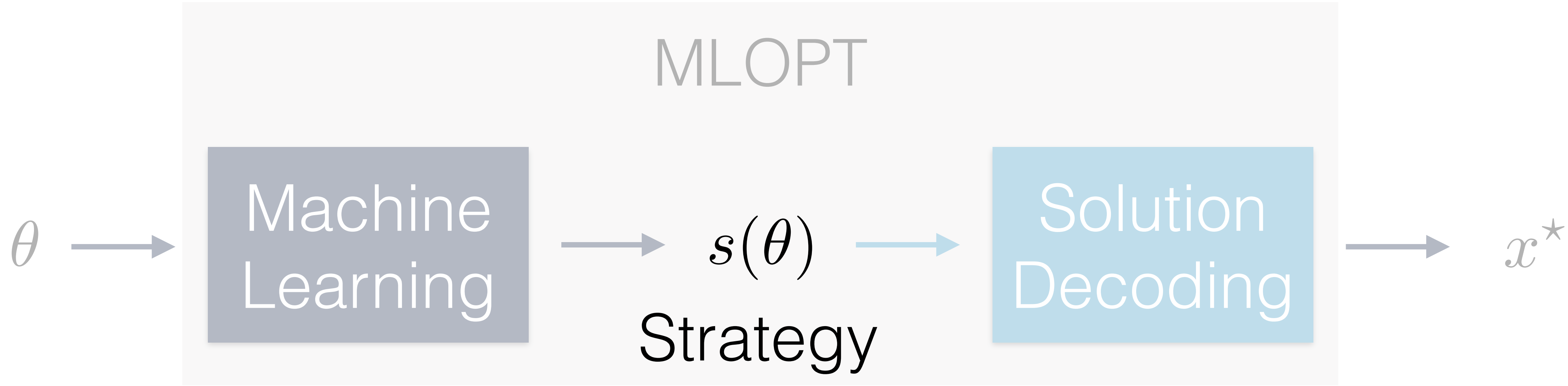
$|\mathcal{T}(\theta)| = \#$ variables if non-degenerate

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

Strategies for linear optimization



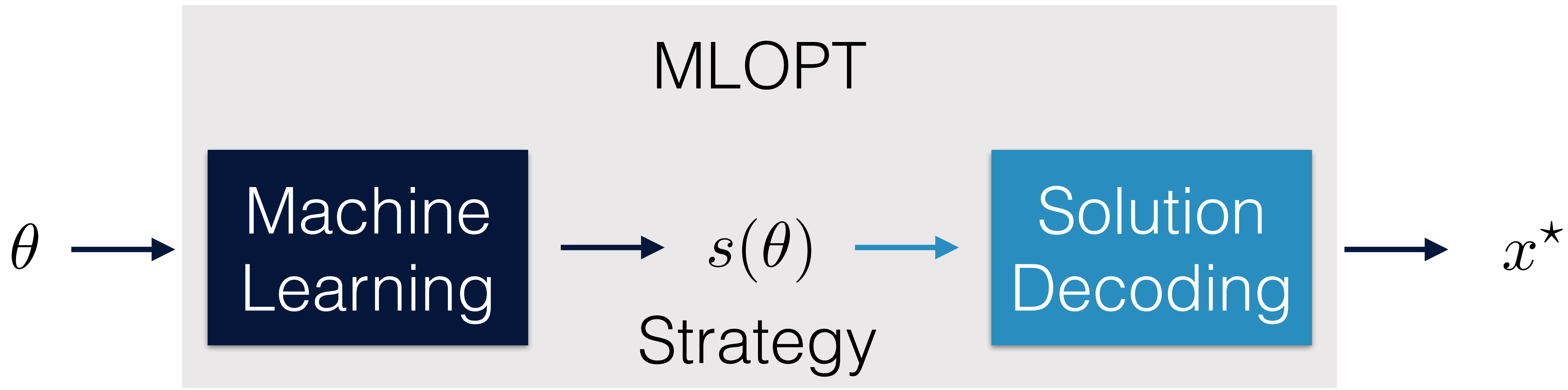
Strategies for linear optimization



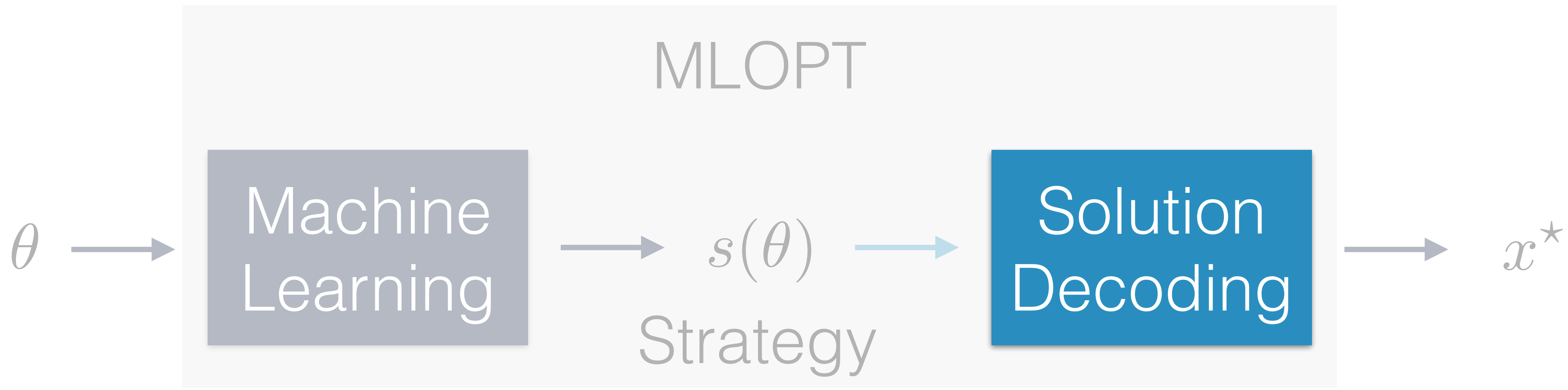
The complete information to efficiently get the optimal solution

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

Computing the solution from the optimal strategy



Computing the solution from the optimal strategy



Computing the solution from the optimal strategy

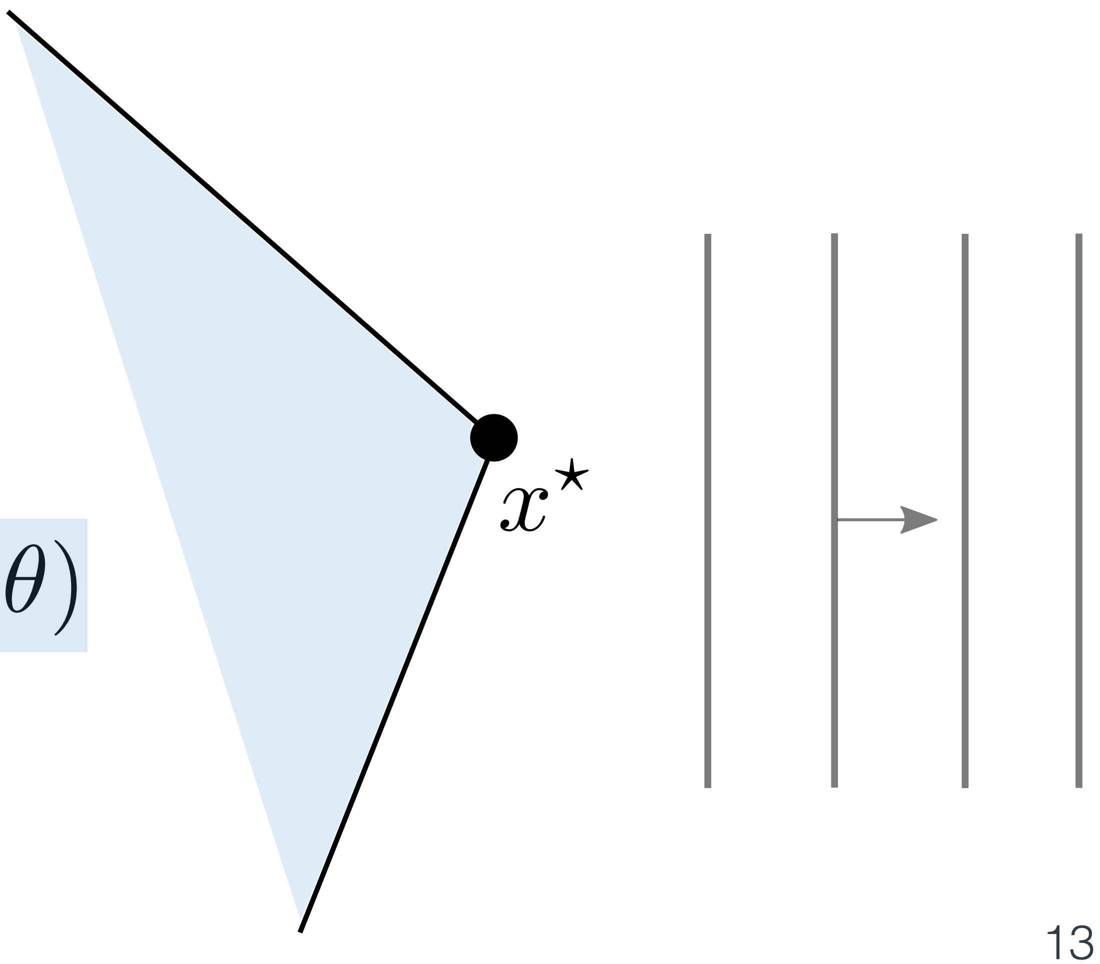
$$\begin{array}{ll} \text{minimize} & c(\theta)^T x \\ \text{subject to} & A(\theta)x \leq b(\theta) \end{array}$$

Computing the solution from the optimal strategy

minimize $\quad c(\theta)^T x$
subject to $\quad A(\theta)x \leq b(\theta)$

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

minimize $\quad c(\theta)^T x$
subject to $\quad A_i(\theta)x = b_i(\theta) \quad \forall i \in \mathcal{T}(\theta)$



Inventory management

$$
\begin{aligned}
\text{minimize} \quad & \sum_{t=0}^{T-1} h(x_t) + o(u_t) \\
\text{subject to} \quad & x_{t+1} = x_t + u_t - d_t \\
& x_0 = x_{\text{init}} \\
& 0 \leq u_t \leq M
\end{aligned}
$$

Inventory management

$$\begin{array}{ll} \text{minimize} & \sum_{t=0}^{T-1} h(x_t) + o(u_t) \\ \text{subject to} & x_{t+1} = x_t + u_t - d_t \\ & x_0 = x_{\text{init}} \\ & 0 \leq u_t \leq M \end{array}$$

Inventory

Inventory management

$$
\begin{aligned}
\text{minimize} \quad & \sum_{t=0}^{T-1} h(x_t) + o(u_t) \\
\text{subject to} \quad & x_{t+1} = x_t + u_t - d_t \\
& x_0 = x_{\text{init}} \\
& 0 \leq u_t \leq M
\end{aligned}
$$

Inventory

Order

Inventory management

$$
\begin{aligned}
\text{minimize} \quad & \sum_{t=0}^{T-1} h(x_t) + o(u_t) \\
\text{subject to} \quad & x_{t+1} = x_t + u_t - d_t \\
& x_0 = x_{\text{init}} \\
& 0 \leq u_t \leq M
\end{aligned}
$$

Inventory (x_{t+1})

Order (u_t)

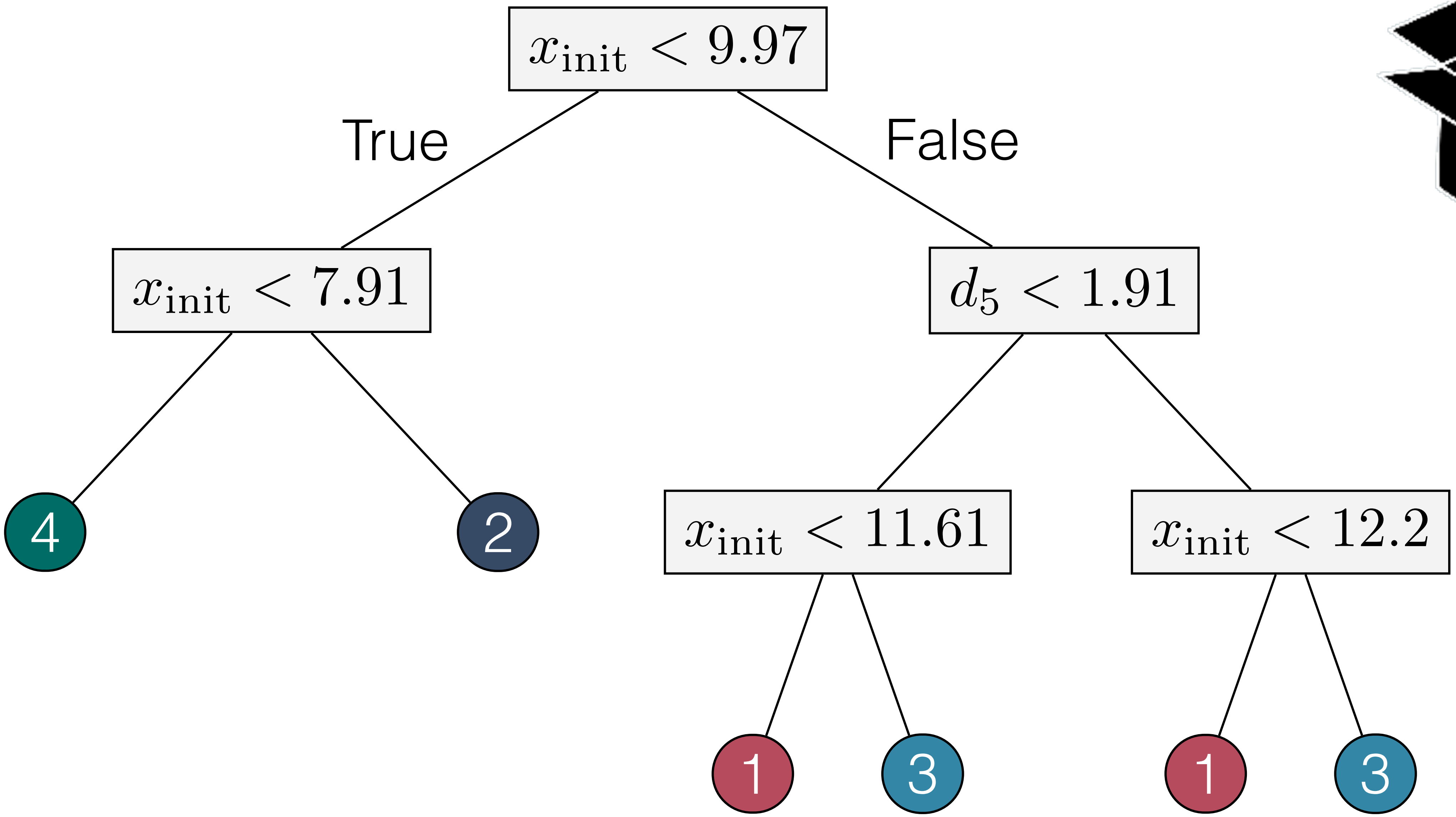
Demand (d_t)

Inventory management

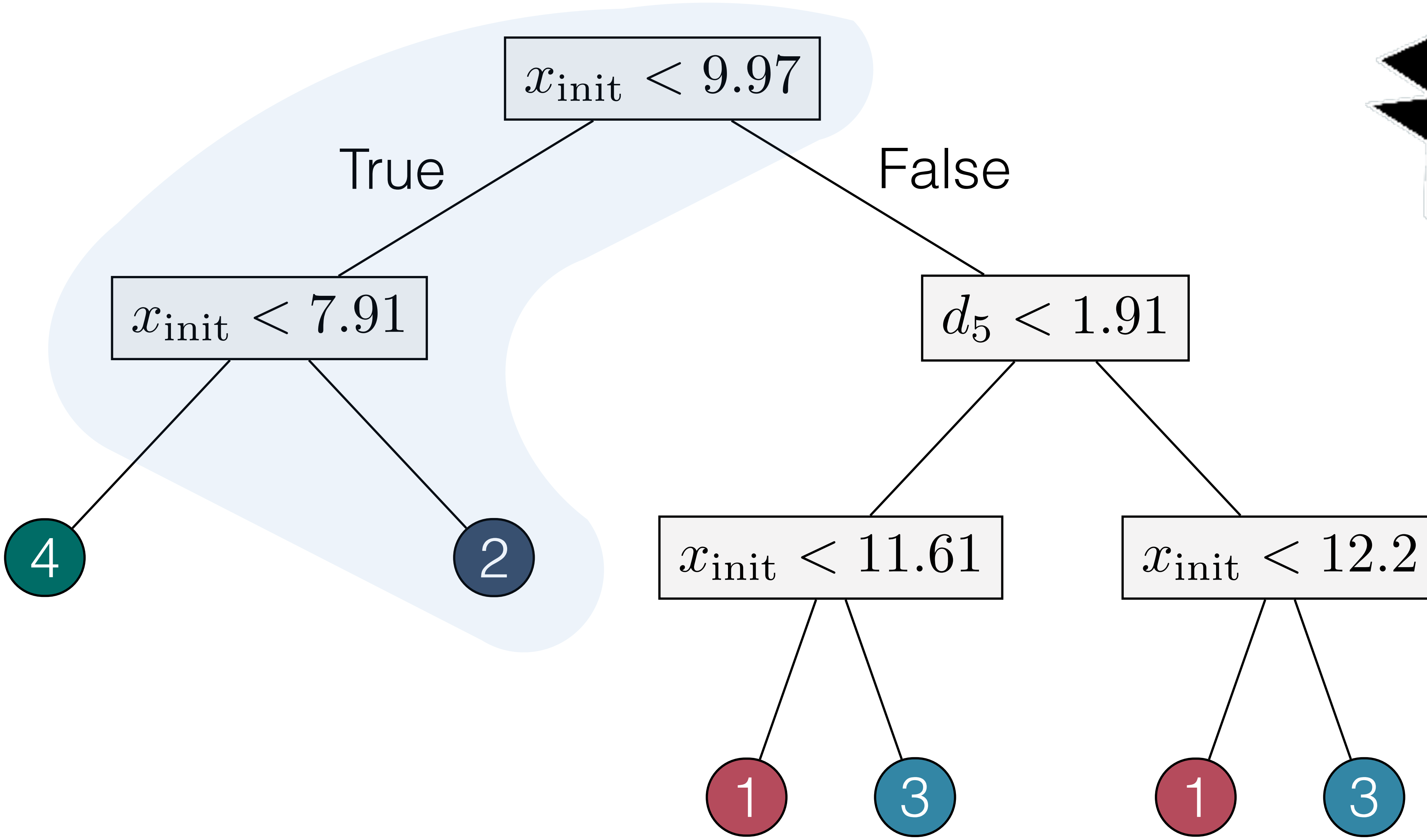
$$\begin{aligned} \text{minimize} \quad & \sum_{t=0}^{T-1} h(x_t) + o(u_t) \\ \text{subject to} \quad & x_{t+1} = x_t + u_t - d_t \\ & x_0 = x_{\text{init}} \\ & 0 \leq u_t \leq M \end{aligned}$$

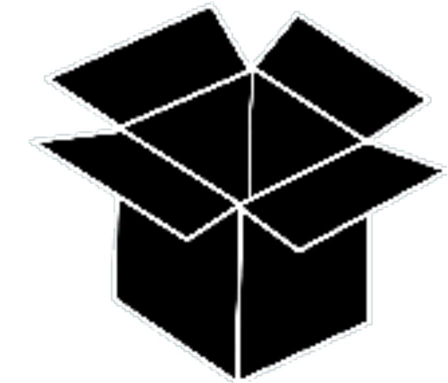
Parameters

Strategy for inventory management



Strategy for inventory management

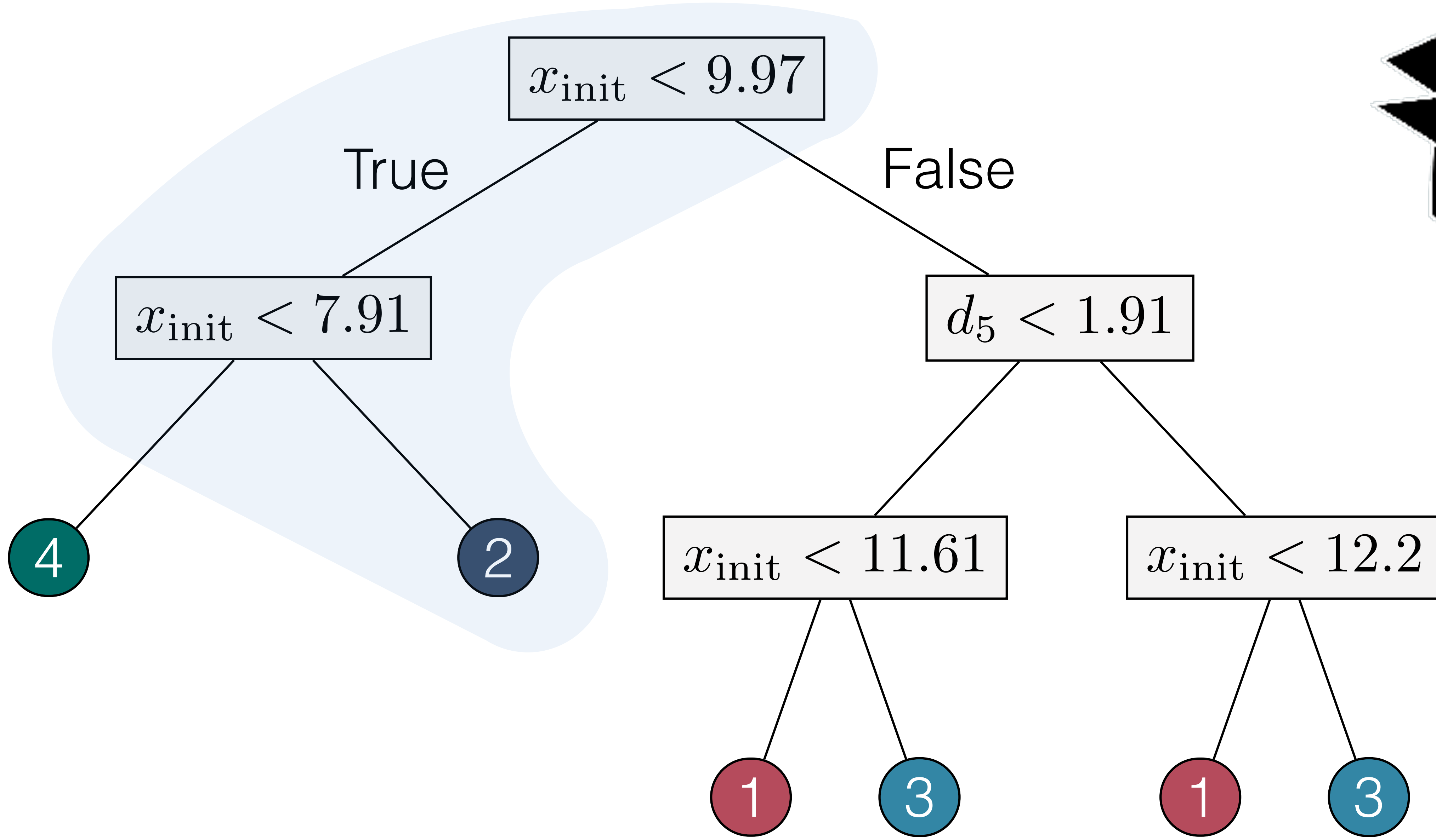


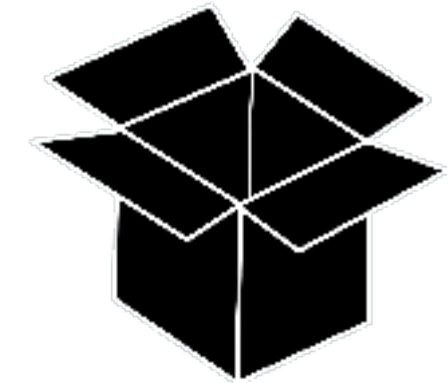


Strategy 2

$$u_t = 0 \qquad t \leq 4$$
$$0 \leq u_t \leq M \qquad t > 4$$

Strategy for inventory management





Strategy 4

$$u_t = 0 \qquad t \leq 3$$
$$0 \leq u_t \leq M \qquad t > 3$$

Strategy 2

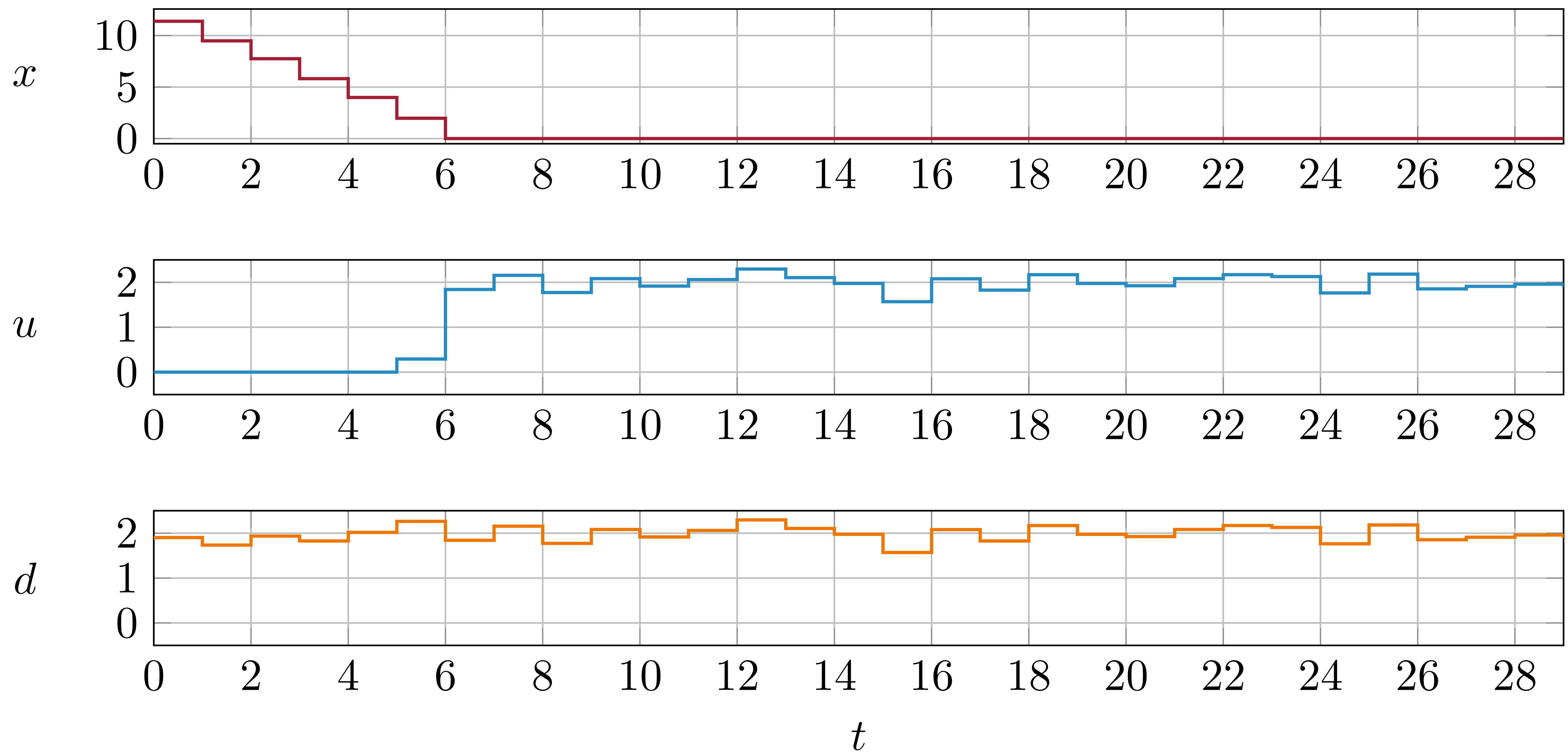
$$u_t = 0 \qquad t \leq 4$$
$$0 \leq u_t \leq M \qquad t > 4$$

Strategies for inventory management

Strategy 2

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

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



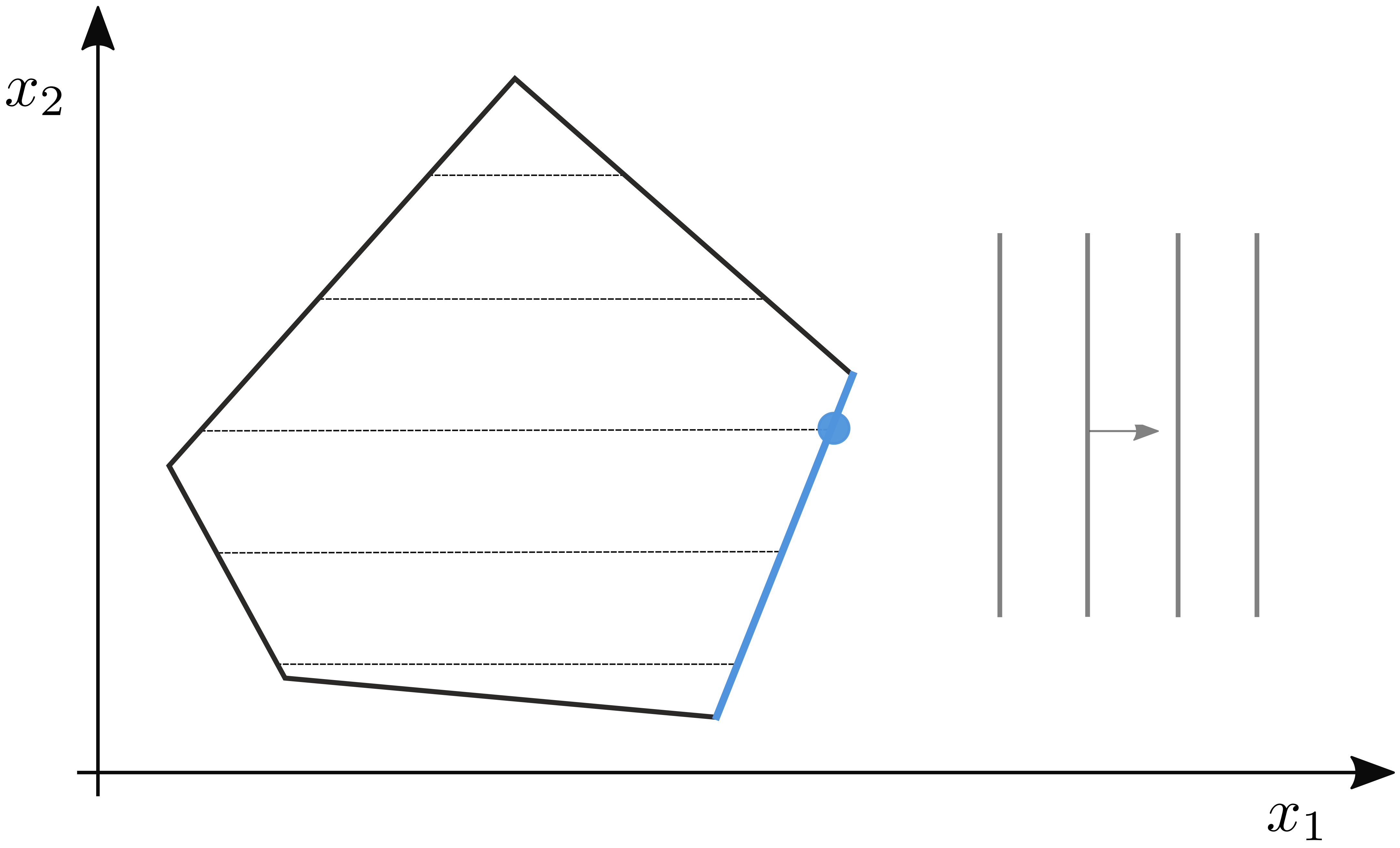
Mixed-integer optimization

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

Mixed-integer optimization

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

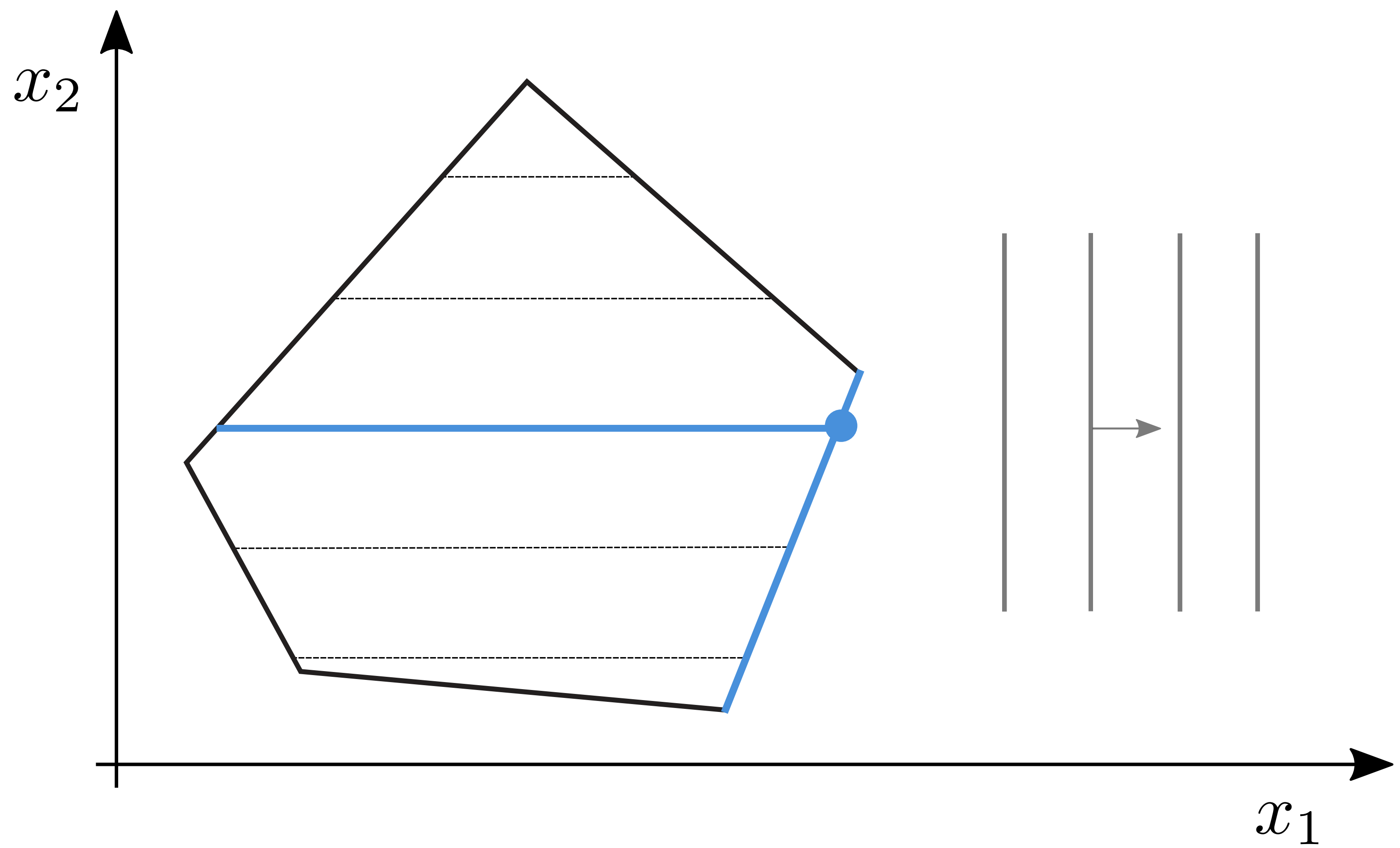
Tight constraints are not enough



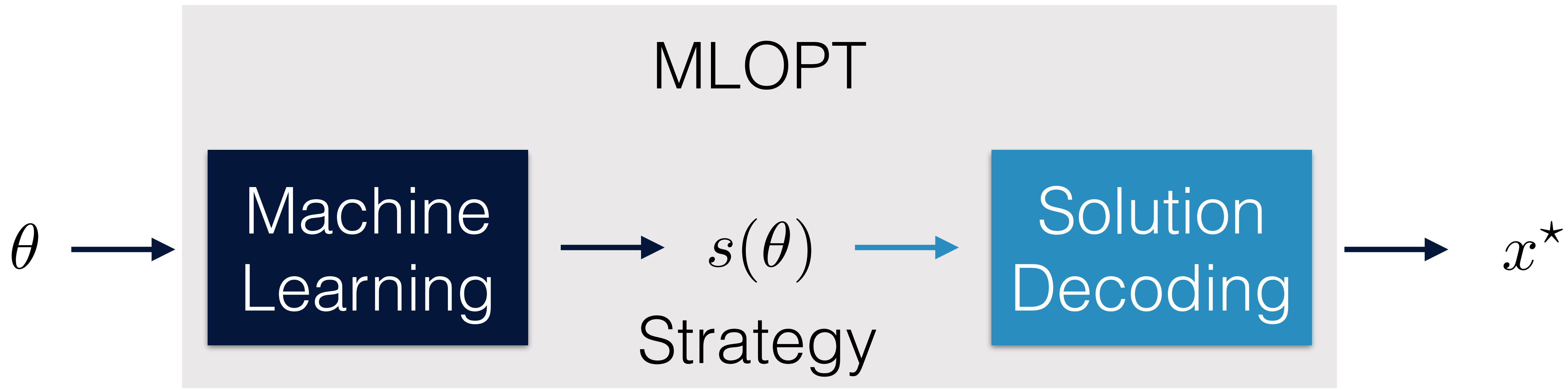
Strategy for mixed-integer optimization

$$s(\theta) = (\mathcal{T}(\theta), x^\star_{\mathcal{I}}(\theta))$$

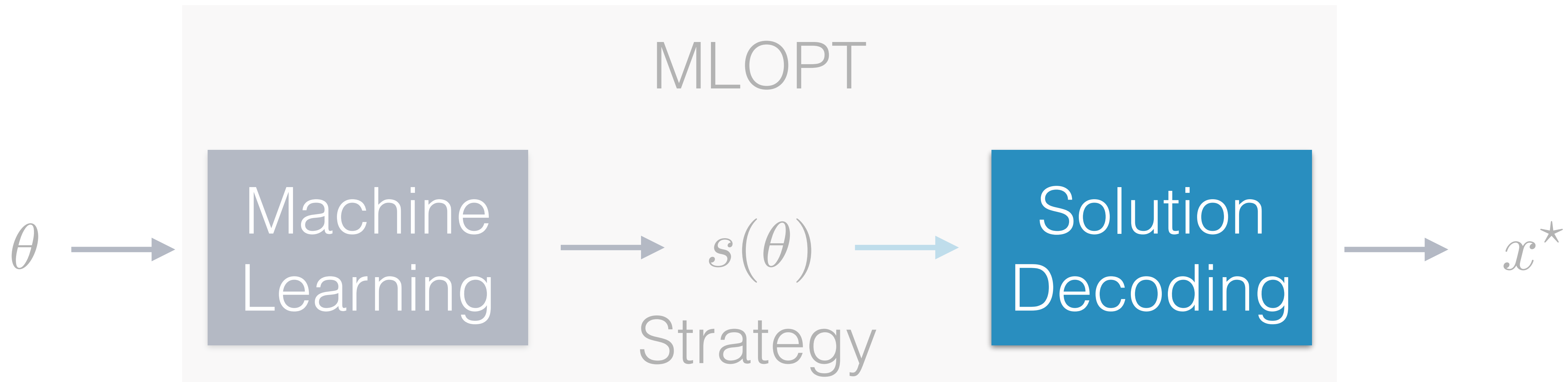
Integer variables



Computing the solution from the optimal strategy



Computing the solution from the optimal strategy



Computing the solution from the optimal strategy

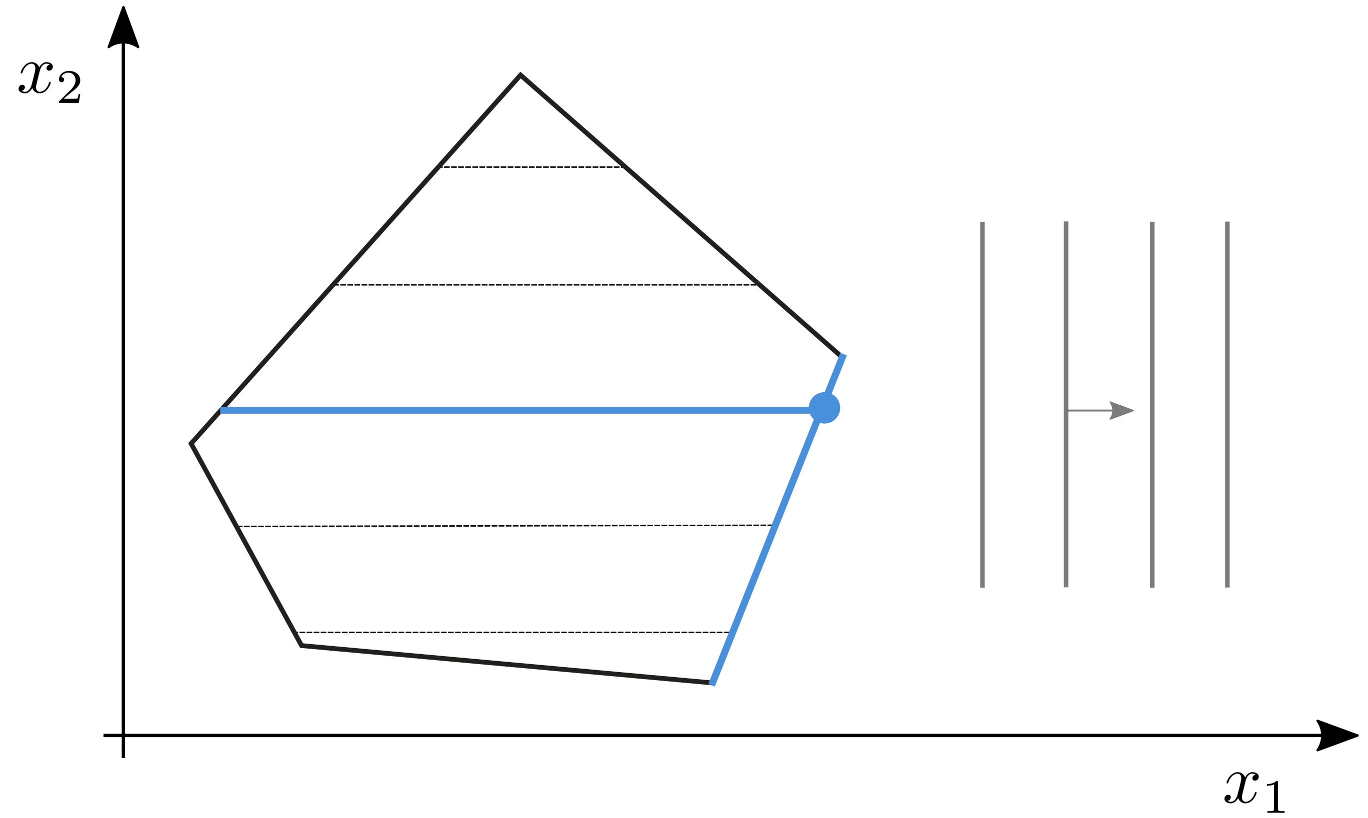
$$\begin{array}{ll} \text{minimize} & c(\theta)^T x \\ \text{subject to} & A(\theta)x \leq b(\theta) \\ & x_{\mathcal{I}} \in \mathbf{Z}^d \end{array}$$

Computing the solution from the optimal strategy

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

Strategy

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

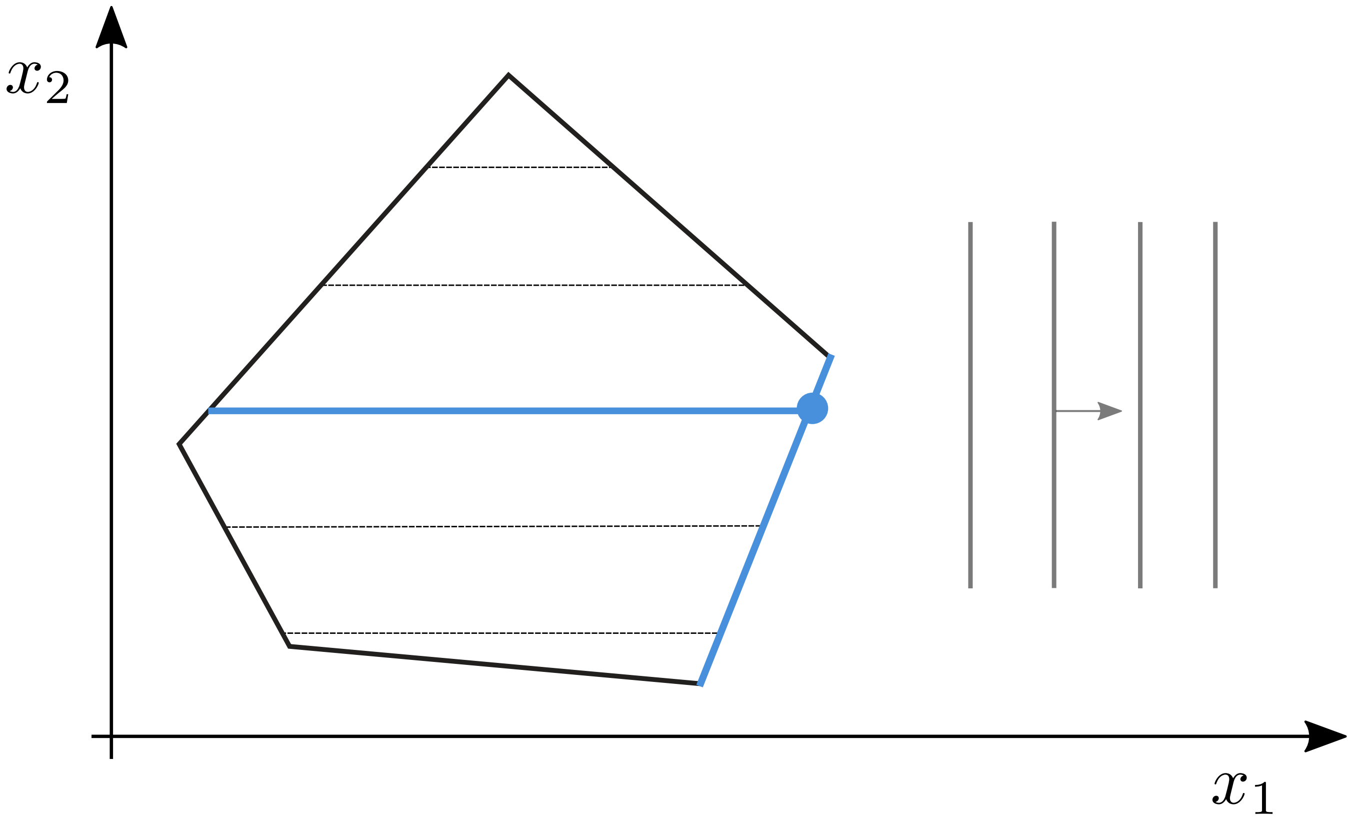


Computing the solution from the optimal strategy

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

Strategy

$$s(\theta) = (\mathcal{T}(\theta), x^\star_{\mathcal{I}}(\theta))$$



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

Integers

Knapsack

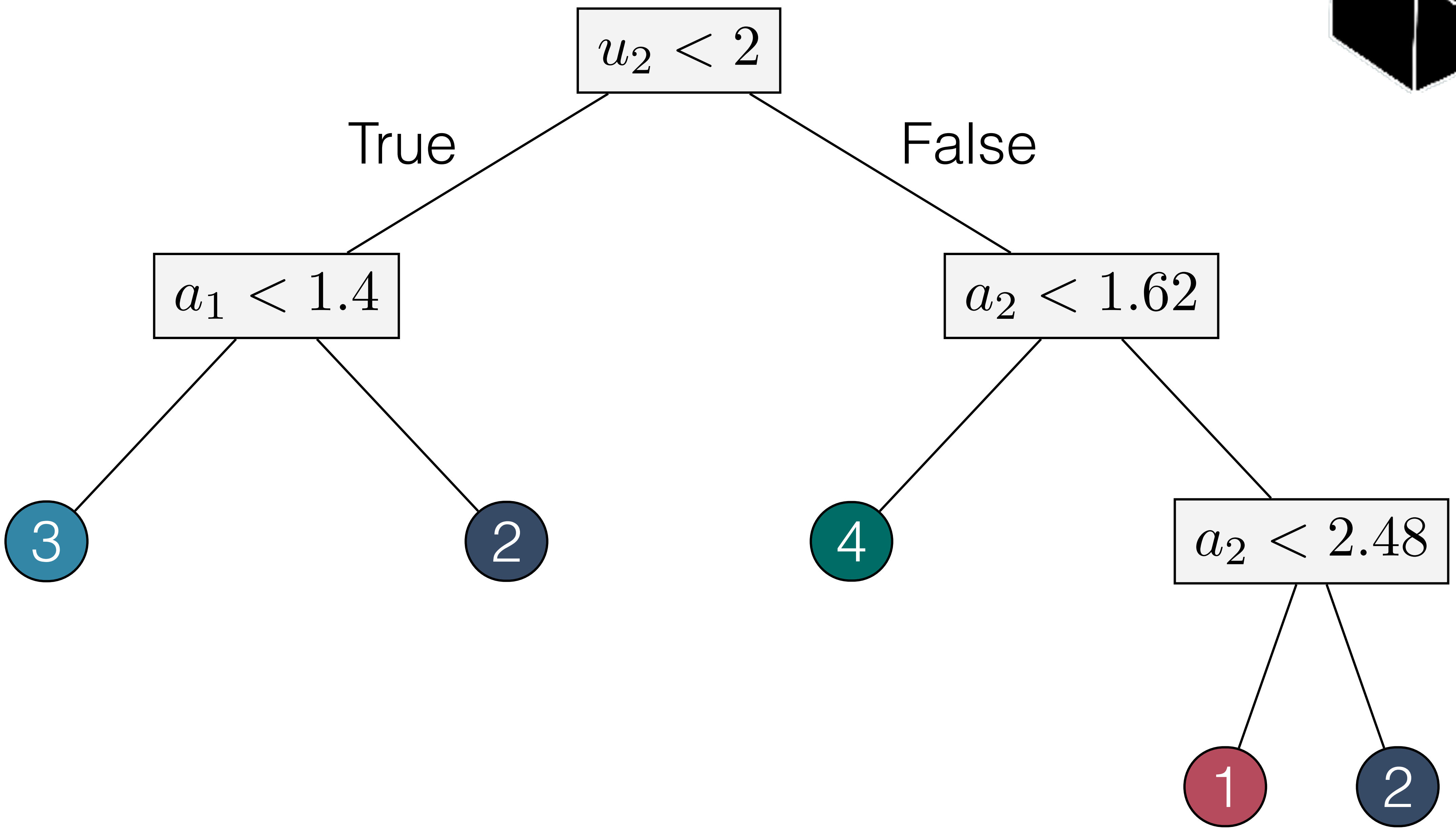
$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & a^T x \leq b \\ & 0 \leq x \leq u \\ & x \in \mathbf{Z}^n \end{array}$$

Knapsack

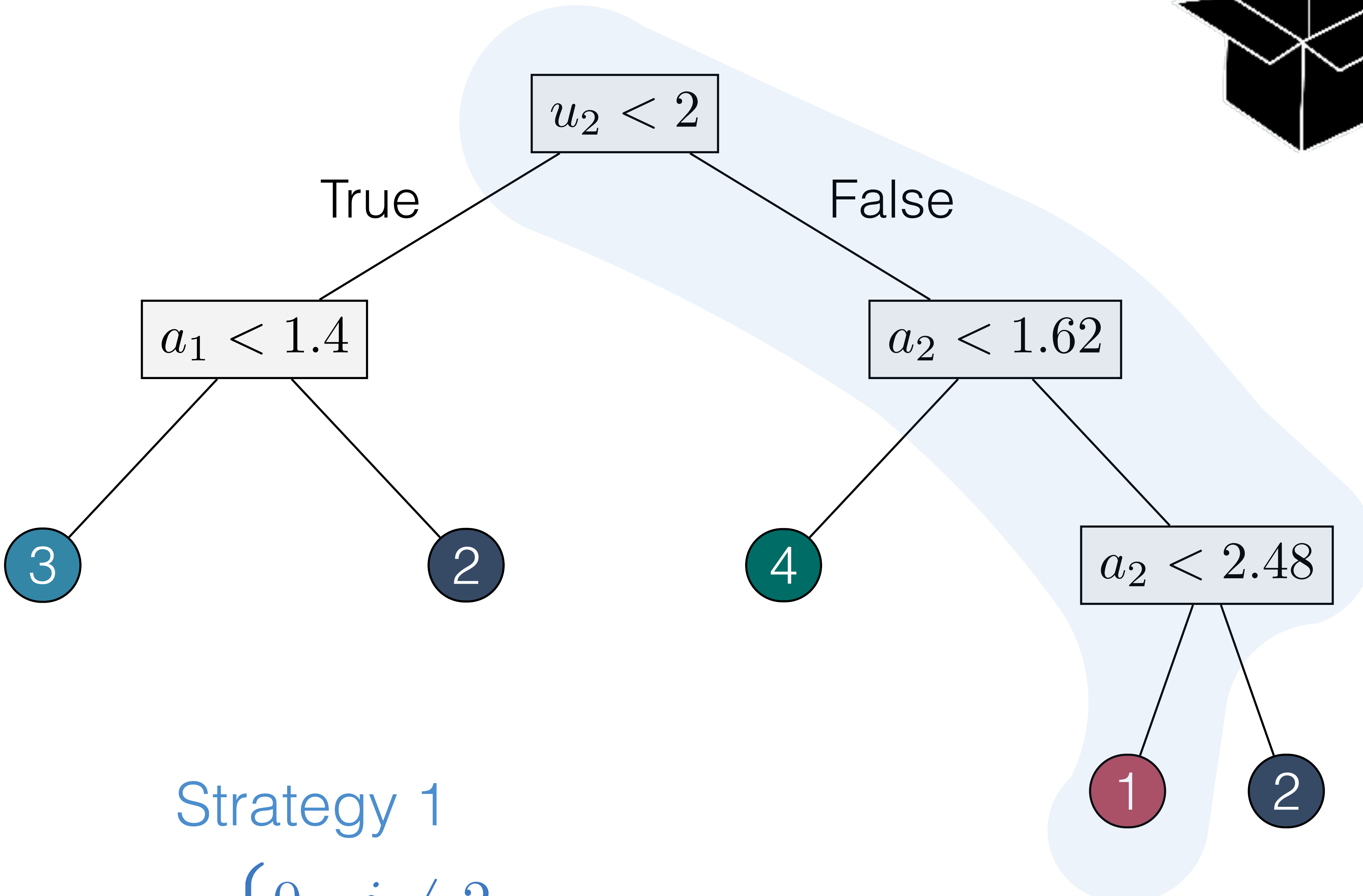
$$
\begin{aligned}
\text{maximize} \quad & c^T x \\
\text{subject to} \quad & a^T x \leq b \\
& 0 \leq x \leq u \\
& x \in \mathbf{Z}^n
\end{aligned}
$$

Parameters

Strategy selection for knapsack



Strategy selection for knapsack

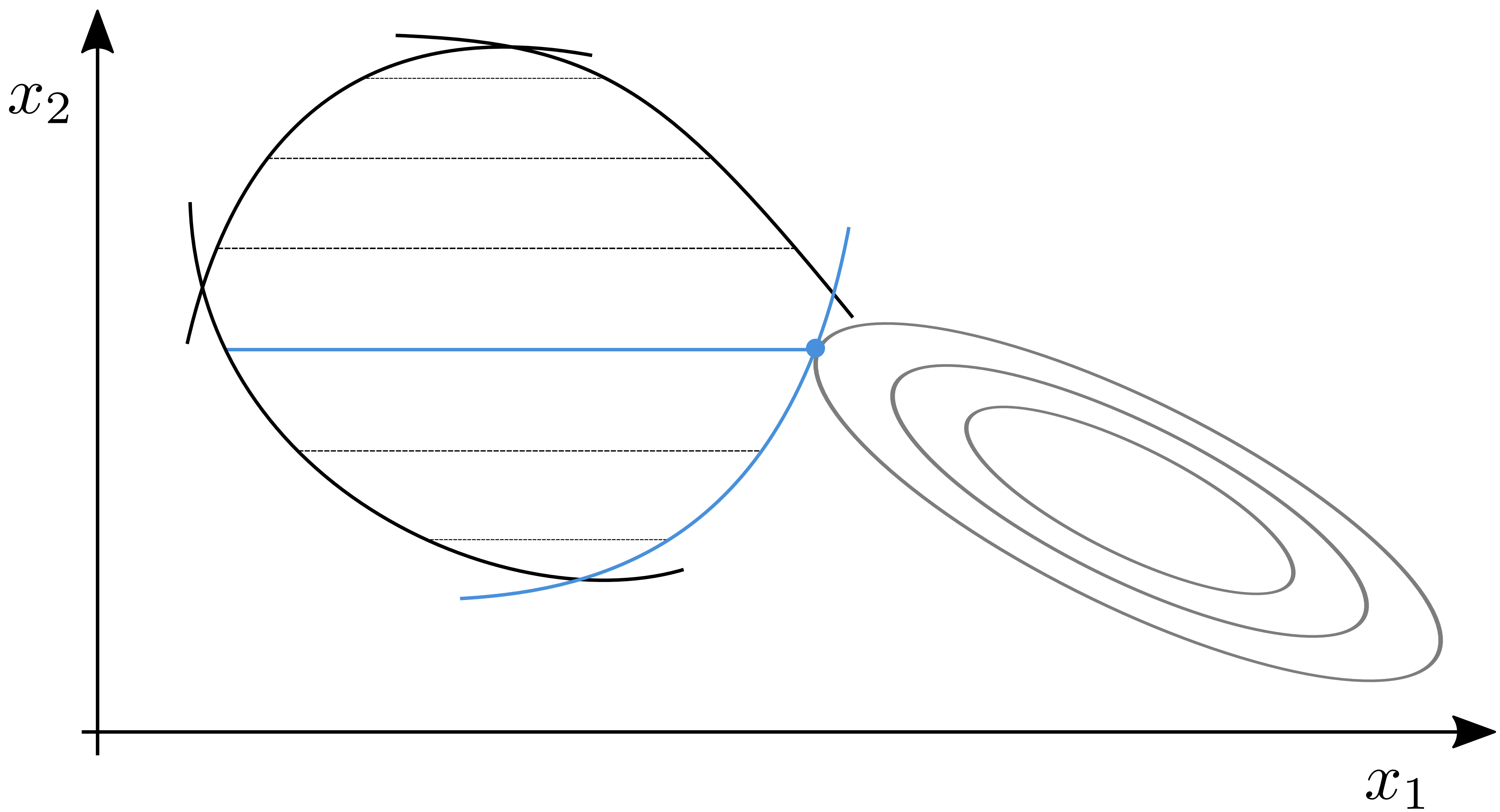


Strategy 1

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

Mixed-integer convex optimization

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



[D. Bertsimas, B. Stellato, *The Voice of Optimization*, Machine Learning (to appear), (2020)]

Machine learning optimizer

- Optimal Strategies
- Strategy Prediction
- Strategy Exploration
- Speedups
- Examples
- Future Work

Machine learning optimizer

Optimal Strategies

Strategy Prediction

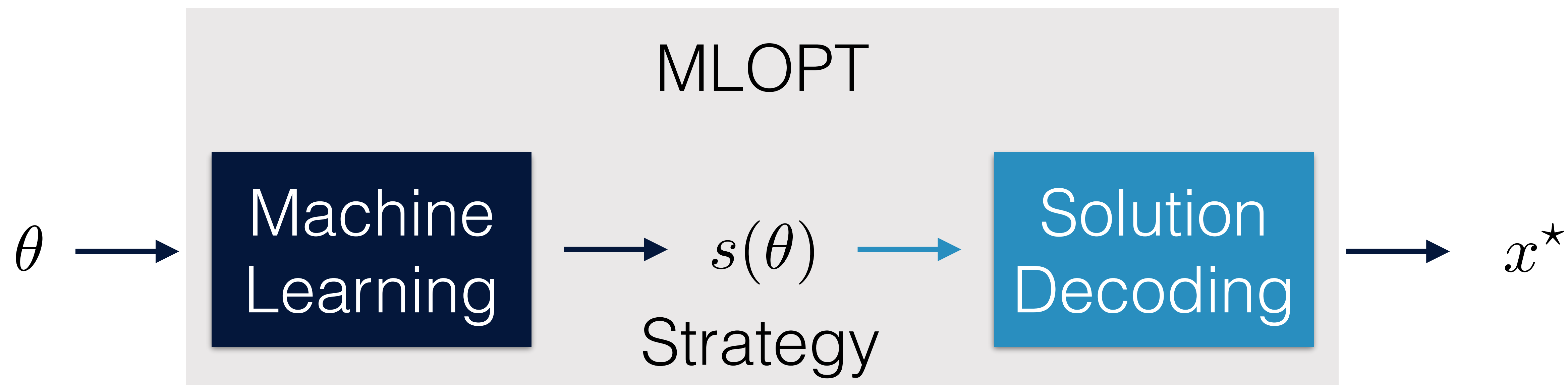
Strategy Exploration

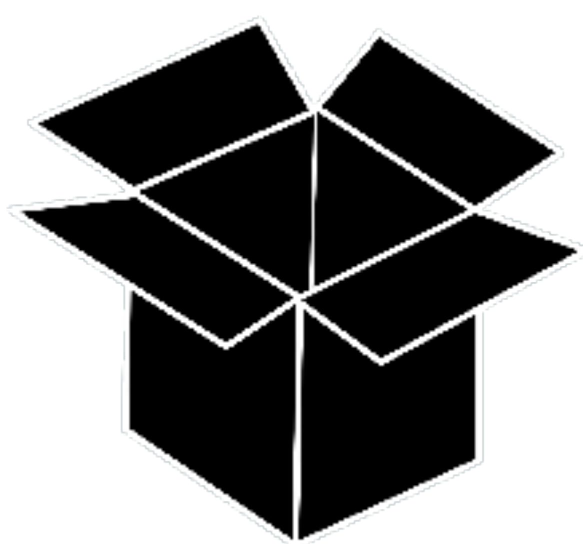
Speedups

Examples

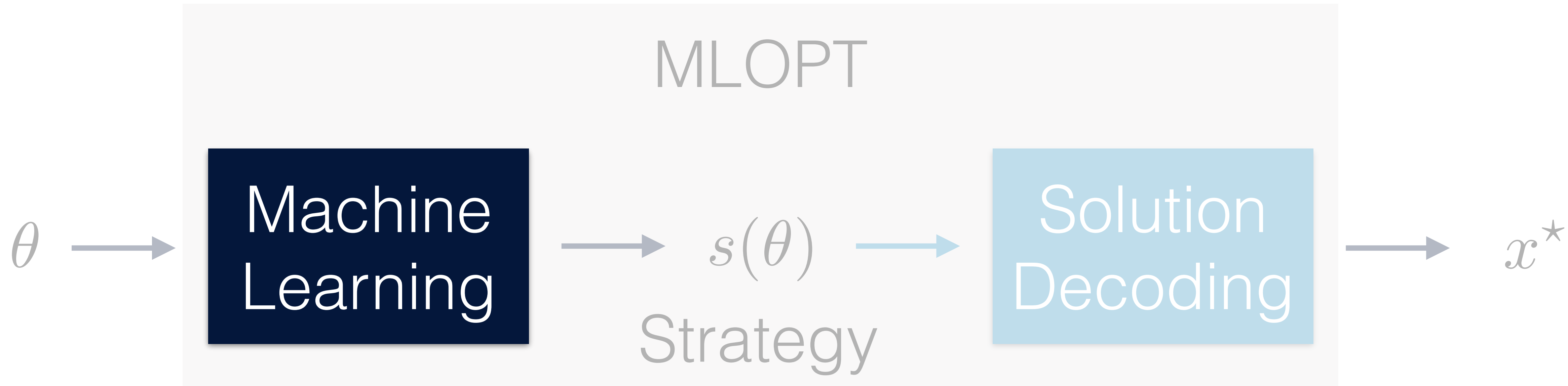
Future Work

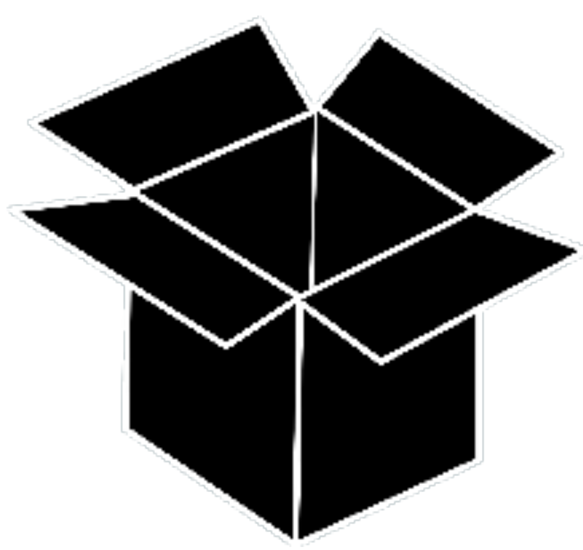
Machine learning optimizer



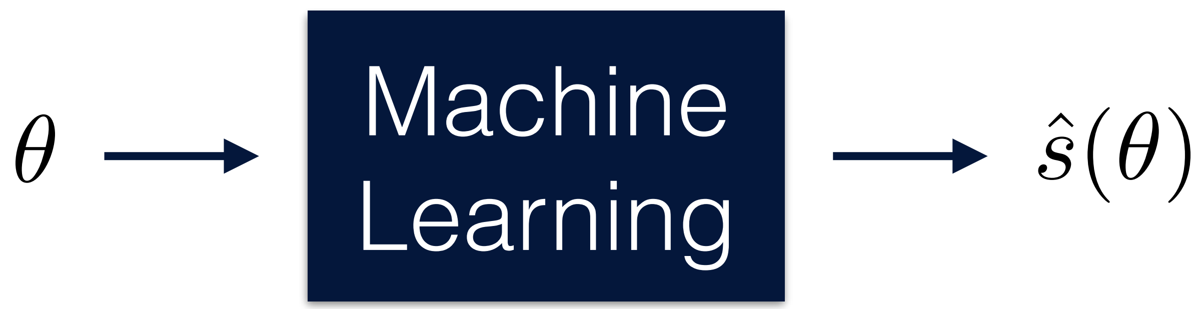


Machine learning optimizer

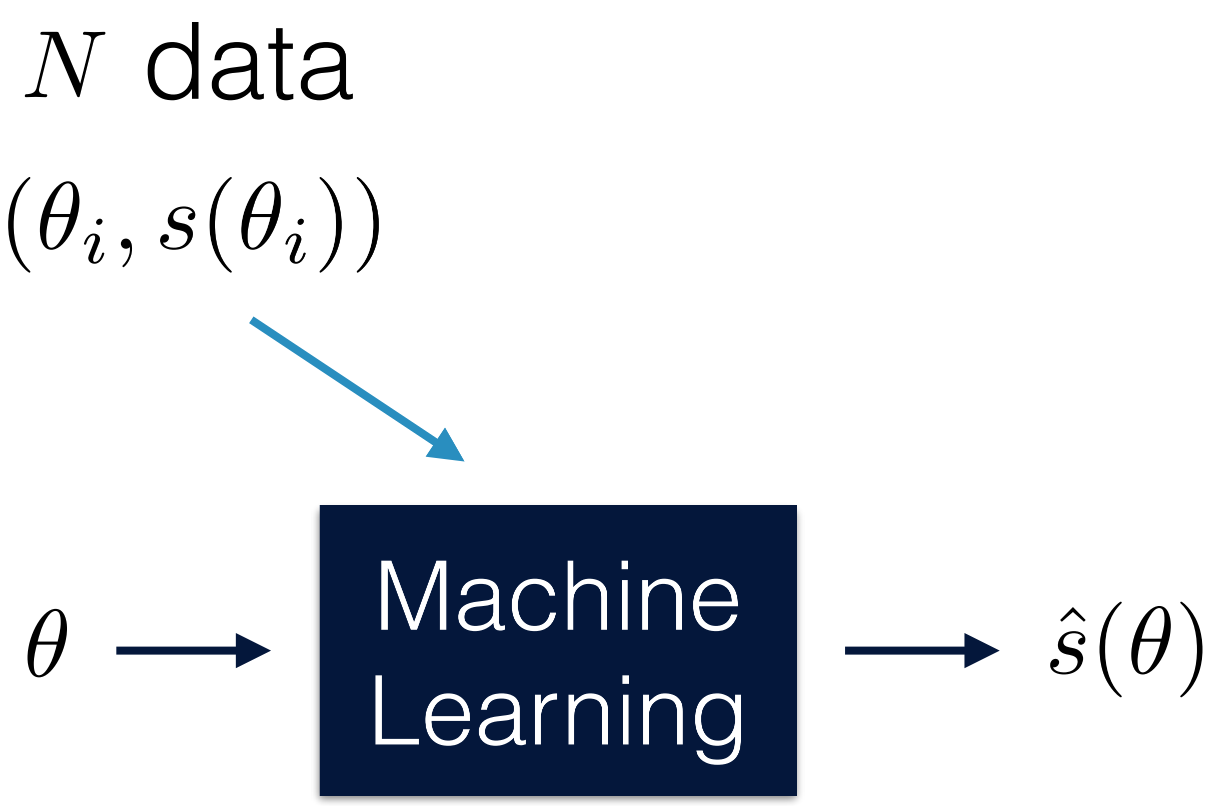




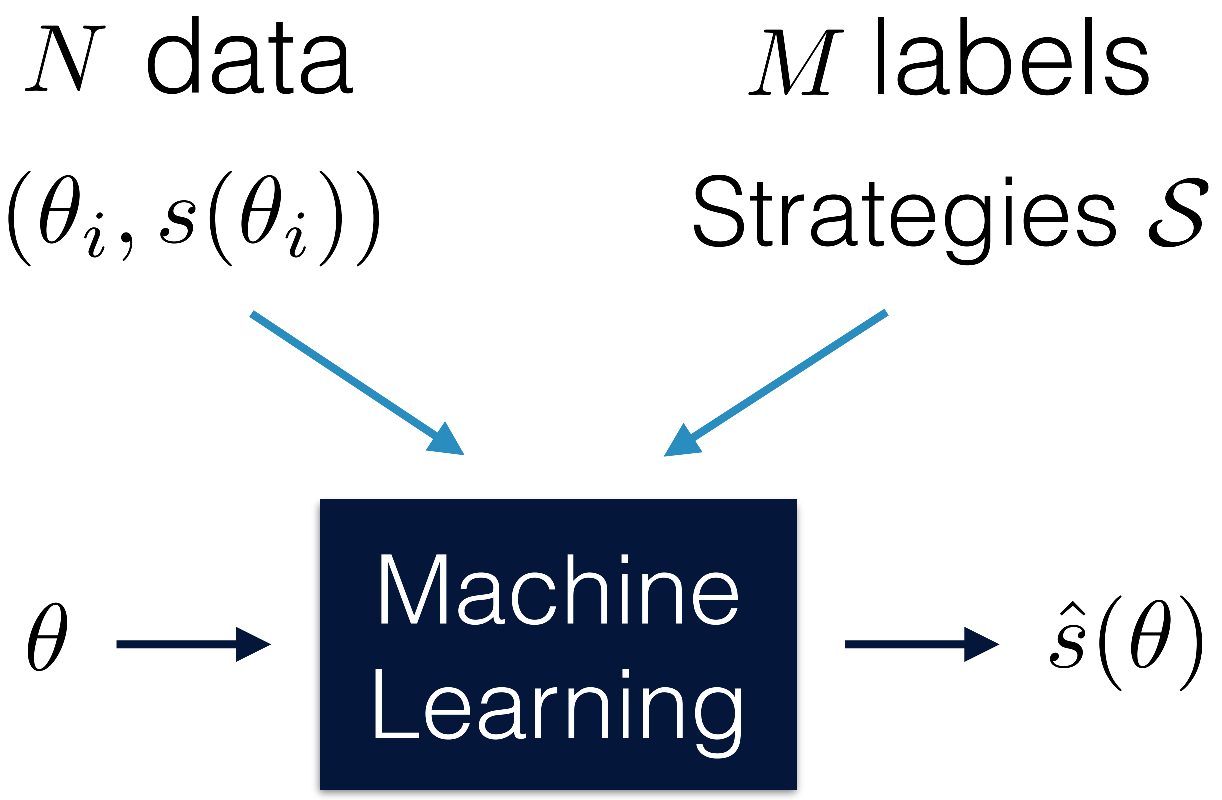
Selecting strategies is a classification problem



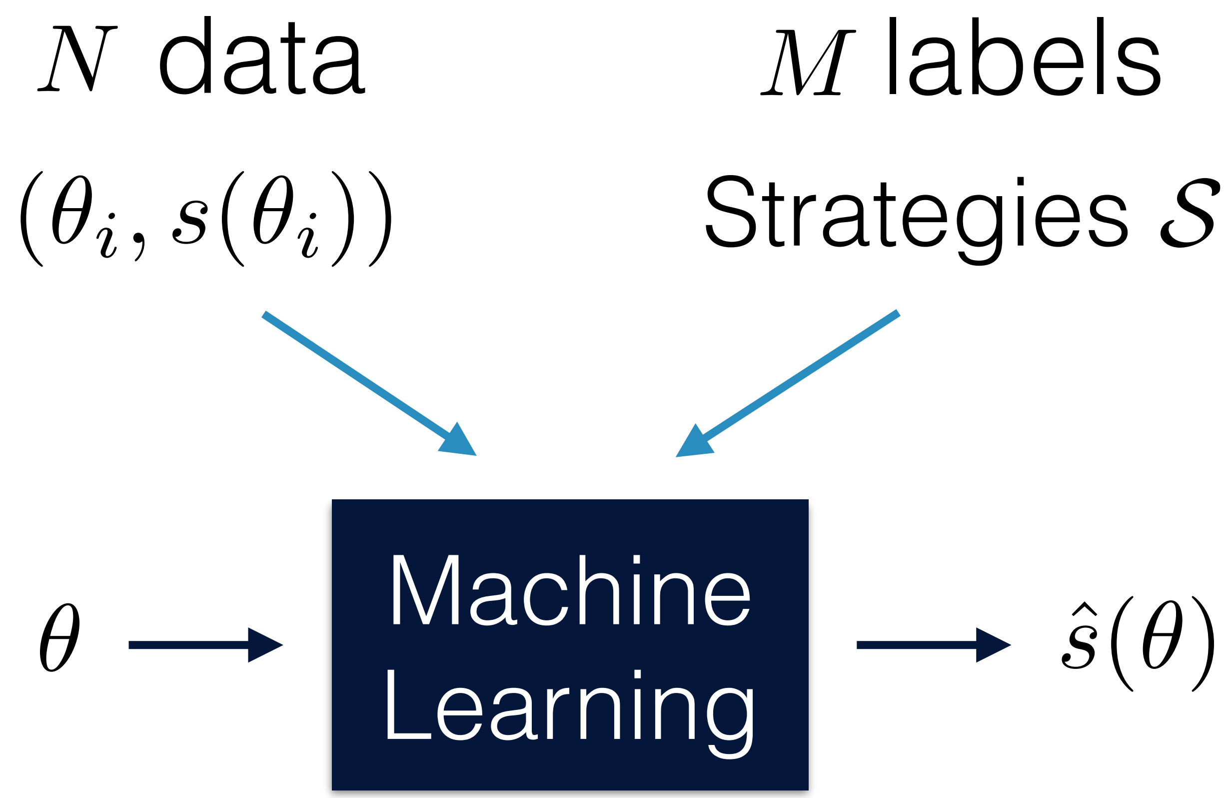
Selecting strategies is a classification problem



Selecting strategies is a classification problem



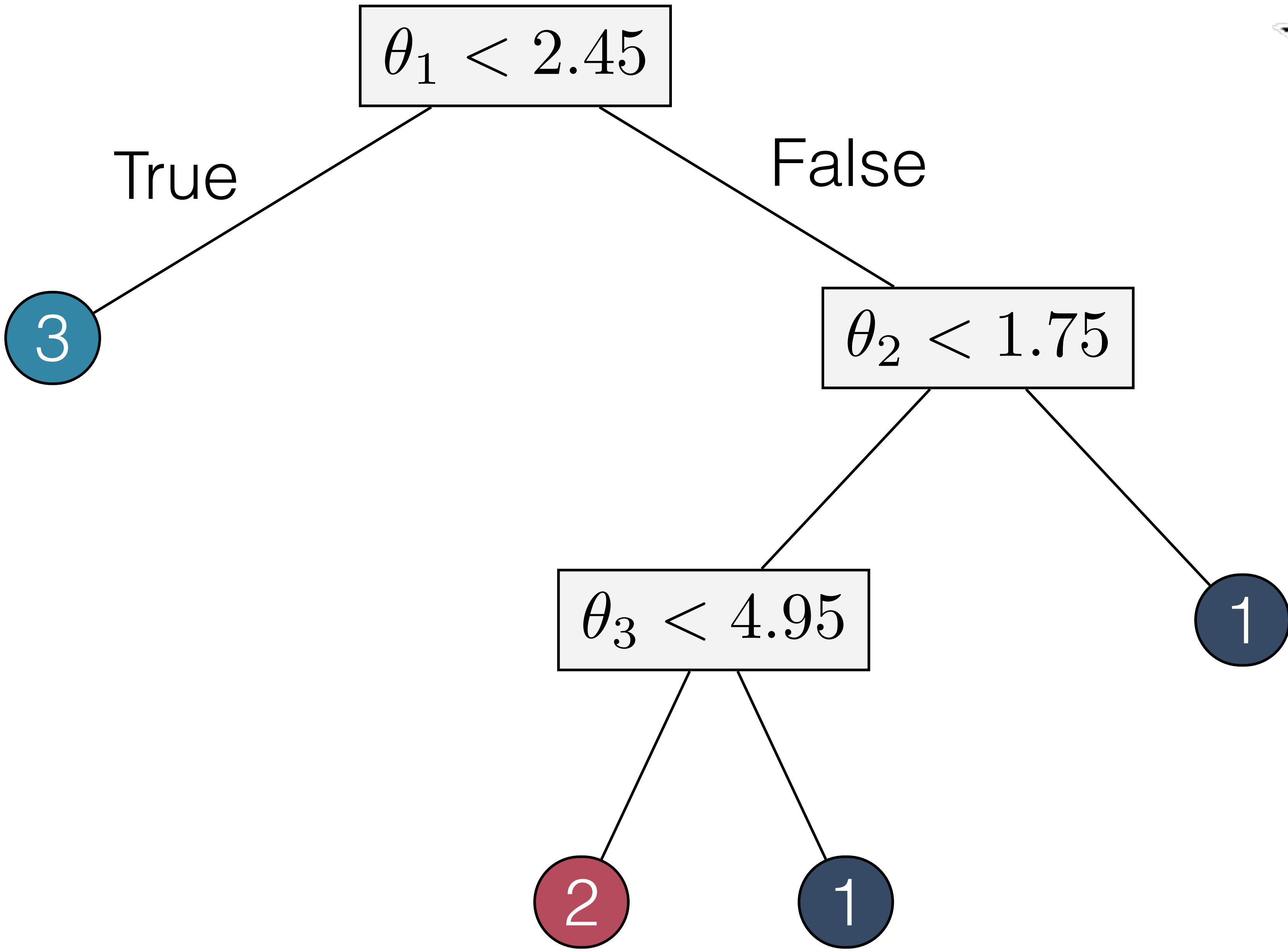
Selecting strategies is a classification problem

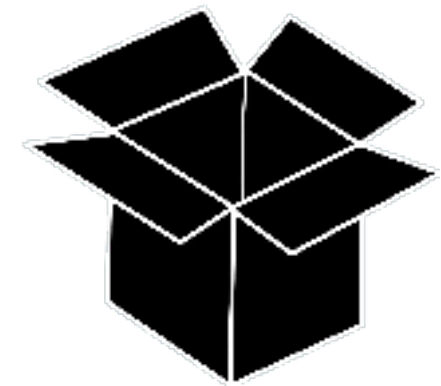


Optimal Trees

Neural Networks

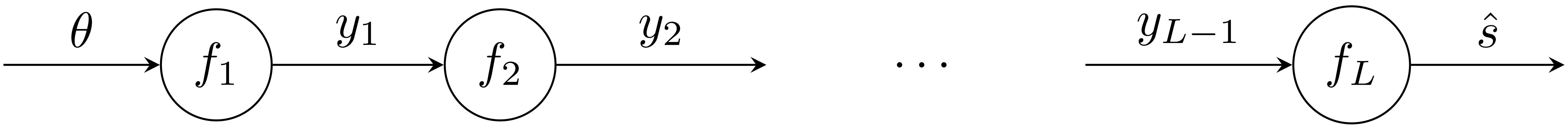
Optimal Classification Trees (OCT)





[Bertsimas and Dunn (2017)]

Neural Networks (NN)



Single layer

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

PyTorch

Machine learning optimizer

Optimal Strategies

Strategy Prediction

Strategy Exploration

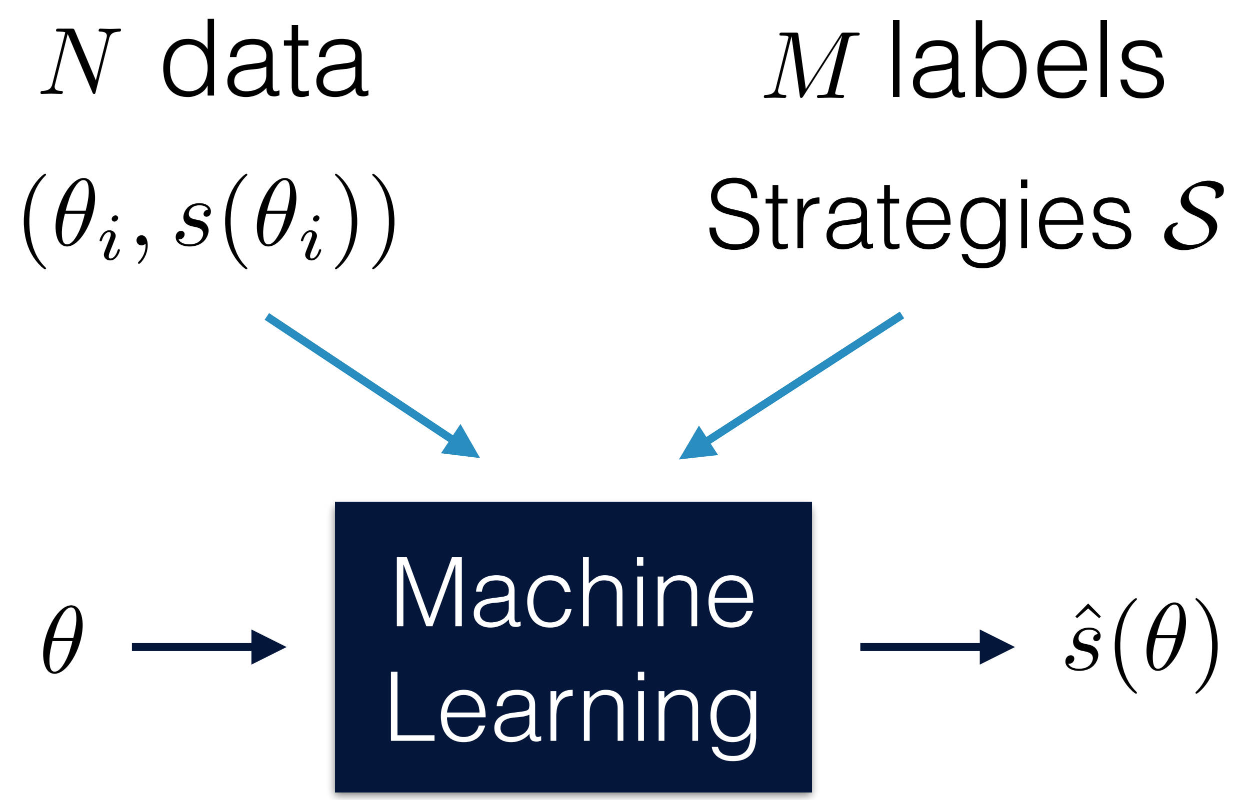
Speedups

Examples

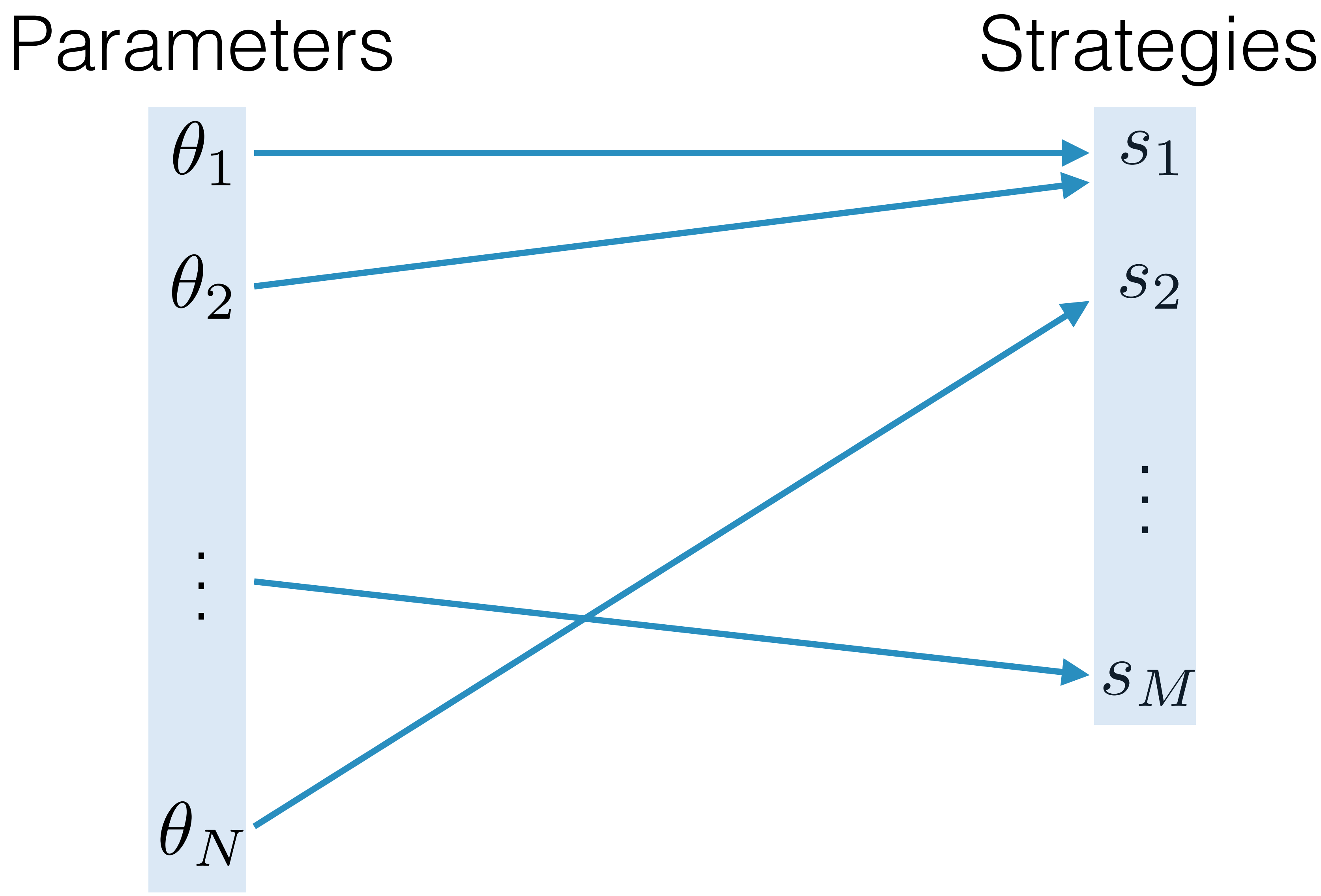
Machine learning optimizer

- Optimal Strategies
- Strategy Prediction
- **Strategy Exploration**
- Speedups
- Examples

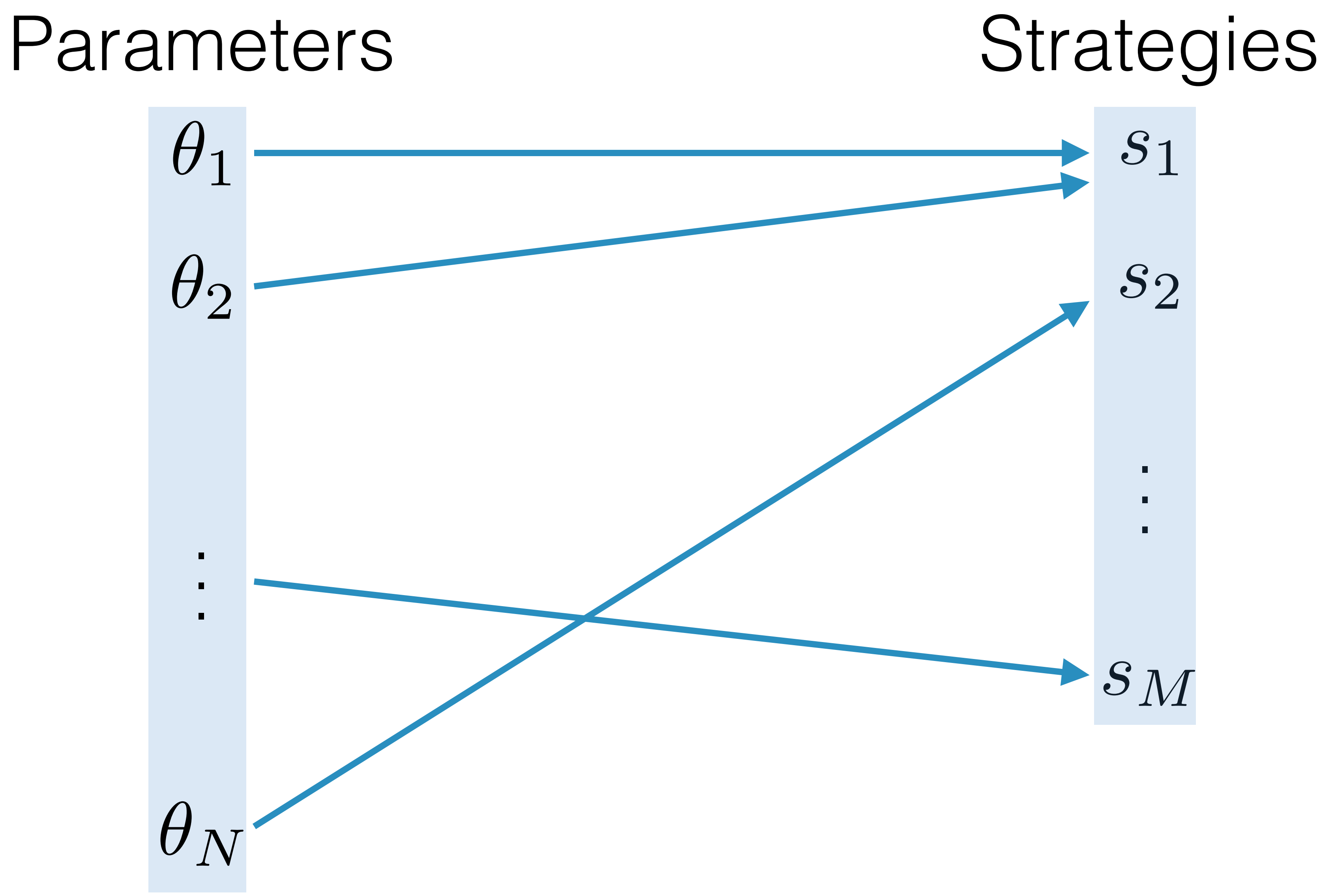
Have we seen enough data?



Will we find new strategies?

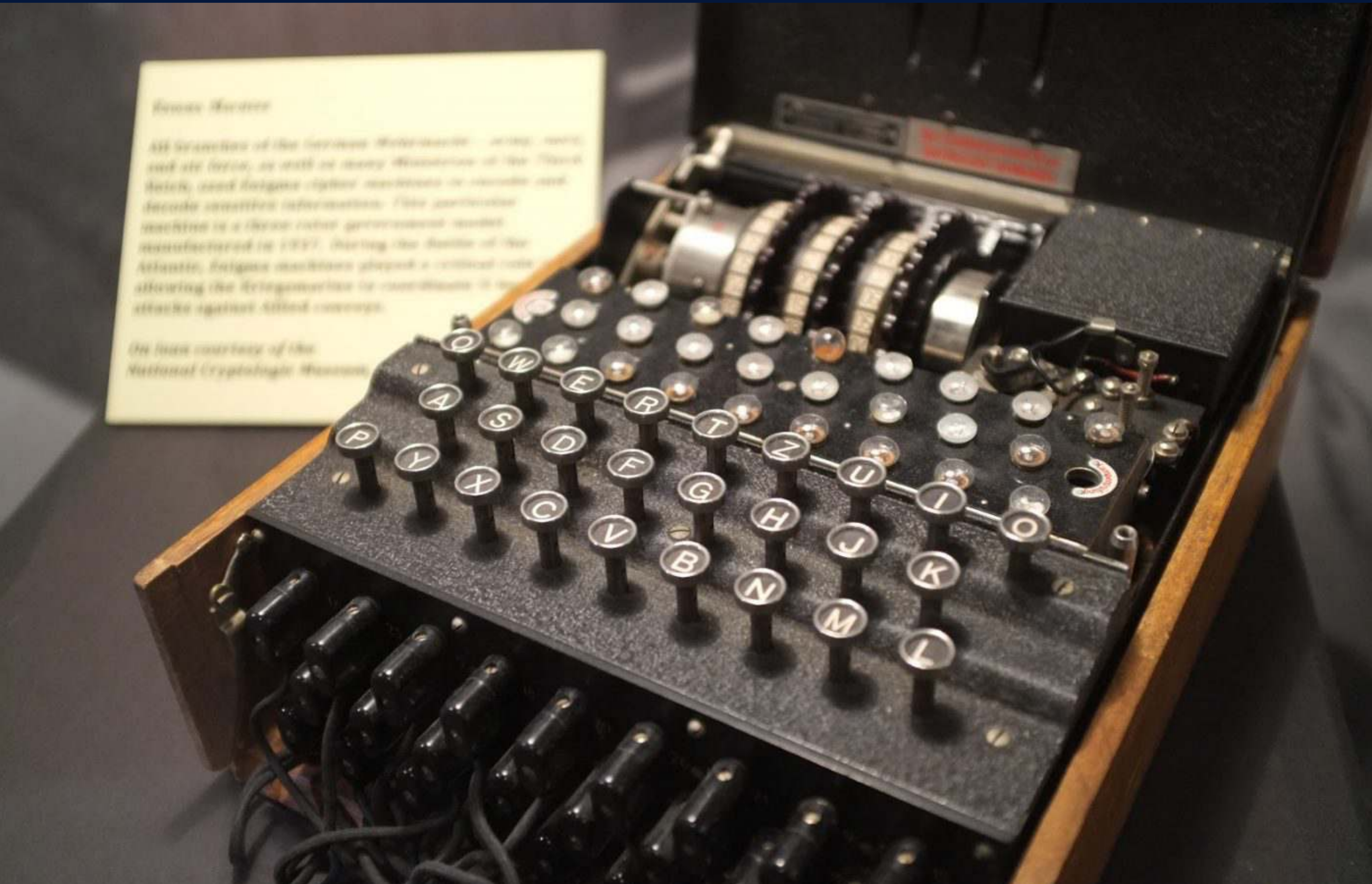


Will we find new strategies?



θ_{N+1}?

Alan Turing already knew that…



Good-Turing estimator

$N = 15$
$M = 5$

→

s_1	6 times
s_2	3 times
s_3	1 time
s_4	3 times
s_5	2 times

Good-Turing estimator

$N = 15$
$M = 5$

→

s_1	6 times
s_2	3 times
s_3	1 time
s_4	3 times
s_5	2 times

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

Good-Turing estimator

$N = 15$
$M = 5$

s_1	6 times
s_2	3 times
s_3	1 time
s_4	3 times
s_5	2 times

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

Probability of unseen strategies

Good-Turing estimator

$$N = 15$$
$$M = 5$$

→

s_1	6 times
s_2	3 times
s_3	1 time
s_4	3 times
s_5	2 times

strategies appeared once

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

Probability of unseen strategies

Good-Turing estimator

$N = 15$
$M = 5$ →

s_1	6 times
s_2	3 times
s_3	1 time
s_4	3 times
s_5	2 times

strategies appeared once

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

Probability of unseen strategies

samples

Good-Turing estimator

$N = 15$
$M = 5$ →

s_1	6 times
s_2	3 times
s_3	1 time
s_4	3 times
s_5	2 times

→ $GT = 1/15$

strategies appeared once

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

samples

Probability of unseen strategies

Good-Turing estimator

$$N = 15$$
$$M = 5$$

→

s_1	6 times
s_2	3 times
s_3	1 time
s_4	3 times
s_5	2 times

→ $GT = 1/15$

strategies appeared once

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

samples

Probability of unseen strategies

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

Good-Turing estimator

$N = 15$
$M = 5$

→

s_1	6 times
s_2	3 times
s_3	1 time
s_4	3 times
s_5	2 times

→ $GT = 1/15$

\# strategies appeared once

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

Probability of unseen strategies

\# samples

Bound

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

[McAllester and Schapire (2001)]

Simple sampling scheme

Bound

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

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

Machine learning optimizer

Optimal Strategies

Strategy Prediction

Strategy Exploration

Speedups

Examples

Machine learning optimizer

Optimal Strategies

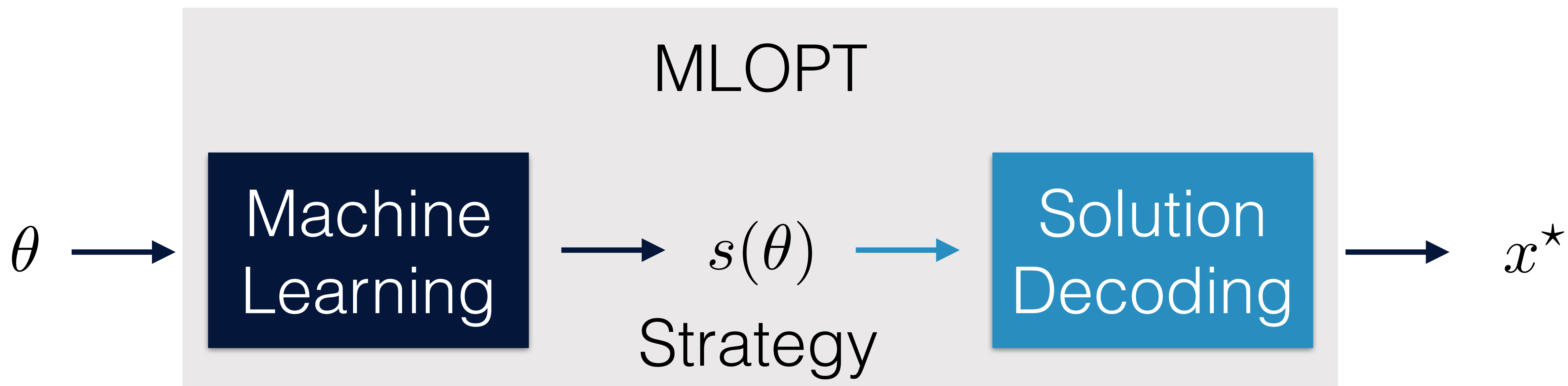
Strategy Prediction

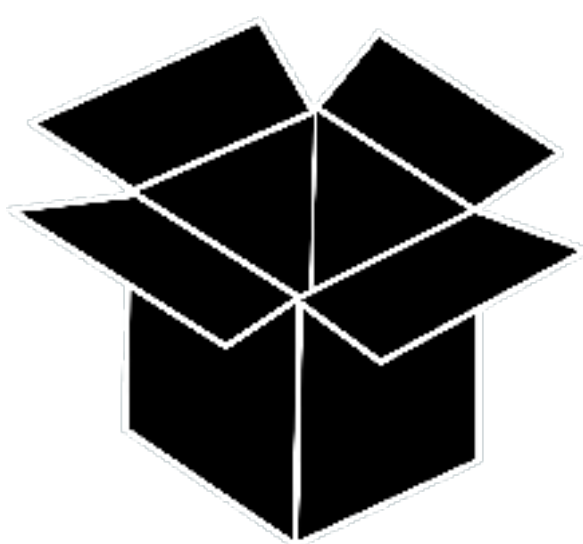
Strategy Exploration

Speedups

Examples

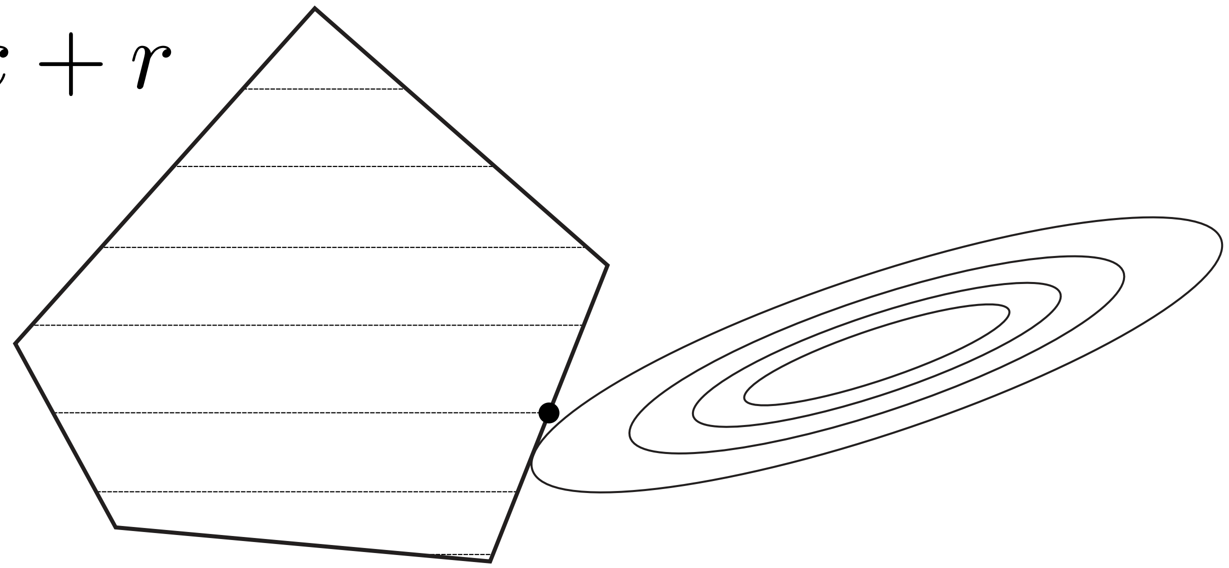
Machine learning optimizer





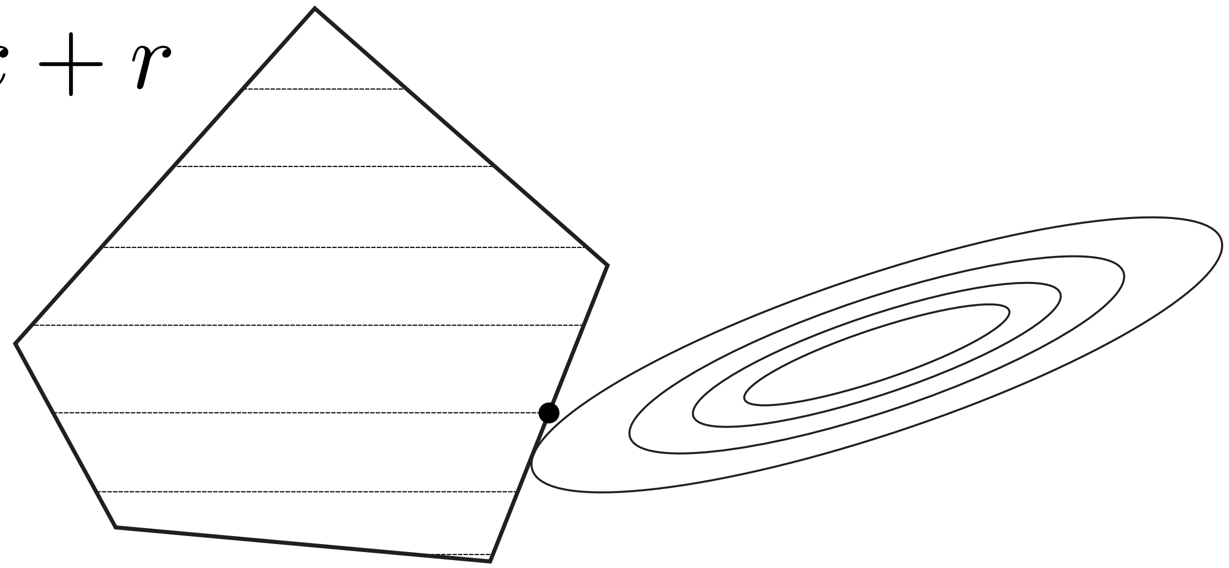
Mixed-integer quadratic optimization

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



Mixed-integer quadratic optimization

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



Strategy

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

Equality Constrained QP

Quick online solution

KKT System

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

[D. Bertsimas, B. Stellato, *Online Mixed-Integer Optimization in Milliseconds*, INFORMS Journal on Computing (under review), (2019)]

Quick online solution

KKT System

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

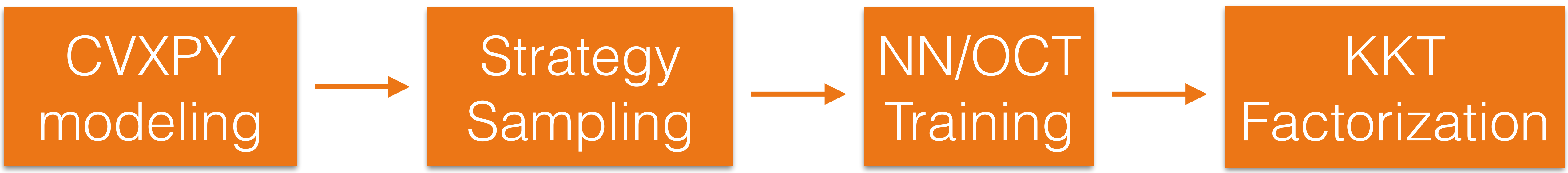
$\mathcal{NP}$-hard problem → Linear System

[D. Bertsimas, B. Stellato, *Online Mixed-Integer Optimization in Milliseconds*, INFORMS Journal on Computing (under review), (2019)]

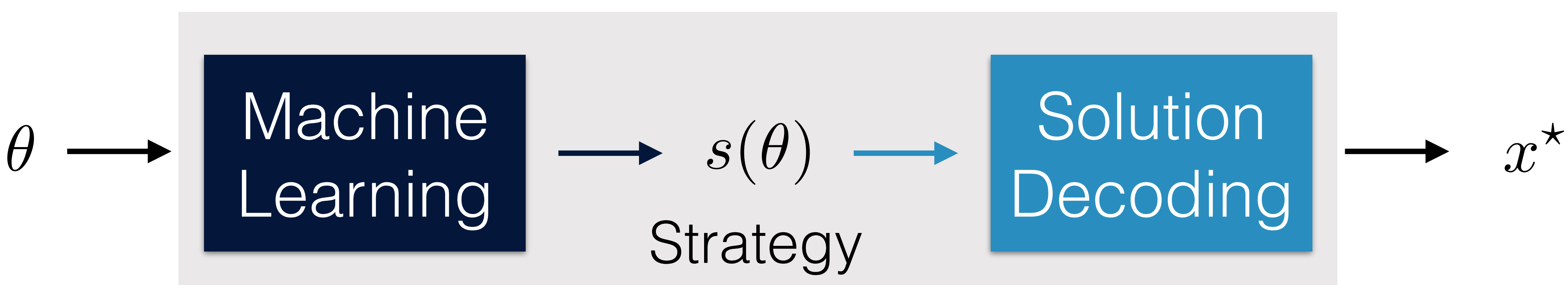
MLOPT: Machine Learning Optimizer

`github.com/bstellato/mlopt`

Offline Learning



Fast Online Optimization



[D. Bertsimas, B. Stellato, *The Voice of Optimization*, Machine Learning (to appear), (2020)]

Machine learning optimizer

Optimal Strategies

Strategy Prediction

Strategy Exploration

Speedups

Examples

Machine learning optimizer

Optimal Strategies

Strategy Prediction

Strategy Exploration

Speedups

Examples

Sparse portfolio optimization

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

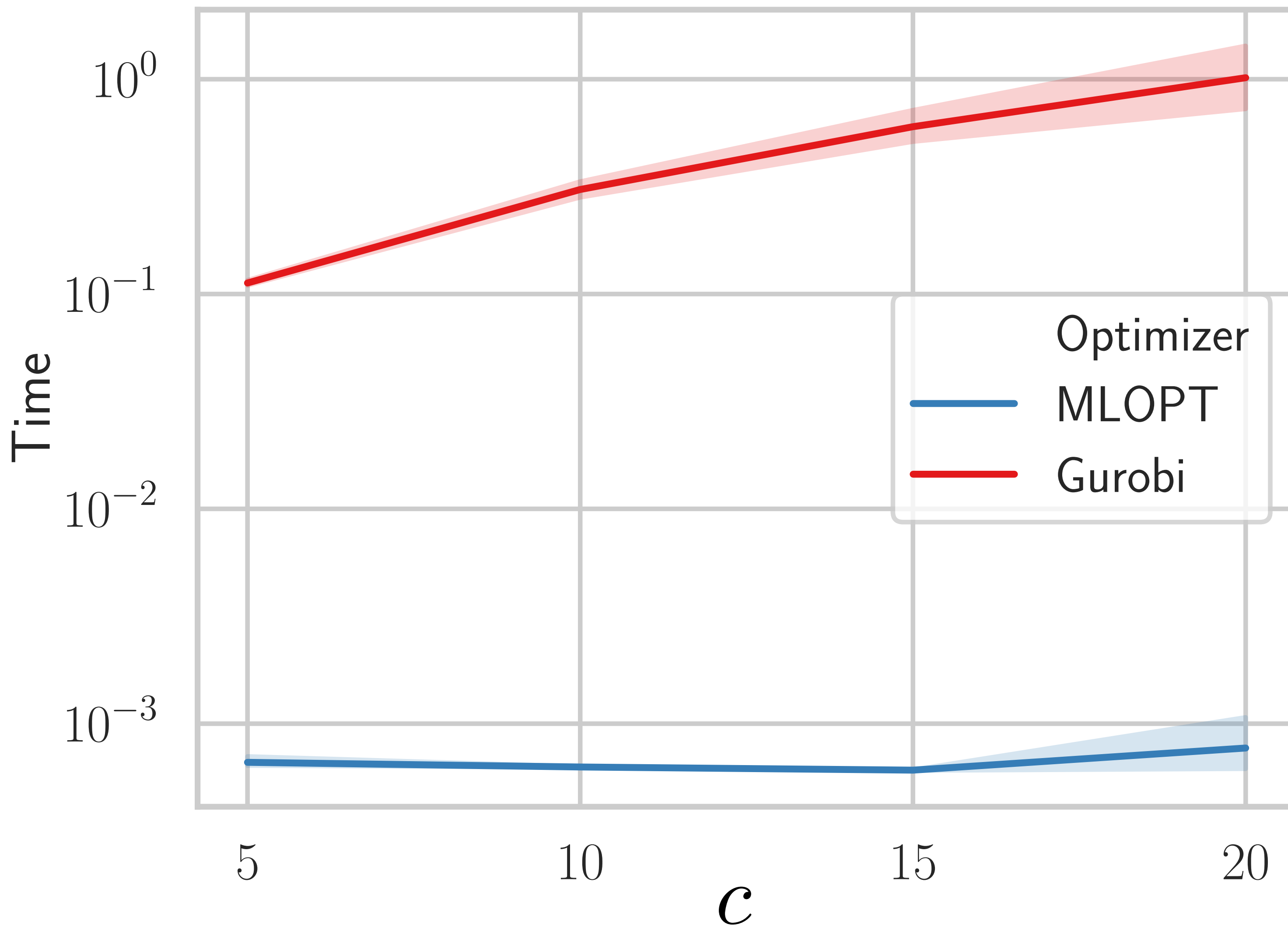
k-factor Risk Model

$$\Sigma = F \Sigma^k F^T + D$$

Parameters

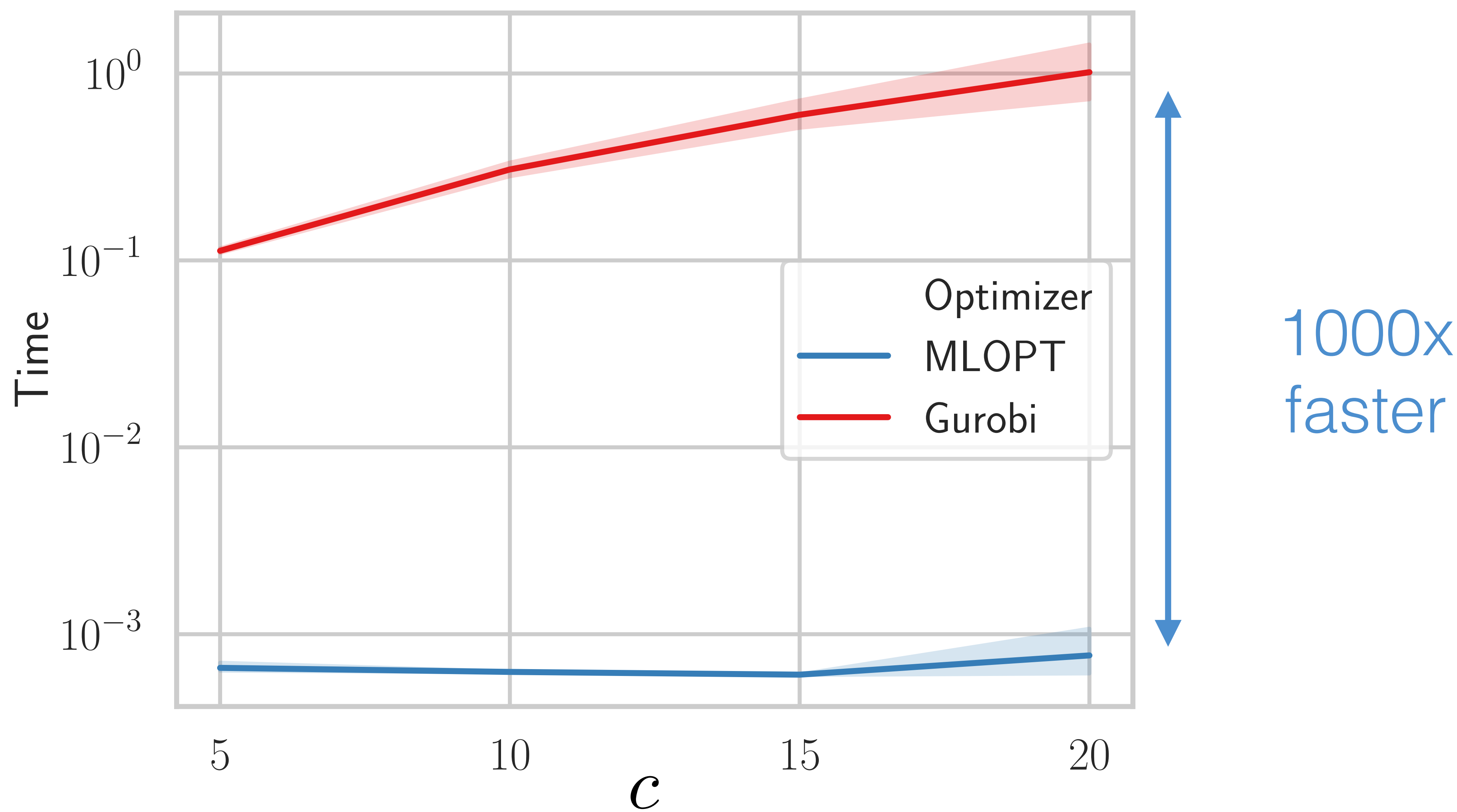
$$\theta = (r, \Sigma^k, F, D)$$

S&P100 backtesting: high speed and accuracy



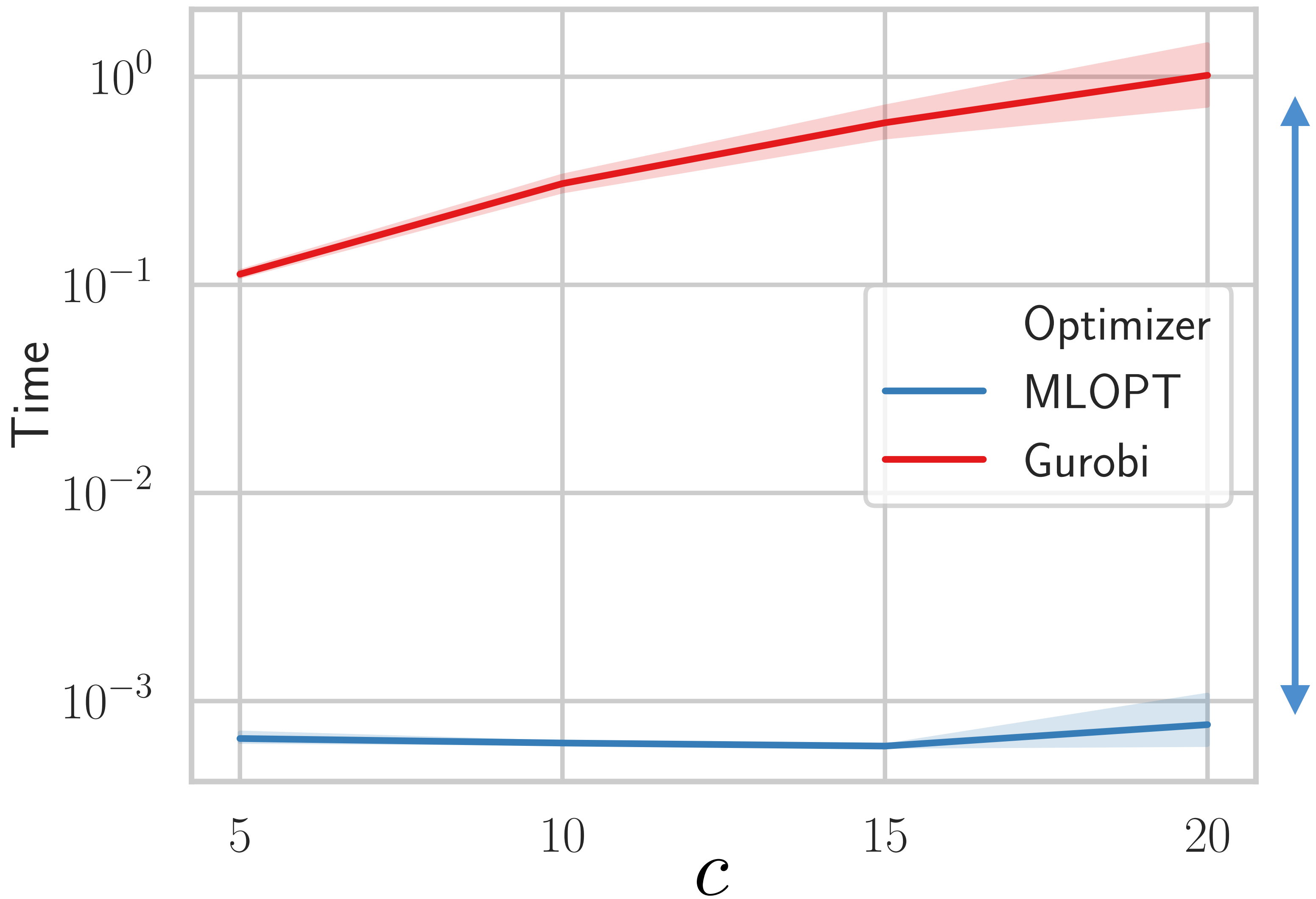
c	M	acc [%]	suboptimality	infeasibility
5	1027	99.11	9.88×10^{-3}	2.25×10^{-9}
10	1083	99.21	7.01×10^{-3}	8.19×10^{-7}
15	1120	99.30	1.92×10^{-2}	2.68×10^{-7}
20	1209	99.50	3.54×10^{-3}	4.12×10^{-7}

S&P100 backtesting: high speed and accuracy



c	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

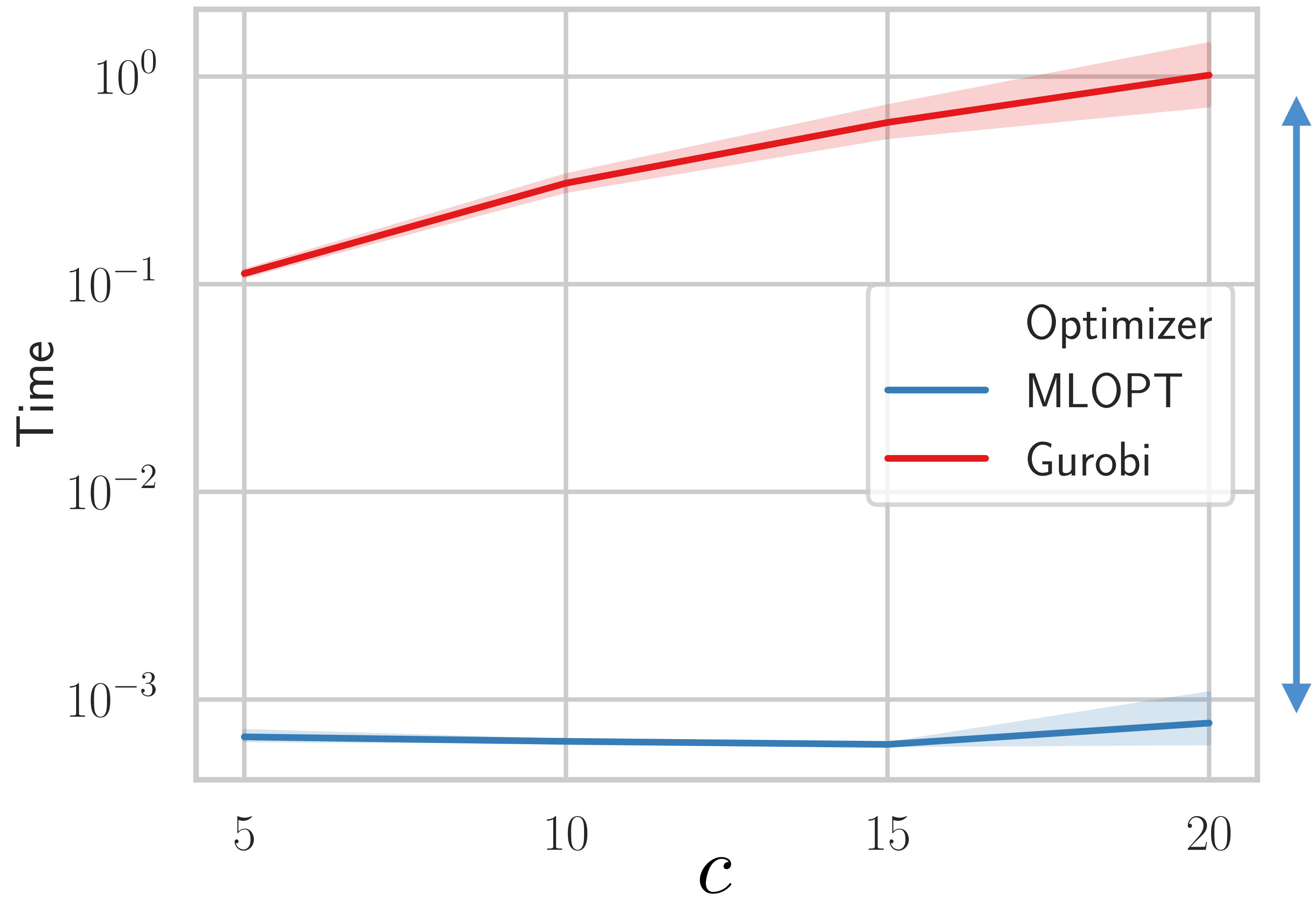


1000x faster

Low suboptimality and infeasibility

c	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



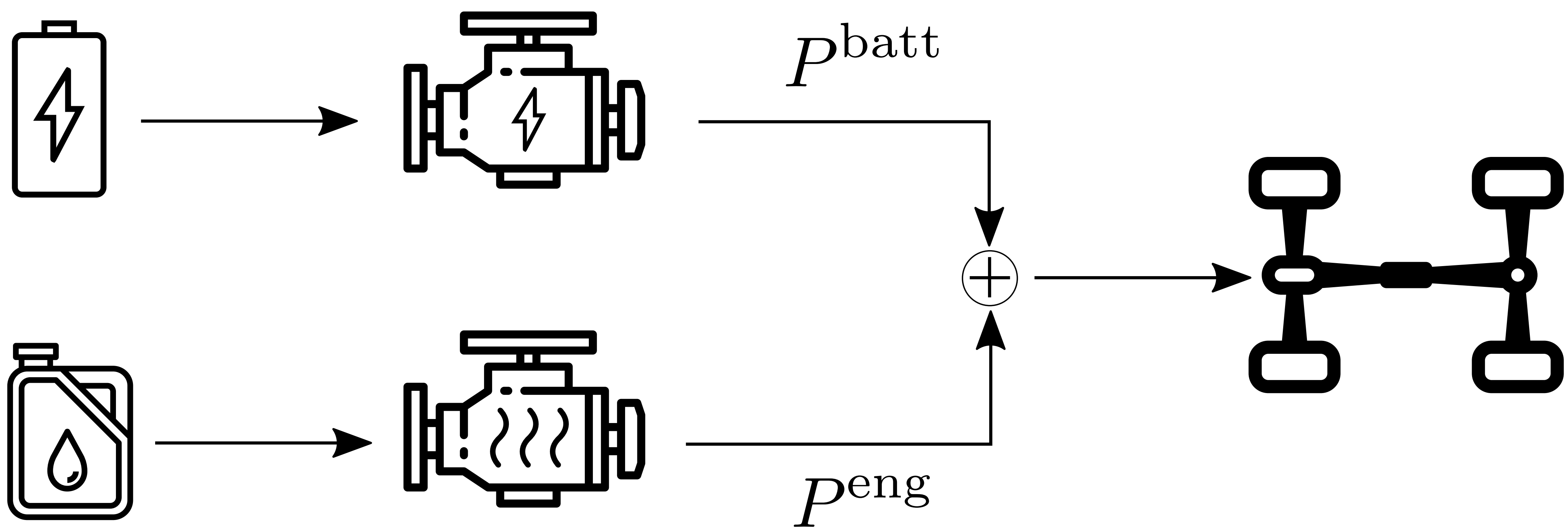
1000x faster

High accuracy

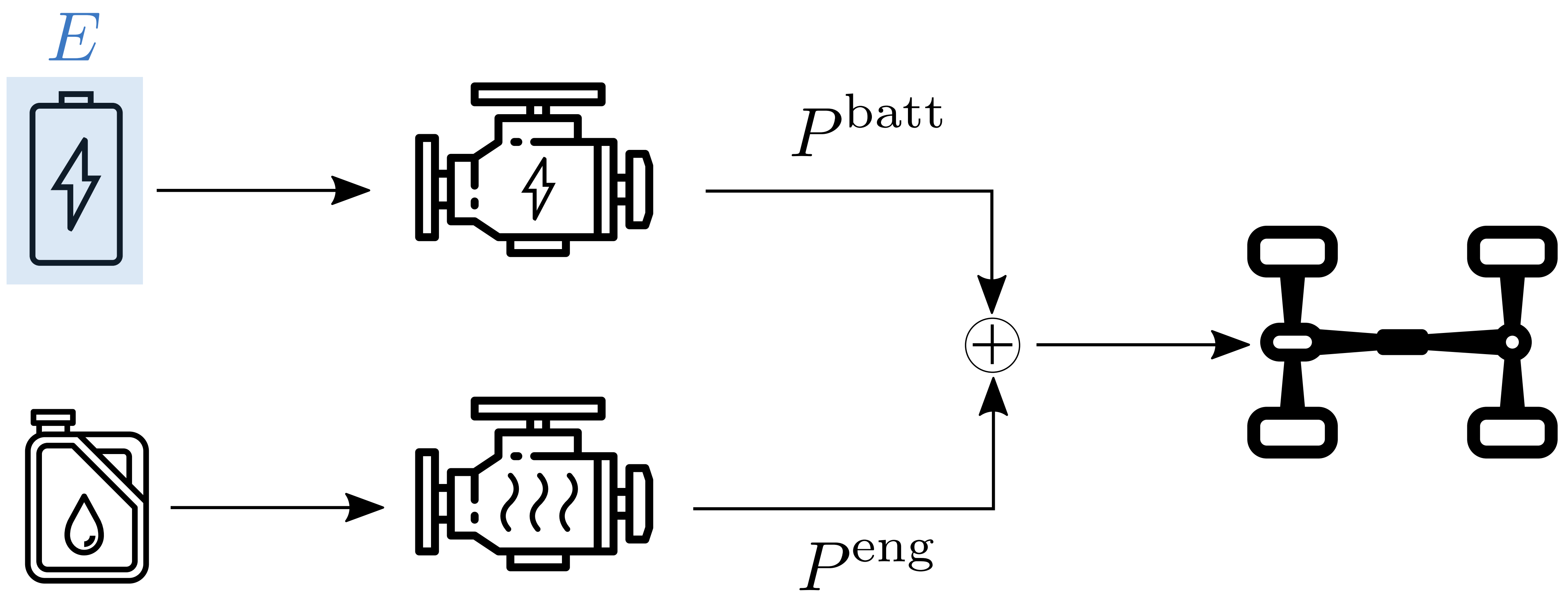
Low suboptimality and infeasibility

c	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}

Hybrid-vehicle control



Hybrid-vehicle control



Hybrid-vehicle control

$$\begin{aligned}
\text{minimize} \quad & \sum_{t=0}^{T-1} f(P_t^{\text{eng}}, z_t) + \delta(z_t - z_{t-1})_+ \\
\text{subject to} \quad & P_t^{\text{batt}} + P_t^{\text{eng}} \geq P_t^{\text{des}} \\
& E_{t+1} = E_t - \tau P_t^{\text{batt}} \\
& E_0 = E_{\text{init}} \\
& z_t \in \{0, 1\}
\end{aligned}$$

Hybrid-vehicle control

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

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

$$\begin{aligned} \text{minimize} \quad & \sum_{t=0}^{T-1} f(P_t^{\text{eng}}, z_t) + \delta(z_t - z_{t-1})_+ \\ \text{subject to} \quad & P_t^{\text{batt}} + P_t^{\text{eng}} \geq P_t^{\text{des}} \\ & E_{t+1} = E_t - \tau P_t^{\text{batt}} \\ & E_0 = E_{\text{init}} \\ & z_t \in \{0, 1\} \end{aligned}$$

Desired Power

Battery Dynamics

Hybrid-vehicle control

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

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

Hybrid-vehicle control

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

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

Desired Power

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

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

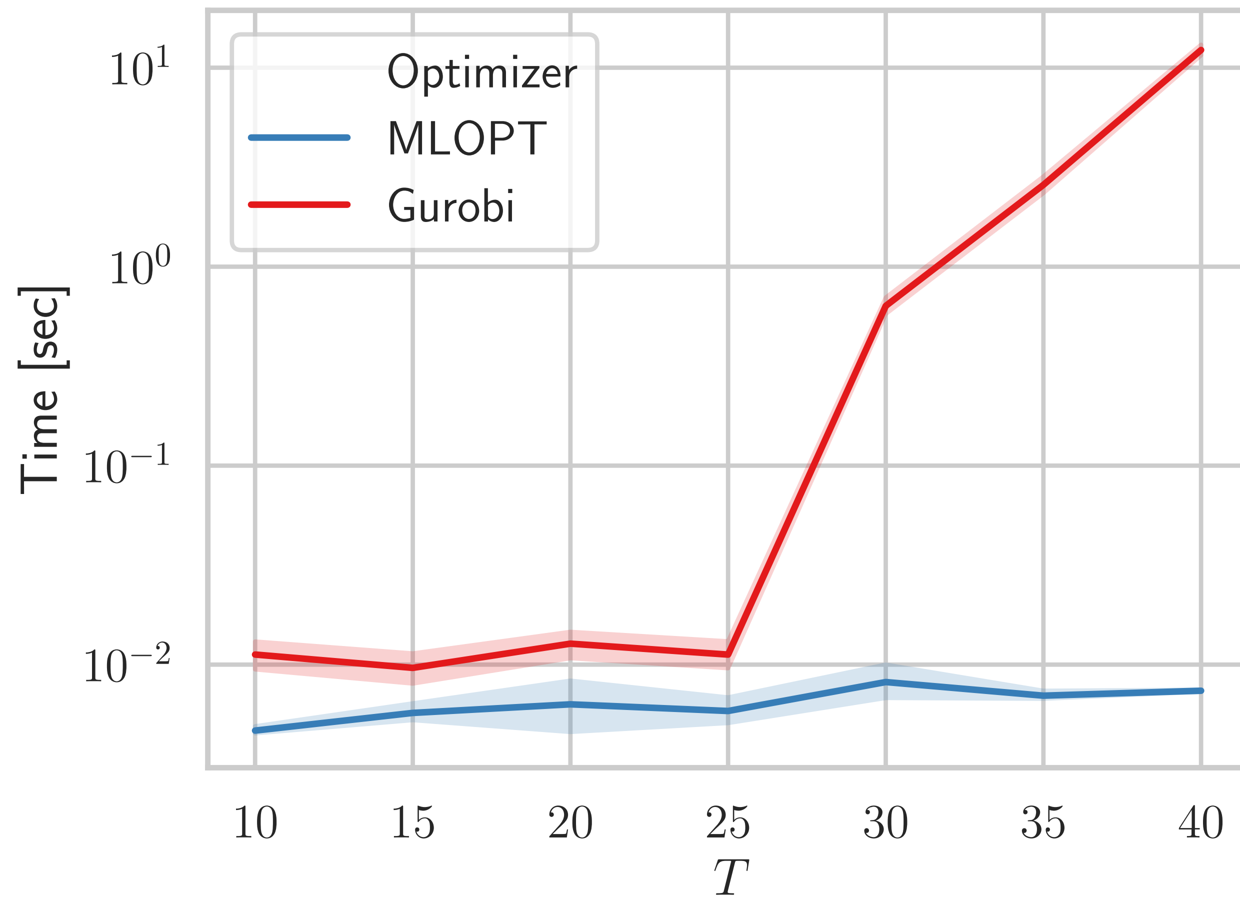
Battery Dynamics

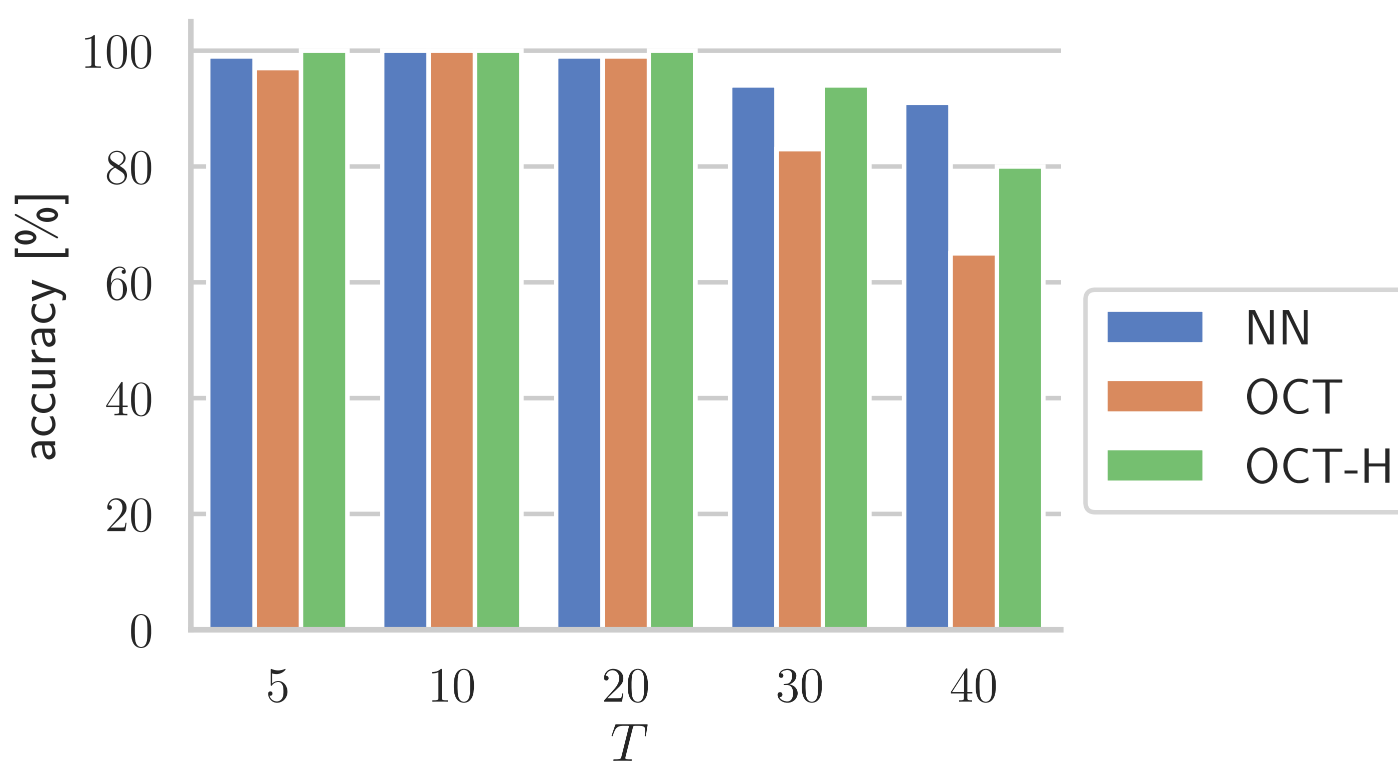
$$z_t \in \{0, 1\}$$

Engine Switch

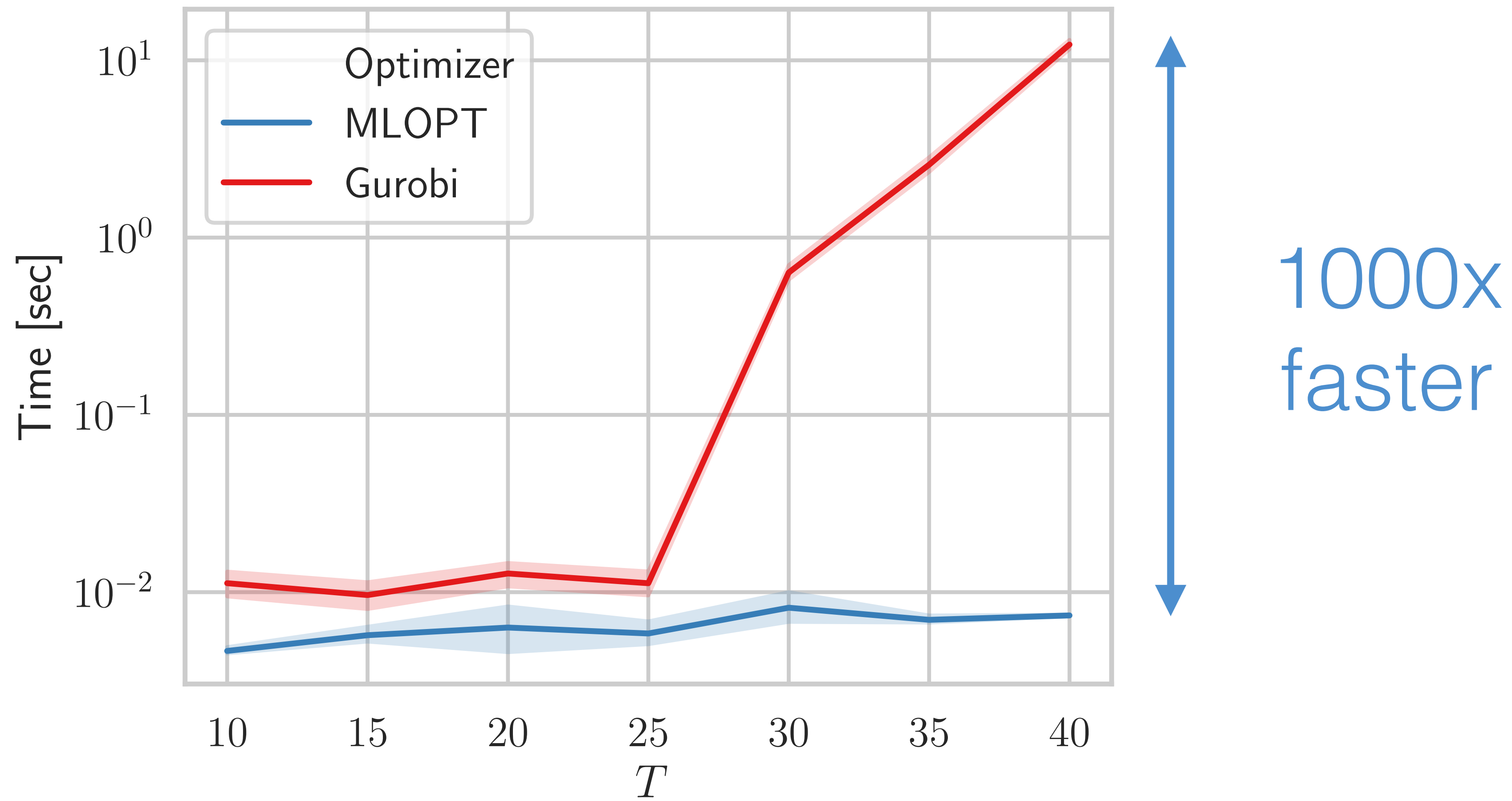
Parameters

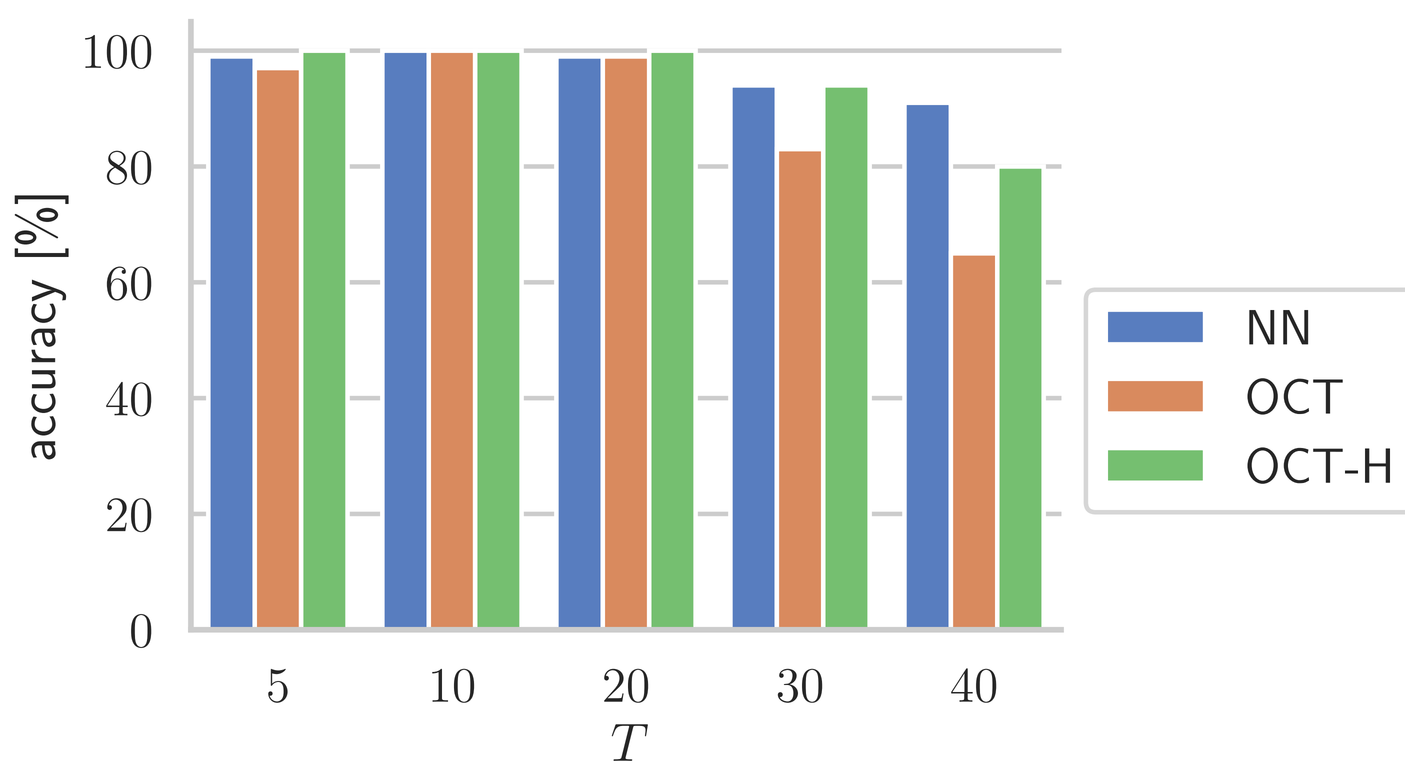
MPC results: high speed and accuracy



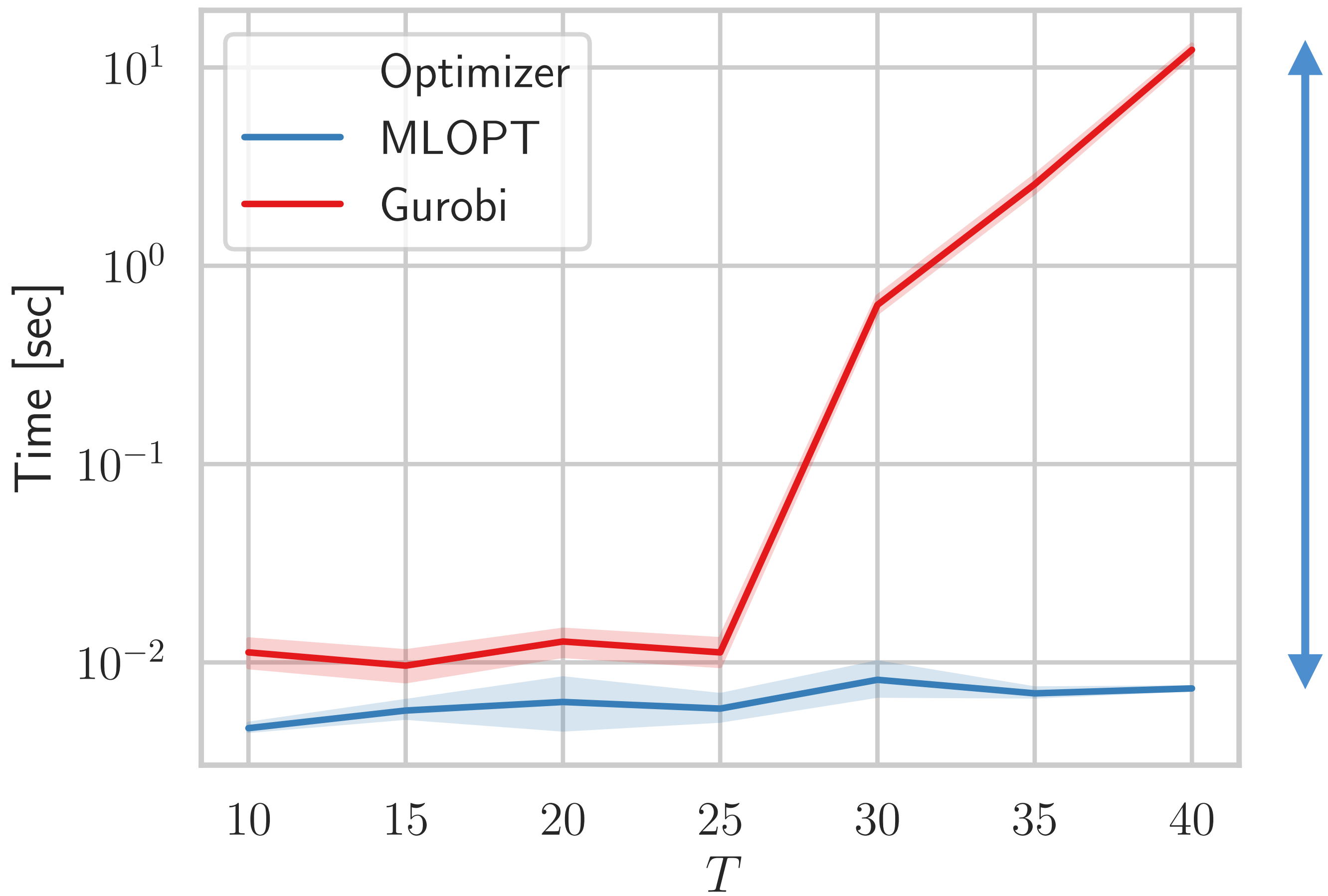


MPC results: high speed and accuracy



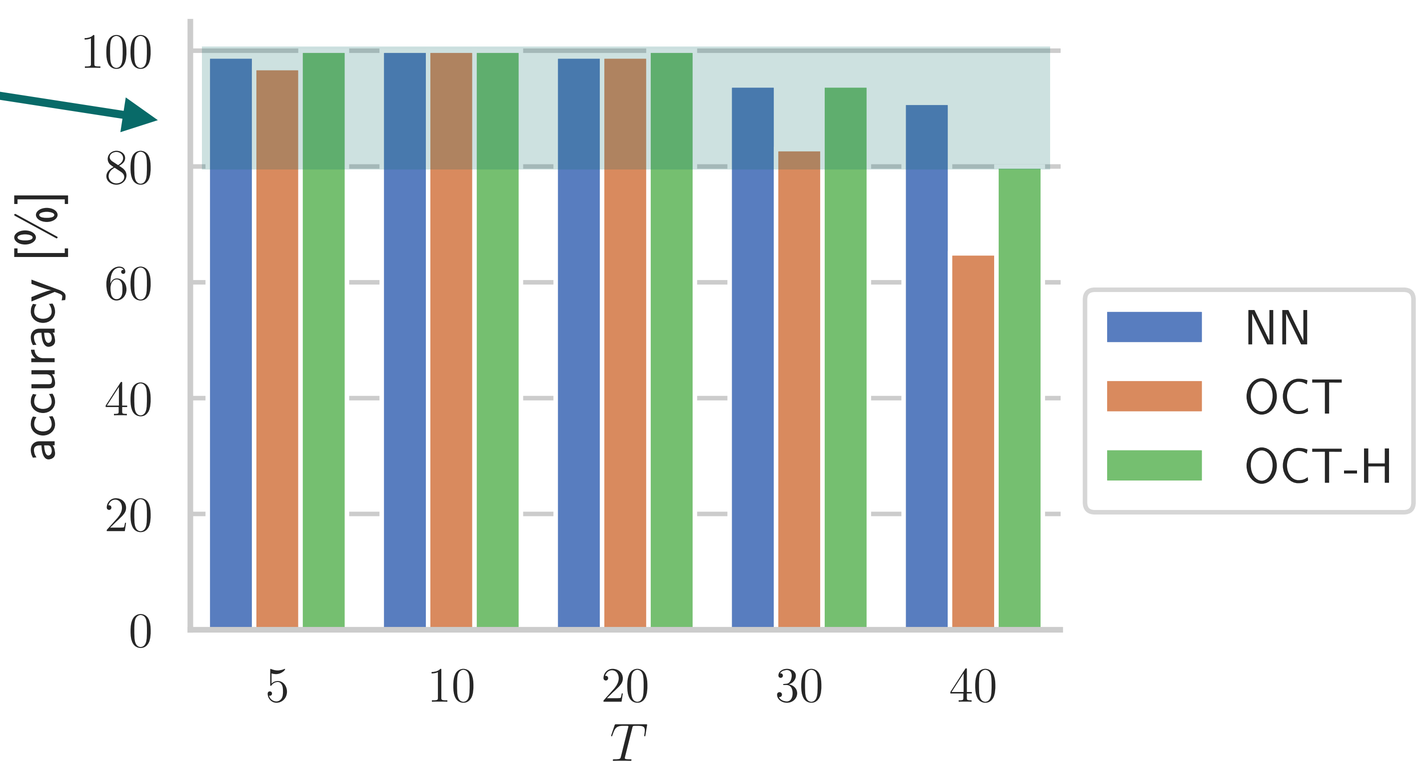


MPC results: high speed and accuracy

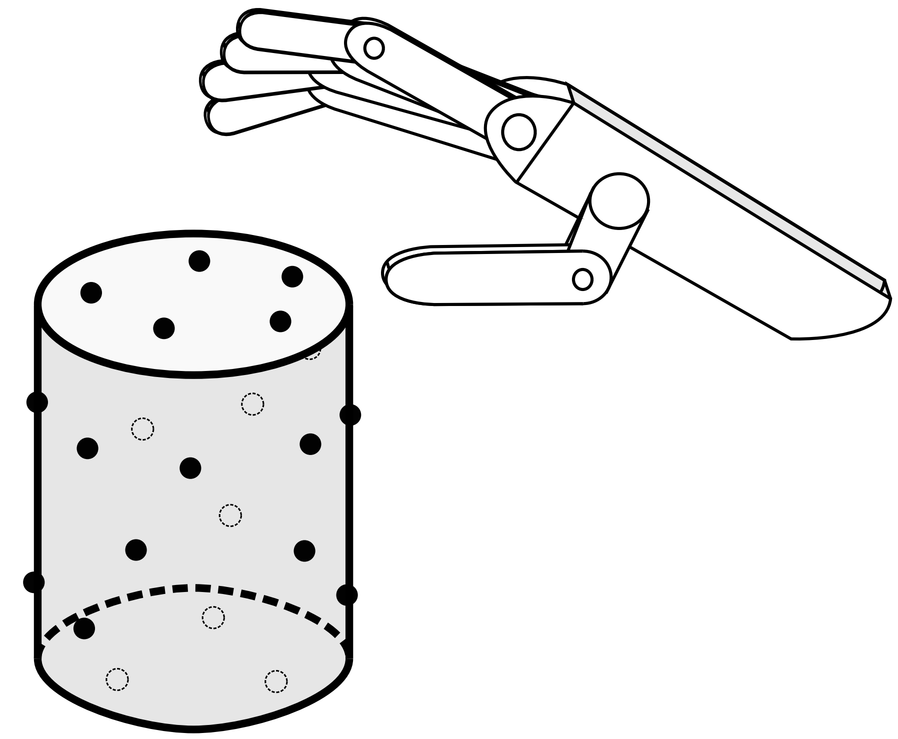


1000x faster

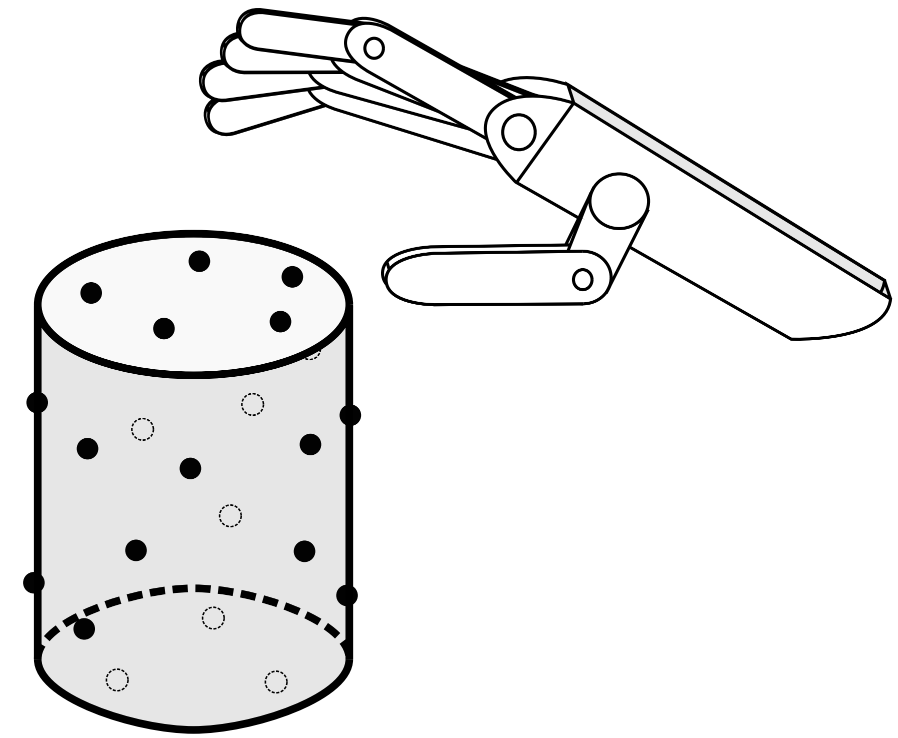
High accuracy



Dexterous Robotic Grasps



Dexterous Robotic Grasps



maximize α

subject to $Gf = \alpha \hat{F}$

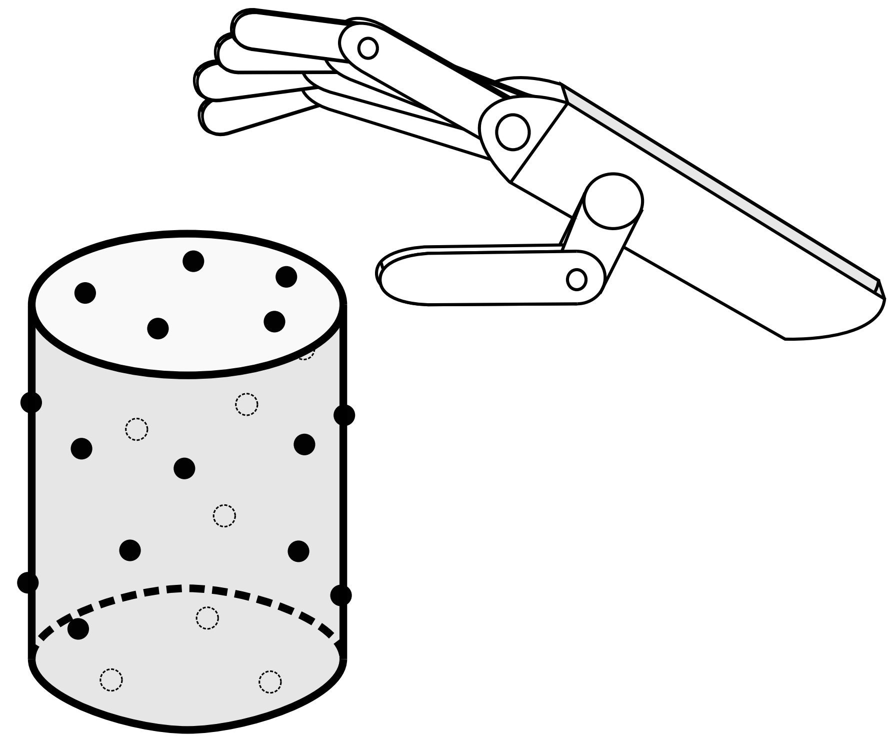
$$f_i \in \mathcal{K}^{(i)}, \quad i = 1, \dots P$$

$$f_i^z \leq \delta_i, \quad i = 1, \dots, P$$

$$\sum_{i=1}^{P} \delta_i \leq k$$

$$\delta \in \{0, 1\}^P$$

Dexterous Robotic Grasps



Grasp metric

maximize α

subject to $Gf = \alpha \hat{F}$

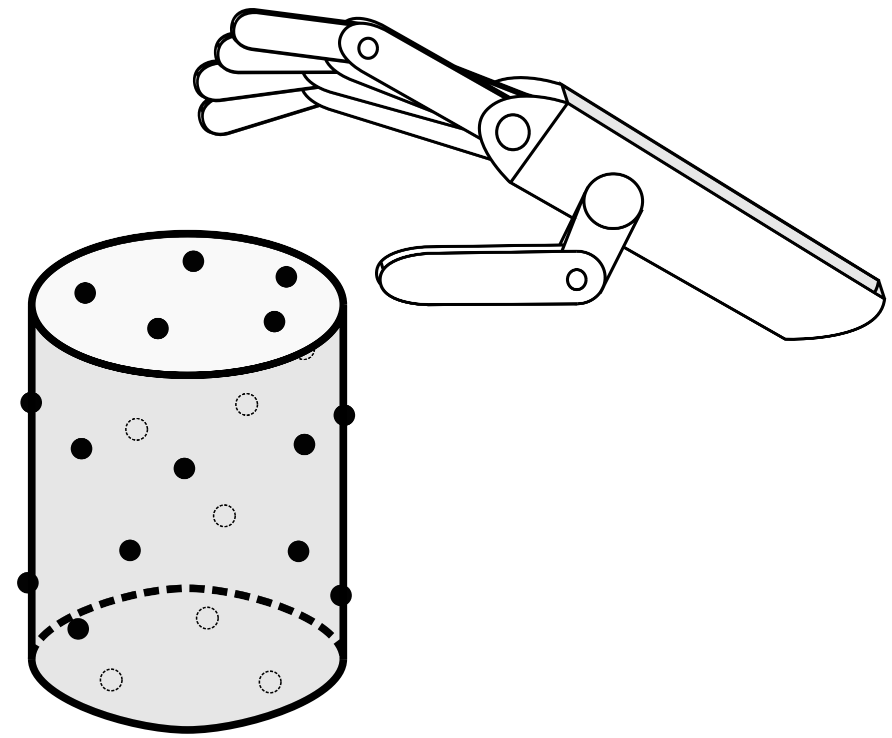
$$f_i \in \mathcal{K}^{(i)}, \quad i = 1, \ldots P$$

$$f_i^z \leq \delta_i, \quad i = 1, \ldots, P$$

$$\sum_{i=1}^{P} \delta_i \leq k$$

$$\delta \in \{0, 1\}^P$$

Dexterous Robotic Grasps



Grasp metric

maximize α

Task wrench

subject to $Gf = \alpha \hat{F}$

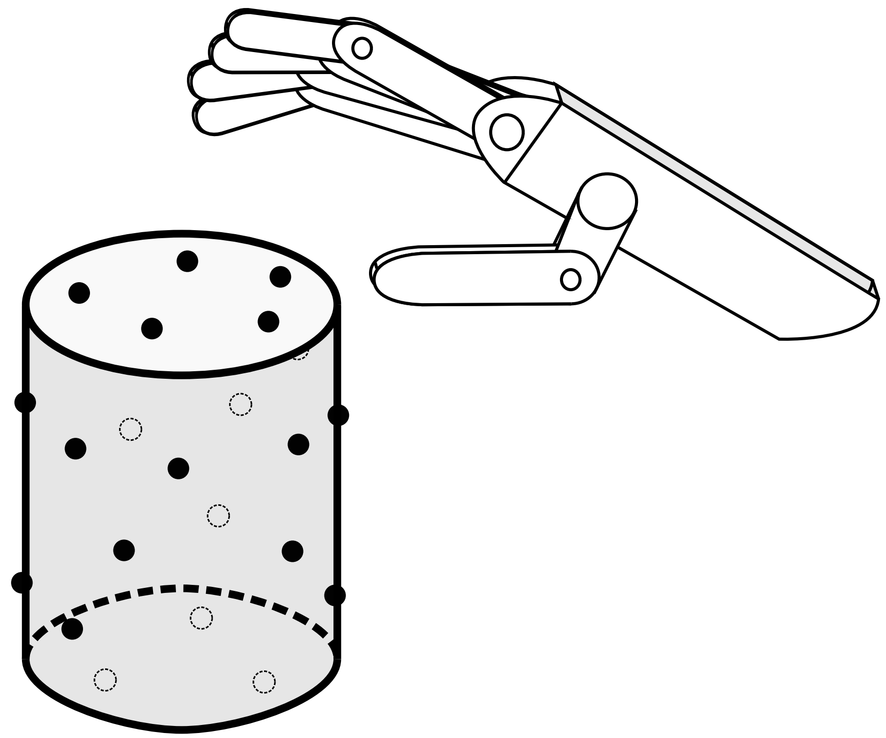
$$f_i \in \mathcal{K}^{(i)}, \quad i = 1, \dots P$$

$$f_i^z \le \delta_i, \quad i = 1, \dots, P$$

$$\sum_{i=1}^{P} \delta_i \le k$$

$$\delta \in \{0, 1\}^P$$

Dexterous Robotic Grasps



Grasp metric

maximize α

Task wrench

subject to $Gf = \alpha \hat{F}$

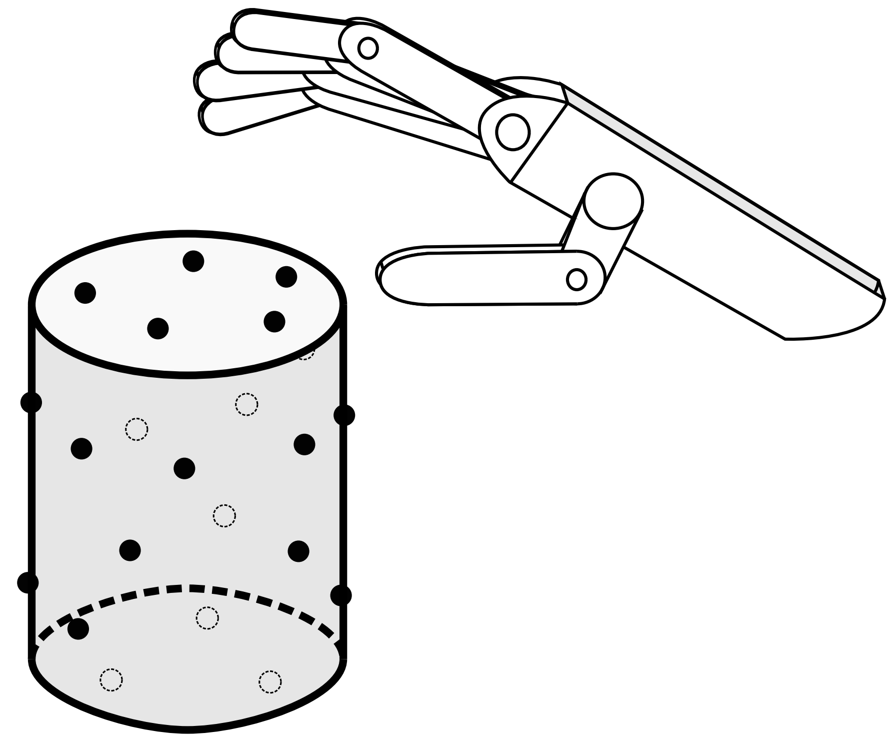
$f_i \in \mathcal{K}^{(i)}, \quad i = 1, \dots P$ Friction Cone

$f_i^z \leq \delta_i, \quad i = 1, \dots, P$

$\sum_{i=1}^{P} \delta_i \leq k$

$\delta \in \{0, 1\}^P$

Dexterous Robotic Grasps



Grasp metric

maximize α

Task wrench

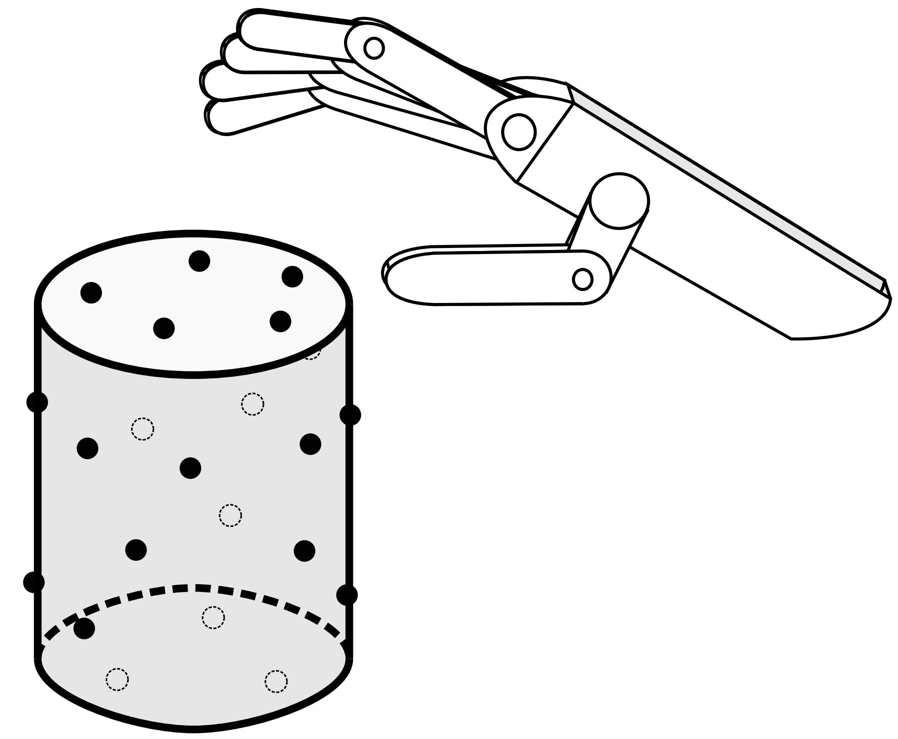
subject to $Gf = \alpha \hat{F}$

$f_i \in \mathcal{K}^{(i)}, \quad i = 1, \ldots P$ Friction Cone

$$f_i^z \leq \delta_i, \quad i = 1, \ldots, P$$
$$\sum_{i=1}^{P} \delta_i \leq k$$
$$\delta \in \{0, 1\}^P$$

Points Selection

Dexterous Robotic Grasps



Grasp metric

maximize α

Task wrench

subject to $Gf = \alpha \hat{F}$

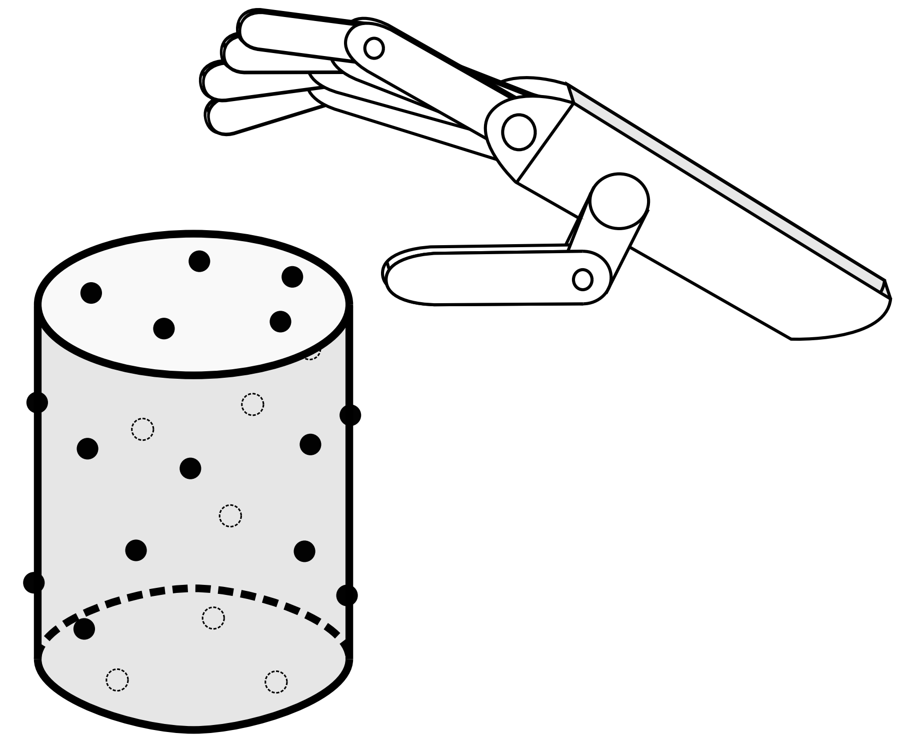
$f_i \in \mathcal{K}^{(i)}, \quad i = 1, \ldots P$ Friction Cone

$$f_i^z \leq \delta_i, \quad i = 1, \ldots, P$$
$$\sum_{i=1}^{P} \delta_i \leq k$$
$$\delta \in \{0, 1\}^P$$

Points Selection

MISOCP

Dexterous Robotic Grasps



Grasp metric

maximize α

Task wrench

subject to $Gf = \alpha \hat{F}$

$f_i \in \mathcal{K}^{(i)}, \quad i = 1, \dots P$ Friction Cone

$f_i^z \leq \delta_i, \quad i = 1, \dots, P$

$\sum_{i=1}^{P} \delta_i \leq k$

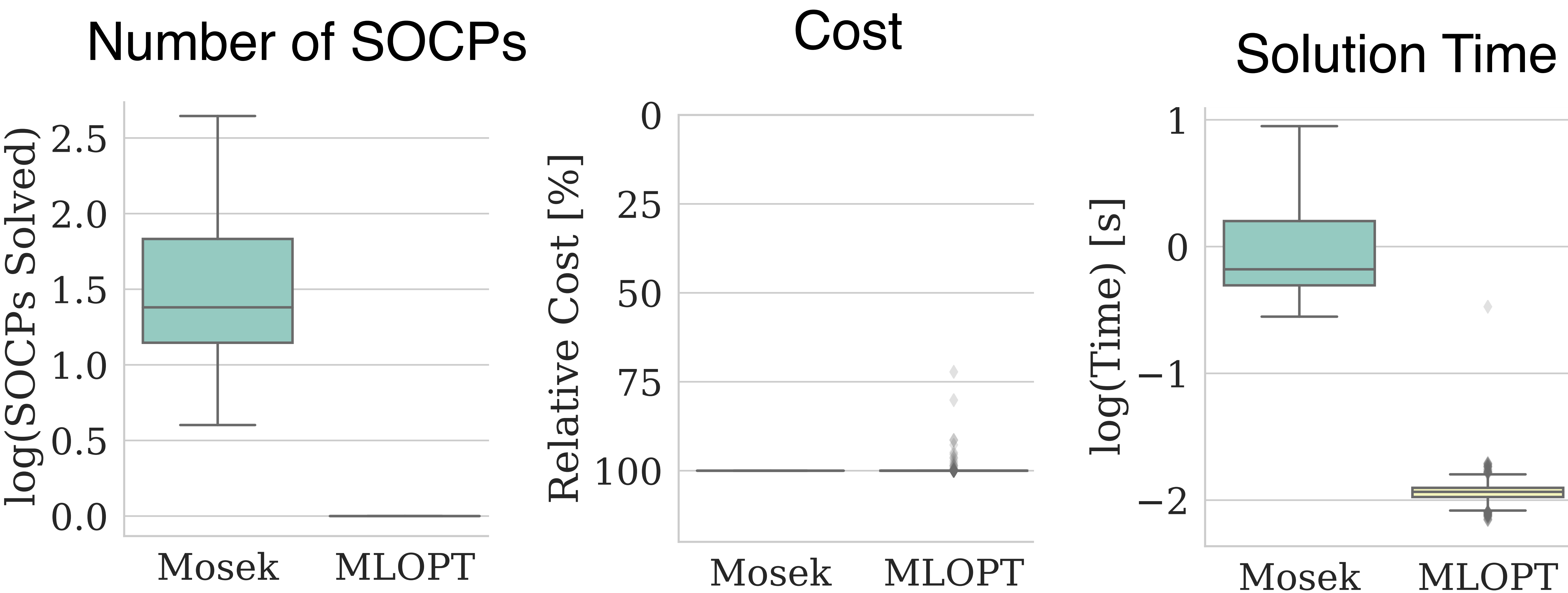
$\delta \in \{0, 1\}^P$

Points Selection

MISOCP

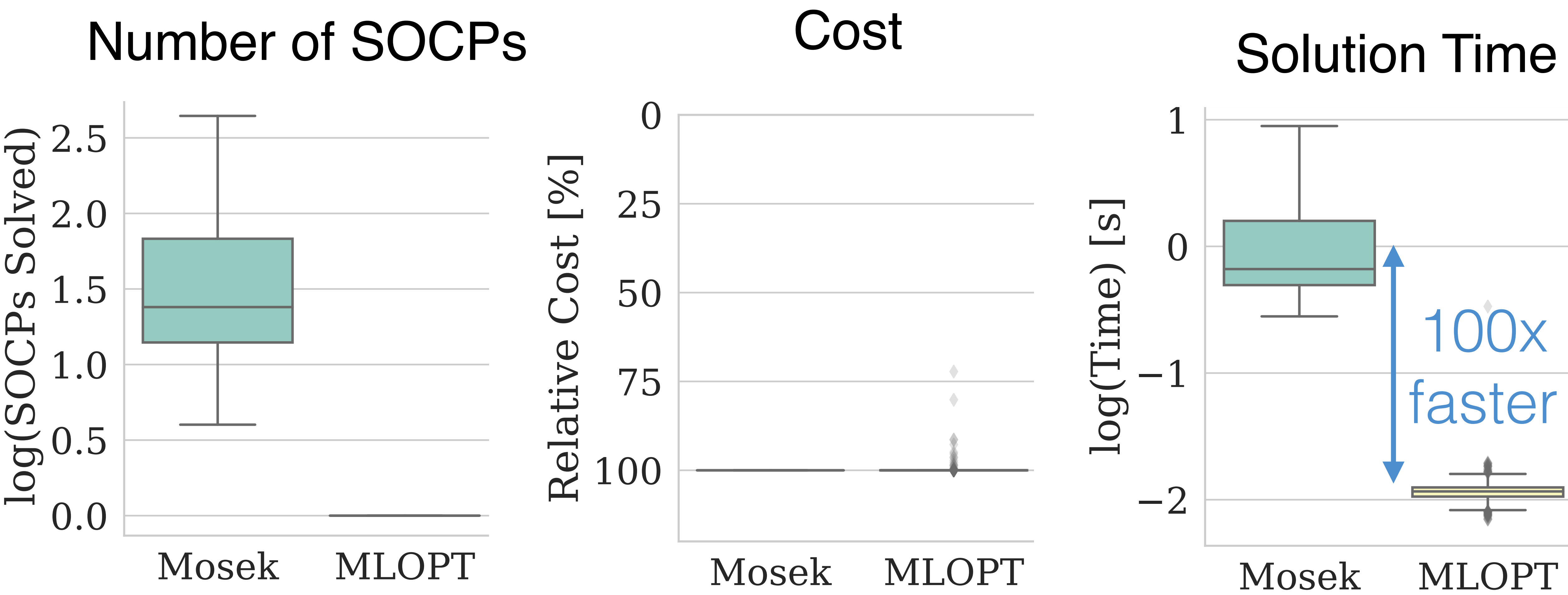
Parameters

High-speed Dexterous Robotic Grasps



[A. Cauligi, P. Culbertson, B. Stellato, D. Bertsimas, M. Schwager, M. Pavone, *Learning Mixed-Integer Convex Optimization Strategies for Robot Planning and Control*, arXiv:1907.02206, (2019)]

High-speed Dexterous Robotic Grasps



[A. Cauligi, P. Culbertson, B. Stellato, D. Bertsimas, M. Schwager, M. Pavone, ***Learning Mixed-Integer Convex Optimization Strategies for Robot Planning and Control***, arXiv:1907.02206, (2019)]

Conclusions

Acknowledgements

Dimitris Bertsimas

MIT

Abhishek Cauligi

Stanford University

Preston Culbertson

Stanford University

Marco Pavone

Stanford University

Mac Schwager

Stanford University

Machine Learning for Optimization

Optimization links data to decisions

Machine Learning for Optimization

Optimization links data to decisions

Understand it using ML

Machine Learning for Optimization

Optimization links data to decisions

Understand it using ML

Solve it very fast

Machine Learning for Optimization

Optimization links data to decisions

Understand it using ML

Solve it very fast

stellato.io

stellato@mit.edu

@b_stellato

github.com/bstellato/mlopt

Machine Learning for Optimization

Optimization links data to decisions

Understand it using ML

Solve it very fast

stellato.io

stellato@mit.edu

@b_stellato

github.com/bstellato/mlopt

[D. Bertsimas, B. Stellato, *The Voice of Optimization*, Machine Learning (to appear), (2020)]

[D. Bertsimas, B. Stellato, *Online Mixed-Integer Optimization in Milliseconds*, INFORMS Journal on Computing (under review), (2019)]

[A. Cauligi, P. Culbertson, B. Stellato, D. Bertsimas, M. Schwager, M. Pavone, *Learning Mixed-Integer Convex Optimization Strategies for Robot Planning and Control*, arXiv:1907.02206, (2019)]

Backup

Sampling scheme

Algorithm 1 Strategies exploration

given $\epsilon, \beta, \Theta = \emptyset, \mathcal{S} = \emptyset, u = \infty$
for $k = 1, \ldots,$ **do**
  Sample θ_k and compute $s(\theta_k)$ ▷ Sample parameter and strategy.
  $\Theta \leftarrow \Theta \cup \{\theta_k\}$ ▷ Update set of samples.
  if $s(\theta_k) \notin \mathcal{S}$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{s(\theta_k)\}$ ▷ Update strategy set if new strategy found
  end if
  if $G + c\sqrt{(1/k)\ln(3/\beta)} \leq \epsilon$ **then** ▷ Break if bound less than ϵ
    break
  end if
end for
return $k, \Theta, \mathcal{S}$