Optimization in the Presence of Model Ambiguity in Markov Decision Processes

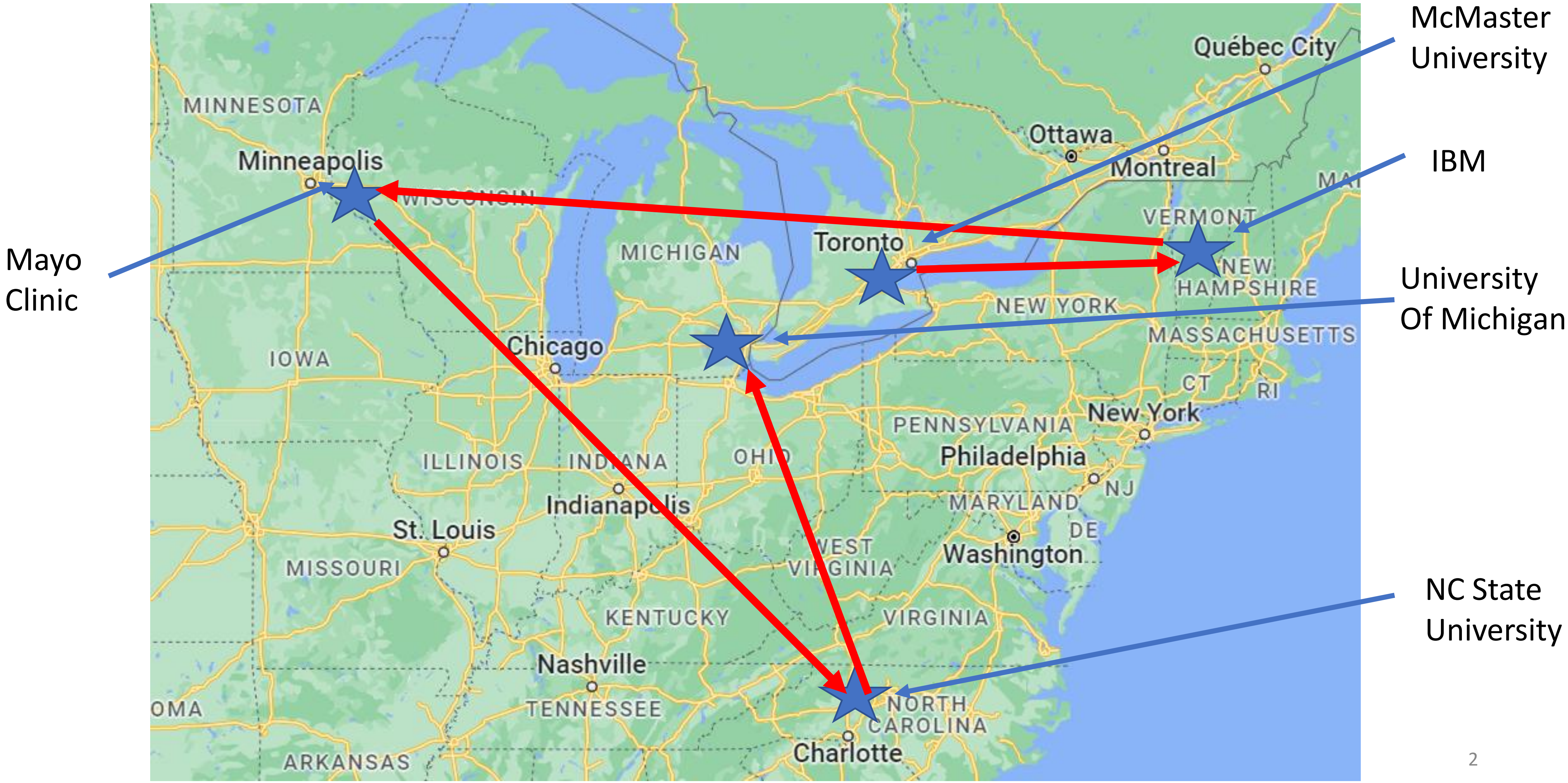
Brian Denton

Department of Industrial and Operations Engineering

University of Michigan

(Work with **Lauren Steimle**, UM/GA Tech, and **David Kaufman** UM-Dearborn)

My background



Sequential decision-making under uncertainty

Finance

Inventory management

Machine maintenance

Medical decision making

Prevention of cardiovascular disease (CVD) involves balancing the benefits and harms of treatment

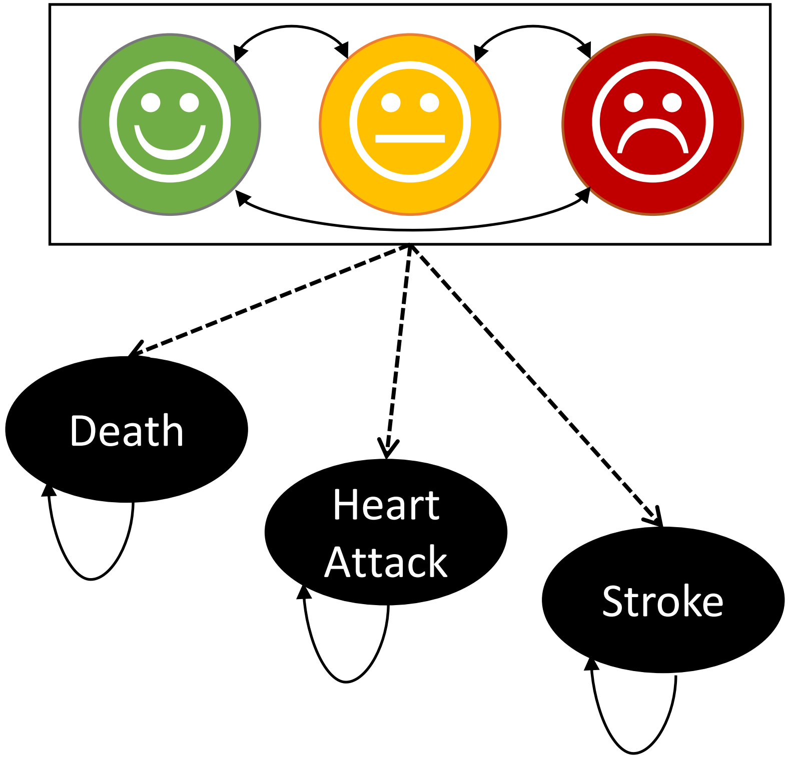
Uncertain Future Benefits

- Delay the onset of potentially deadly and debilitating heart attacks and strokes

Immediate harms

- Side effects (e.g., muscle pain, frequent urination)

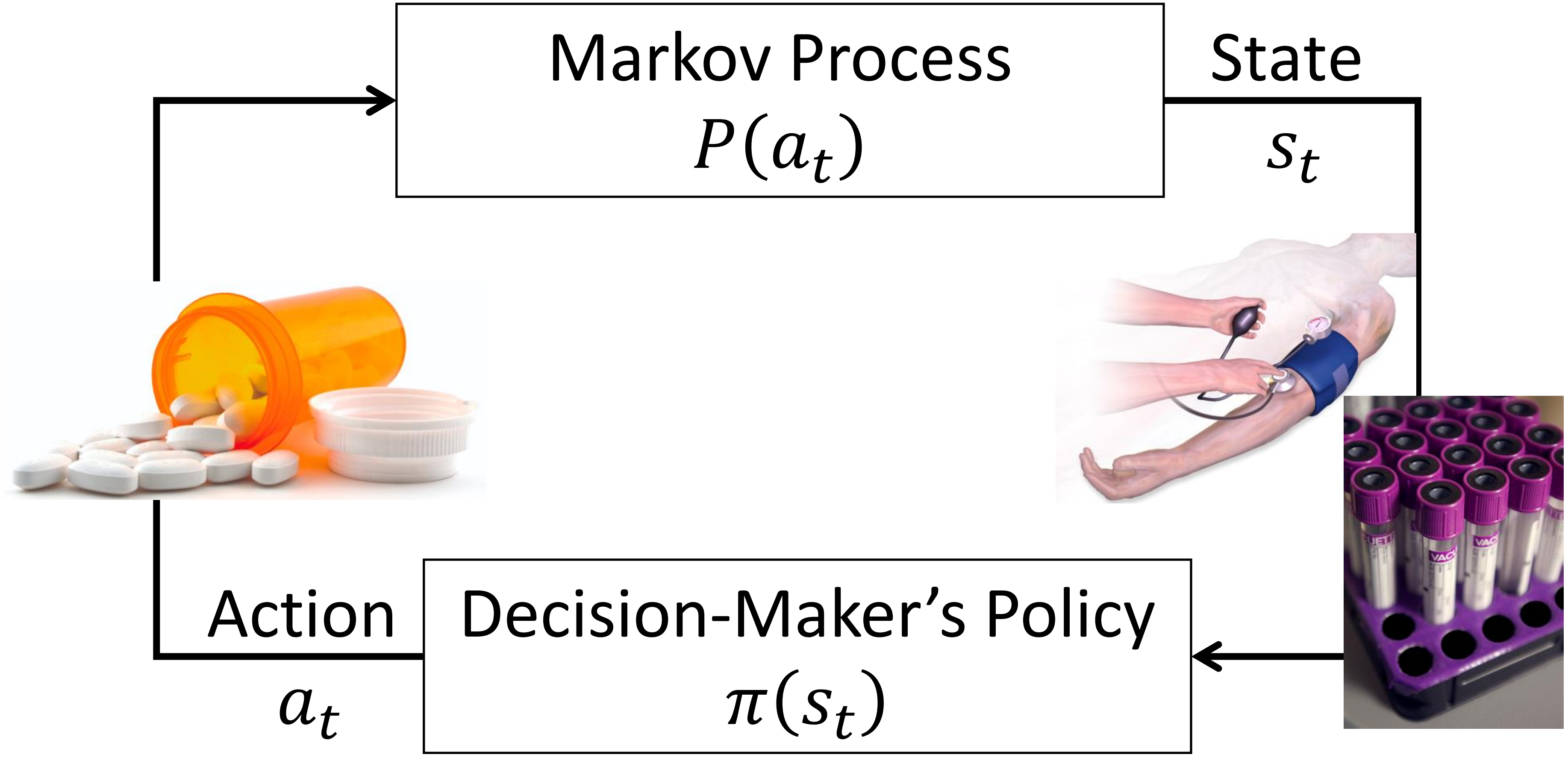
Markov decision processes generalize Markov chains to include decisions



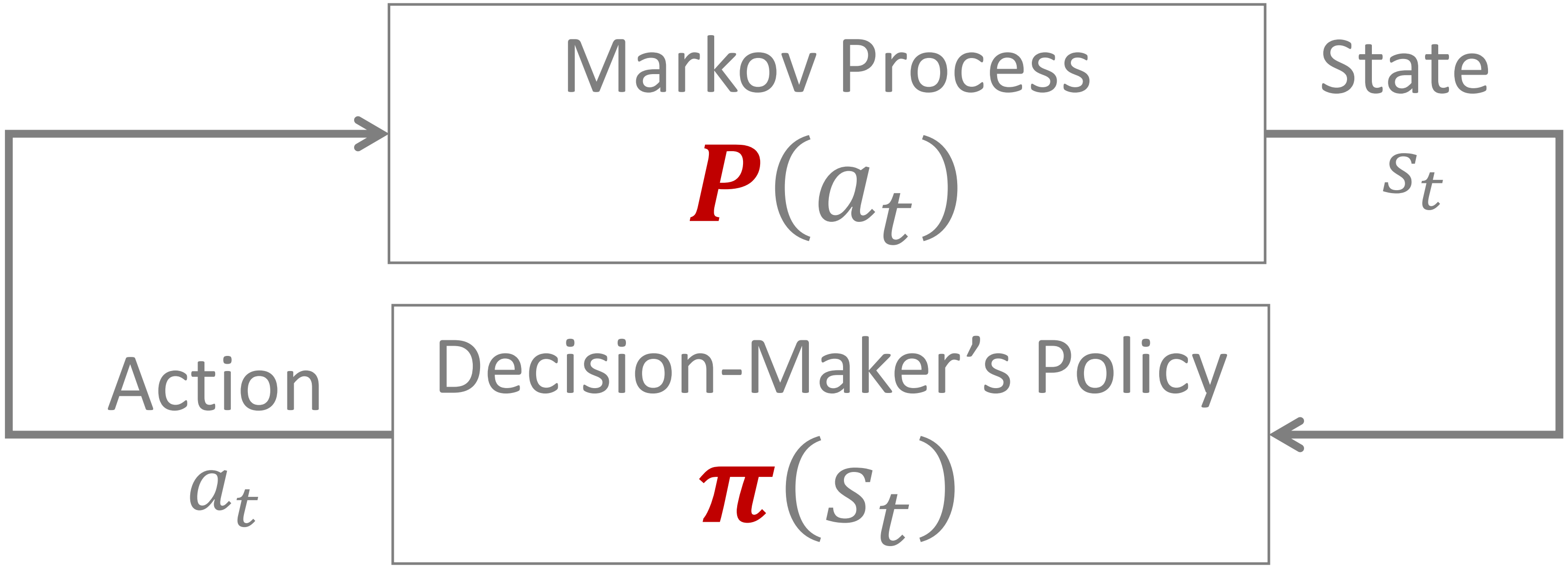
Health states
- Blood pressure levels
- Cholesterol levels
- Current medications

Steimle, L. N., & Denton, B. T. (2017). Markov decision processes for screening and treatment of chronic diseases. In *Markov Decision Processes in Practice* (pp. 189-222). Springer, Cham.

Markov decision process sequence of steps

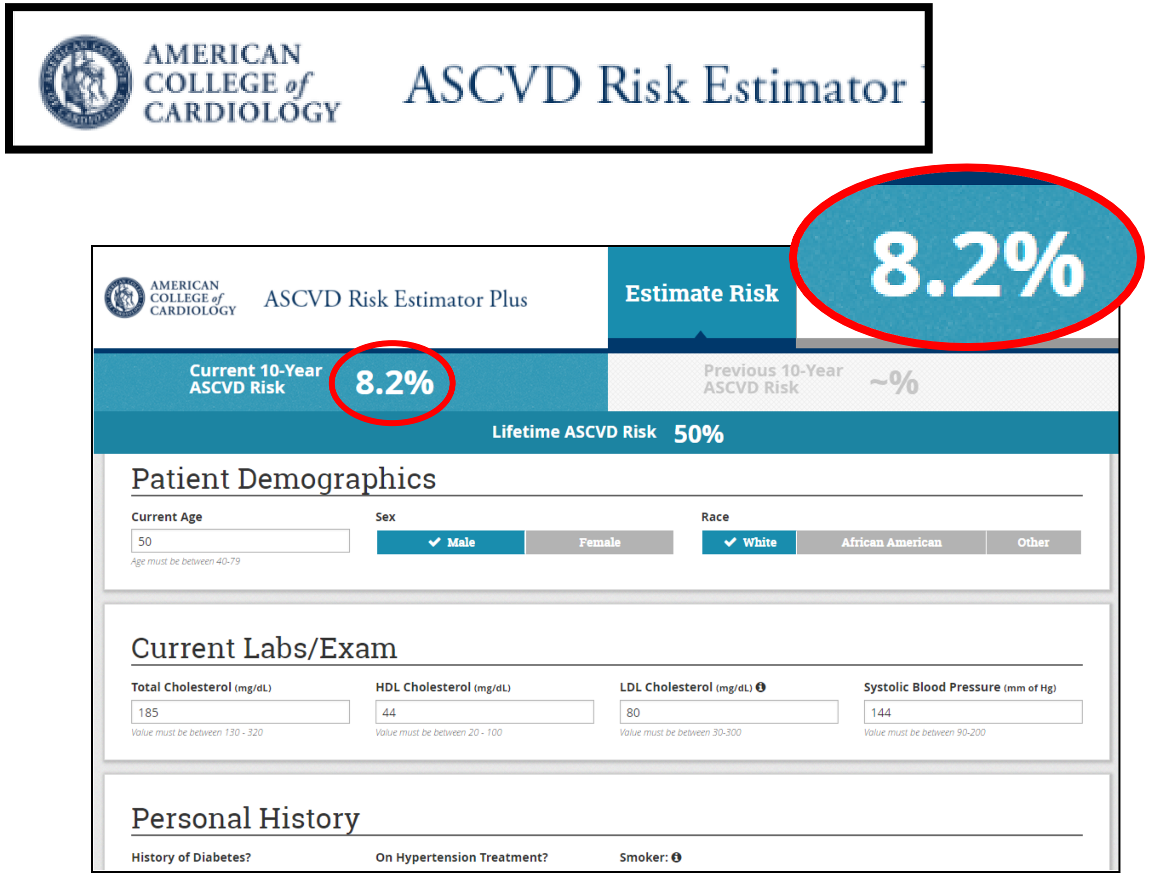


Markov decision process optimal policy



$$\max_{\pi \in \Pi} \left\{ \mathbb{E}^{\pi,P} \left[\sum_{t=1}^{T} r_t(s_t, a_t) + r_{T+1}(s_{T+1}) \right] \right\}$$

Clinical risk calculators are used to estimate a patient's risk



Inputs:

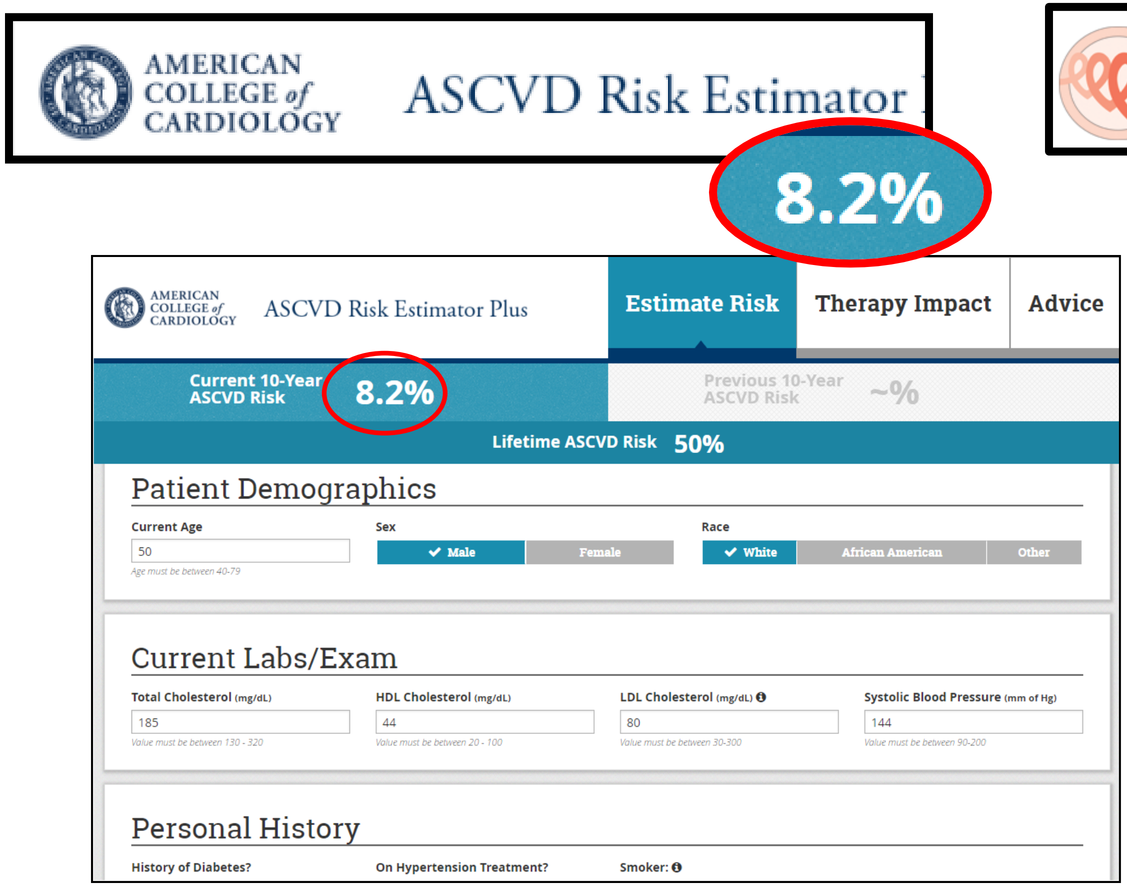
- Age
- Sex
- Race
- Cholesterol
- Blood Pressure
- History of Diabetes
- On Hypertensive Treatment
- Smoking status

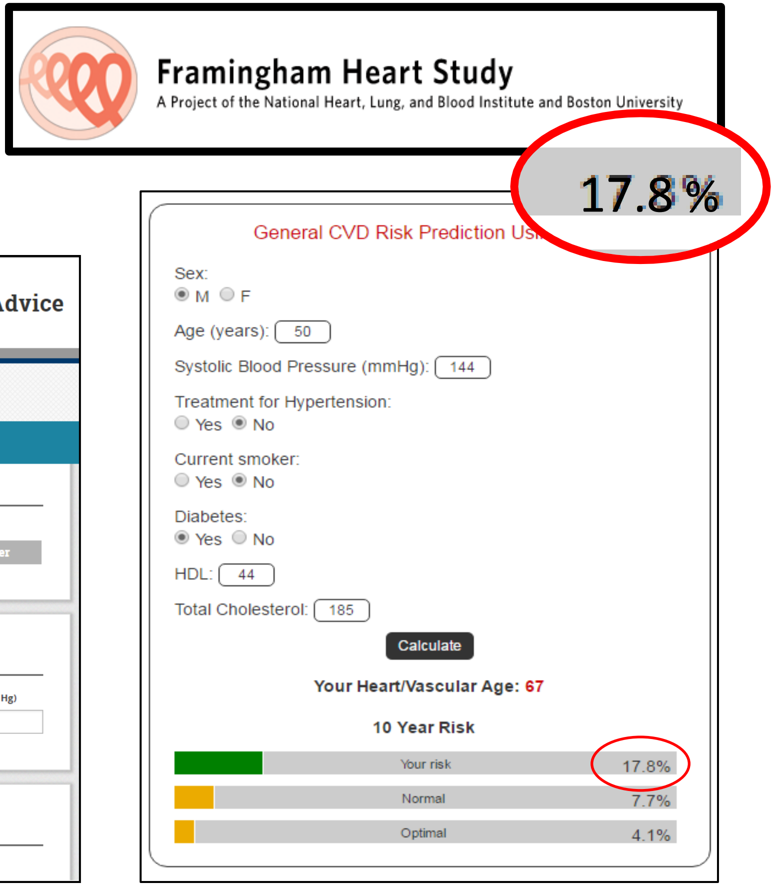
Output:

Current 10-Year Risk

2013 ACC/AHA Guideline on the Assessment of Cardiovascular Risk: A Report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines. 2014

Well-established clinical studies give conflicting estimates about CVD risk





1 Wilson et. al. Prediction of Coronary Heart Disease Using Risk Factor Categories. *Circulation.* 1998
2 2013 ACC/AHA Guideline on the Assessment of Cardiovascular Risk: A Report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines. 2014

Research Questions

How can we improve Markov decision processes to account for model ambiguity?

How much benefit is there really?

The remainder of this presentation

Multi-model Markov decision processes

Branch-and-bound algorithms

Alternative ambiguity-aware formulations

MMDPs have two layers of uncertainty

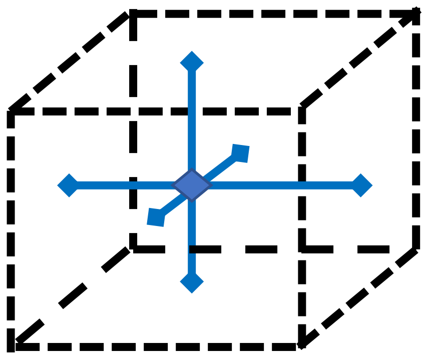
Optimal control of a stochastic system…

- Markov decision processes

…under model uncertainty

- Robust optimization
- Stochastic optimization

Early robust optimization approaches to MDPs with model parameter uncertainty



Assume that *P* lies within some *ambiguity set*

e.g., Interval Model

Goal is to maximize worst-case performance

(s,a)-rectangularity property gives a tractable model for MDPs

Robust optimization approach to ambiguity in Markov decision processes can be modeled as a two-player zero-sum game

- Decision-maker selects an action to maximize expected rewards

- Adversary selects transition probabilities to minimize DM's expected rewards

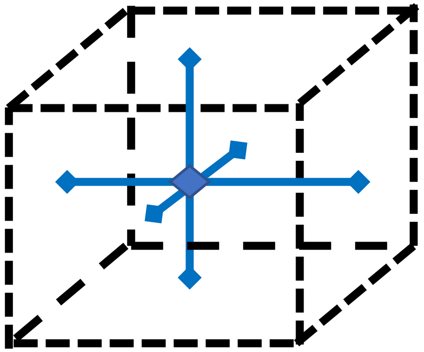
$$\max_{a \in \mathcal{A}} \min_{p_t(s,a) \in \mathcal{P}_t(s,a)} \{ r_t(s,a) + \sum_{s' \in \mathcal{S}} p_t(s'|s,a) v_{t+1}(s) \}$$

(s,a)-rectangularity property gives a tractable model based on the assumption the adversary can select each row independently

Nilim, A. and El Ghaoui, L. "Robust control of Markov decision processes with uncertain transition matrices." *Operations Research* 53.5 (2005): 780-798.

Iyengar, G. "Robust dynamic programming." *Mathematics of Operations Research* 30.2 (2005): 257-280.

(s,a)-rectangularity is computationally attractive, but has its drawbacks

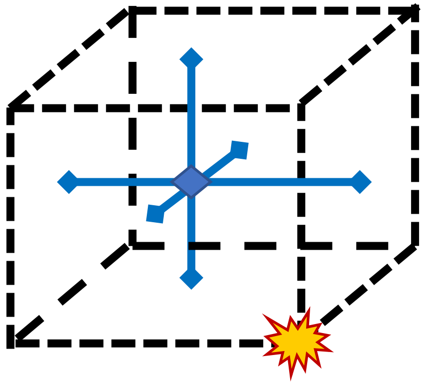


Leads to overly-protective policies

- Optimizing for cases where all parameters take on worst-case values simultaneously

Transition matrices might lose known structure

- Ambiguity is realized independently across states, actions, and/or decision epochs



Relaxing (s,a)-*rectangularity* causes the max-min problem to be NP-hard*

*Wiesemann, Wolfram, Daniel Kuhn, and Berç Rustem. "Robust Markov decision processes." *Mathematics of Operations Research* 38.1 (2013): 153-183.

Multi-model Markov Decision Process notation

Generalizes a standard Markov decision process

- State space, $\mathcal{S} \equiv \{1, \dots, S\}$
- Decision epochs, $\mathcal{T} \equiv \{1, \dots, T\}$
- Action space, $\mathcal{A} \equiv \{1, \dots, A\}$
- Rewards, $R \in \mathbb{R}^{S \times A \times T}$

Finite set of models, $\mathcal{M} = \{1, \dots, |\mathcal{M}|\}$

- Model m: An MDP $(\mathcal{S}, \mathcal{A}, \mathcal{T}, R, P^m)$
- Transition probabilities P^m are model-specific
- Model weights: $\lambda_1, \lambda_2, \dots, \lambda_{|\mathcal{M}|}$

Steimle, L. N., Kaufman, D.L., and Denton B.T. "Multi-model Markov Decision Processes." *IISE Transactions, 2021.*

The **weighted value problem** seeks to find a single policy that performs well in expectation

Performance of policy π in model m:

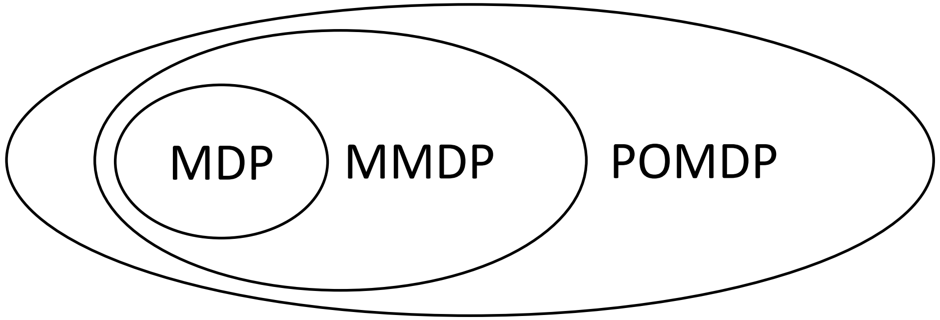
$$v^m(\pi) = \mathbb{E}^{\pi,P^m}\left[\sum_{t=1}^{T} r_t(s_t, a_t) + r_{T+1}(s_{T+1})\right]$$

Weighted value of policy π:

$$W(\pi) = \sum_{m\in\mathcal{M}} \lambda_m v^m(\pi)$$

Weighted value problem: $W^* = \max_{\pi\in\Pi} W(\pi)$

The weighted value problem is hard



The MMDP is a special case of a partially-observable MDP.

Proposition: The optimal policy may be history-dependent.

Proof by contradiction

Proposition: In general, the Weighted Value Problem is PSPACE-hard.

Reduction from *Quantified Satisfiability*

Special case of an MMDP with deterministic Markov policies

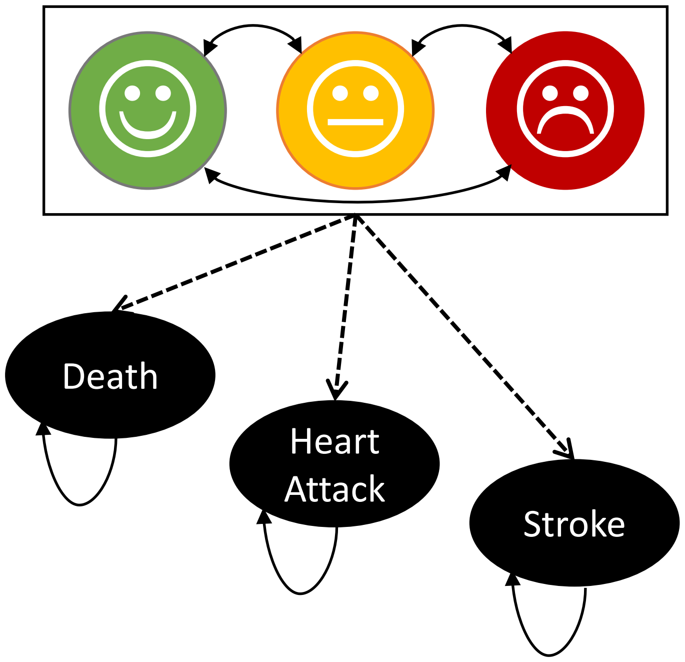
Proposition: There exists a deterministic policy that is optimal when restricting to Markov policies

Proposition: The Weighted Value Problem for Markov deterministic policies is NP-hard

Reduction from 3-CNF-SAT

Initially, we focused on finding near-optimal Markov deterministic policies, $\pi \in \Pi^{\mathrm{MD}}$, using a polynomial time approximation.

Example: approximation algorithm for cardiovascular disease prevention MMDP



Multi-model Markov decision process

- 4,096 states
- 64 actions
- 40 decision epochs
- 2 models

Case study data

- Longitudinal data from Mayo Clinic
- Framingham, ACC risk calculators
- Disutilities from medical literature

Mason, J. E., Denton, B. T., Shah, N. D., & Smith, S. A. (2014). Optimizing the simultaneous management of blood pressure and cholesterol for type 2 diabetes patients. *European Journal of Operational Research*, *233*(3), 727-738.

We compared our approximation algorithm policy to policies that ignore model ambiguity

Quality-Adjusted Life Years Gained
Over No Treatment, per 1000 Men

Optimal Decisions for FHS Model

MMDP Decisions

Optimal Decisions for ACC Model

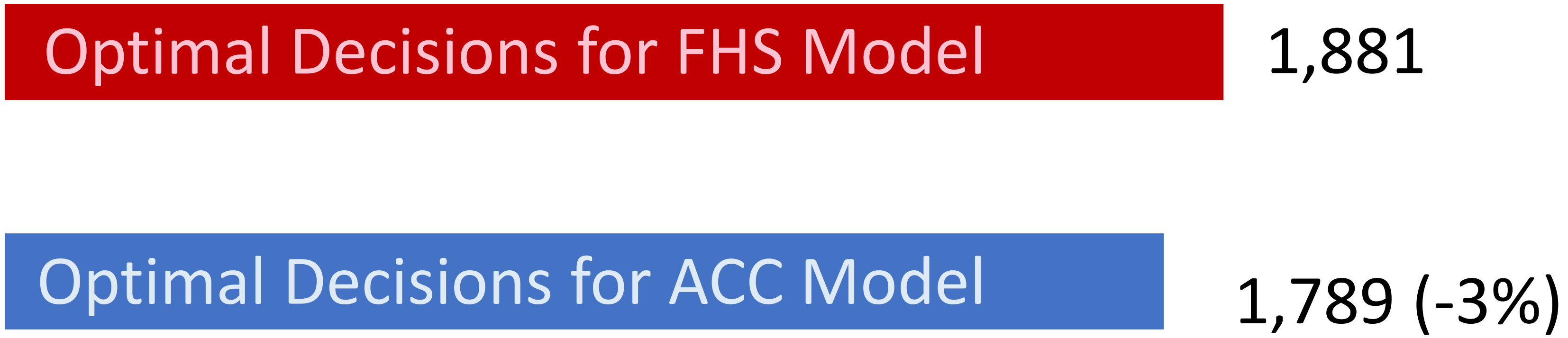
In some cases, ignoring ambiguity has relatively minor implications

Quality-Adjusted Life Years Gained
Over No Treatment, per 1000 Men

Framingham Heart Study Model

In some cases, ignoring ambiguity has relatively minor implications

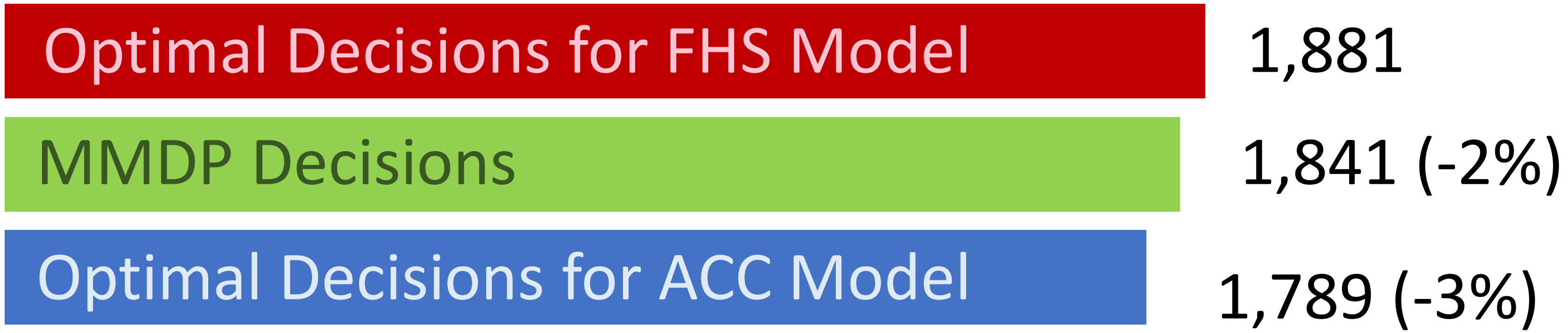
Quality-Adjusted Life Years Gained
Over No Treatment, per 1000 Men



Framingham Heart Study Model

In some cases, ignoring ambiguity has relatively minor implications

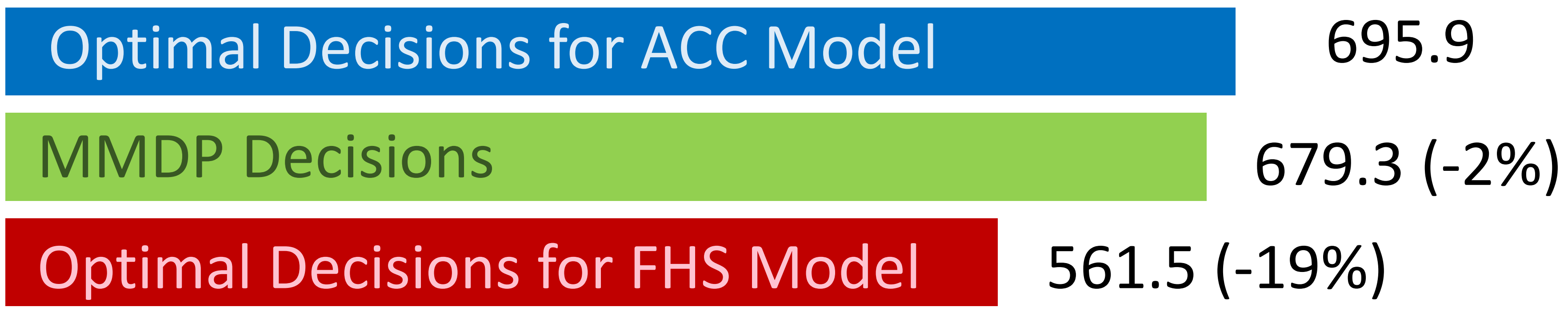
Quality-Adjusted Life Years Gained
Over No Treatment, per 1000 Men



Framingham Heart Study Model

But in other cases, ignoring ambiguity can have major implications

Quality-Adjusted Life Years Gained
Over No Treatment, per 1000 Men



American College of Cardiology Model

Observations

- MMDPs are difficult to solve computationally, but a polynomial-time approximation algorithm can provide near-optimal solutions in many instances

- Based on a CVD case study, it can be important to address ambiguity when there are multiple plausible models

The remainder of this presentation

Multi-model Markov decision processes

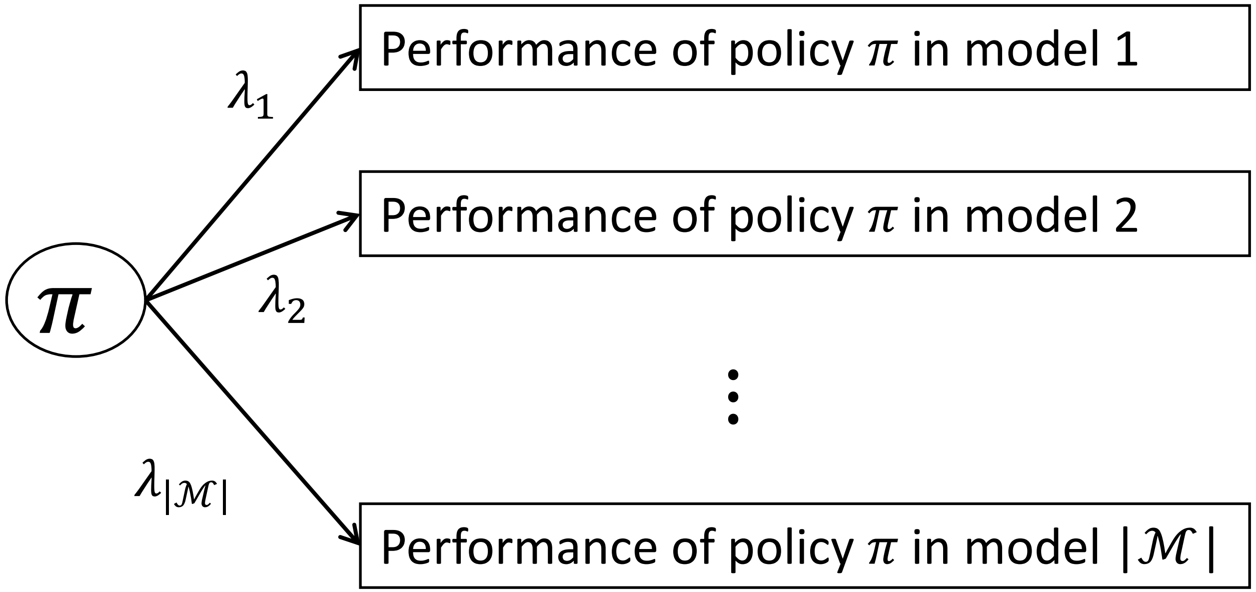
Branch-and-bound algorithms

Alternative ambiguity-aware formulations

Approaches to solve the weighted value problem

- Mixed-integer programming (MIP)
- Branch-and-cut on a 2-stage stochastic integer program formulation
- Custom branch-and-bound that exploits MMDP structure

The connection between MMDP and two-stage stochastic program



Stochastic program	MMDP
Scenarios	Model of MDP
Binary first-stage decision variables	Policy
Continuous second-stage decision variables	MDP model value functions

The MMDP is largely decomposable but…

Big-M's in logic-based constraints cause difficulty for standard stochastic programming methods

- Weak linear programming relaxation for the MIP
- Weak optimality cuts in Benders Decomposition

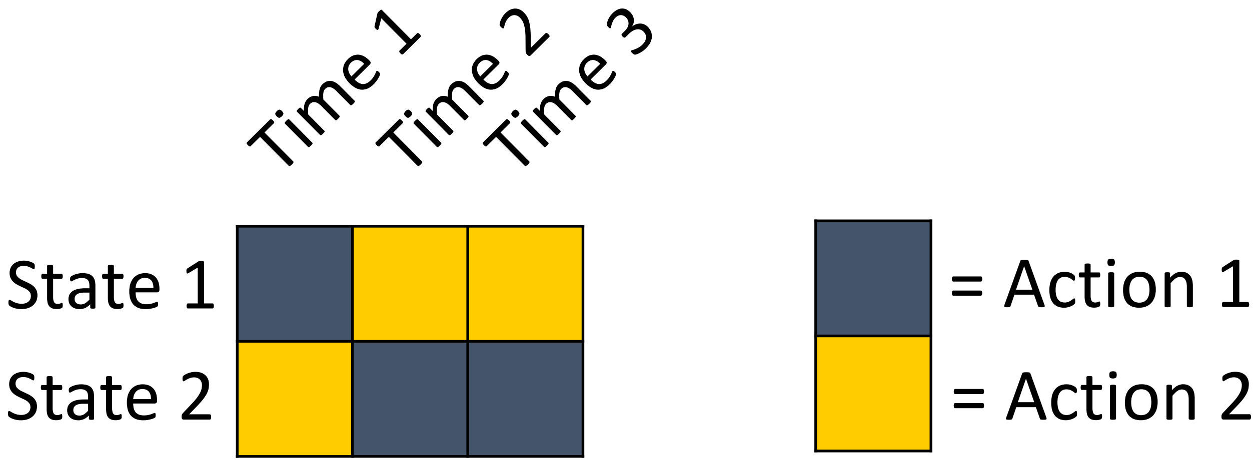
Problem is very decomposable

- Evaluation of a fixed policy is easily done by solving $|\mathcal{M}|$ independent MDPs

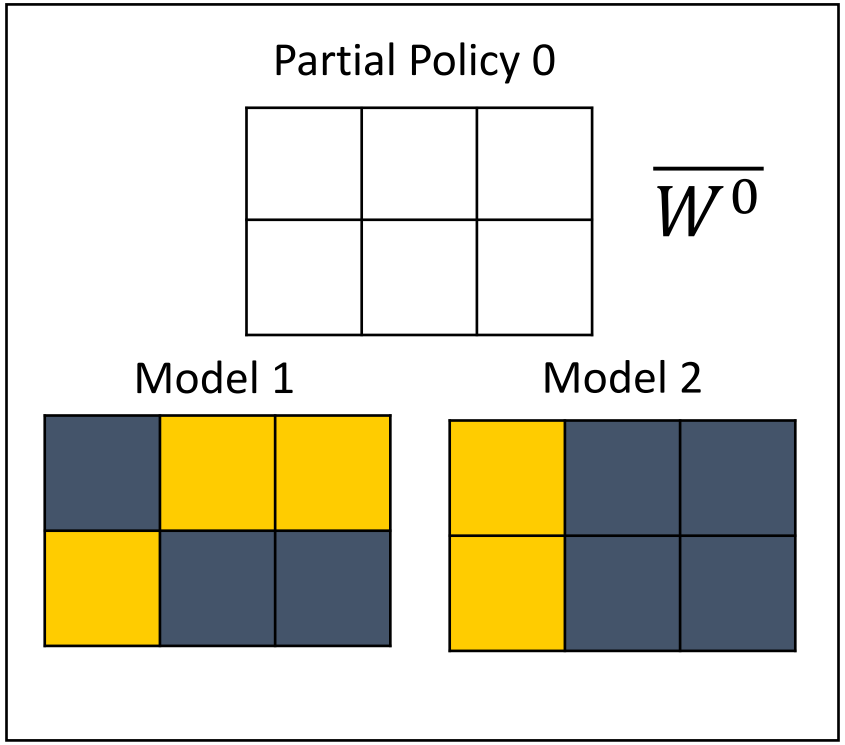
Branch-and-bound searches for policies that match across all models

Root Node: Relax requirement that policy must be same in each model

Goal: Find an *implementable policy* (policy is the same in all models) that maximizes weighted value



Branch & Bound begins by solving each model independently

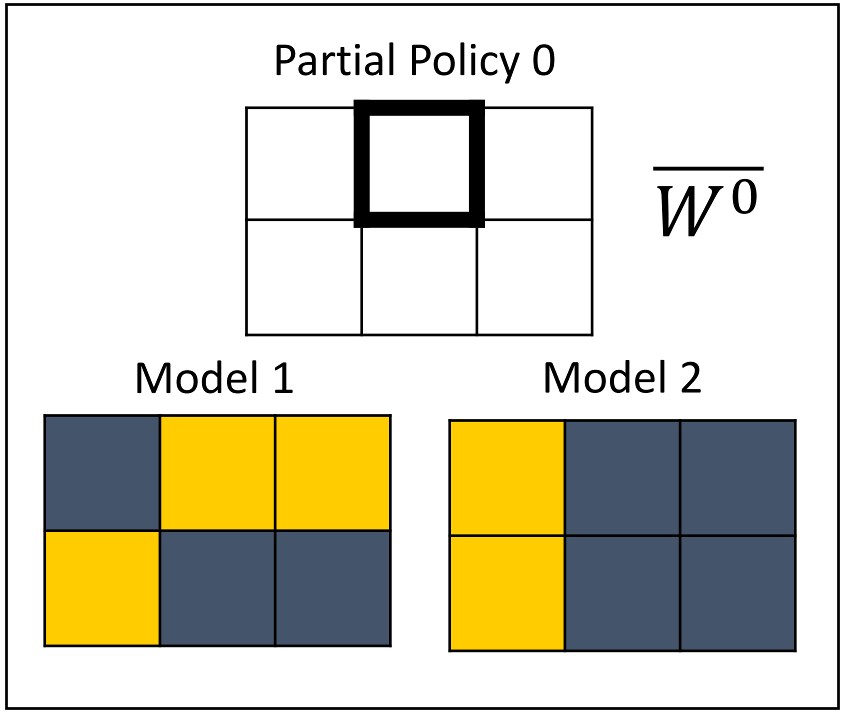


No actions have been fixed at the **root node**

Each model solved independently via backward induction

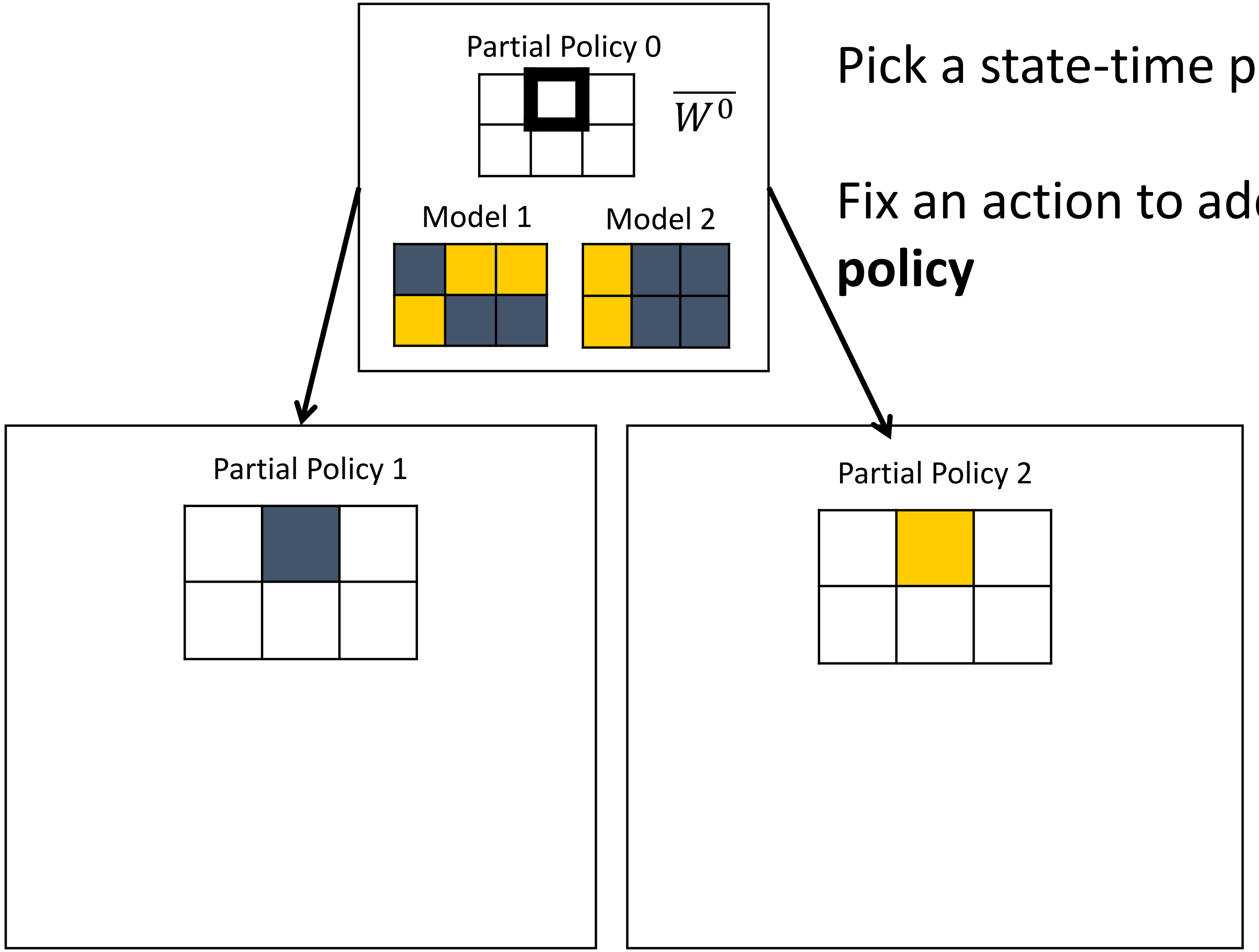
Gives an upper bound $\overline{W^0}$

Branch & Bound proceeds by fixing a part of the policy that must match in all models



Pick a state-time pair to branch on

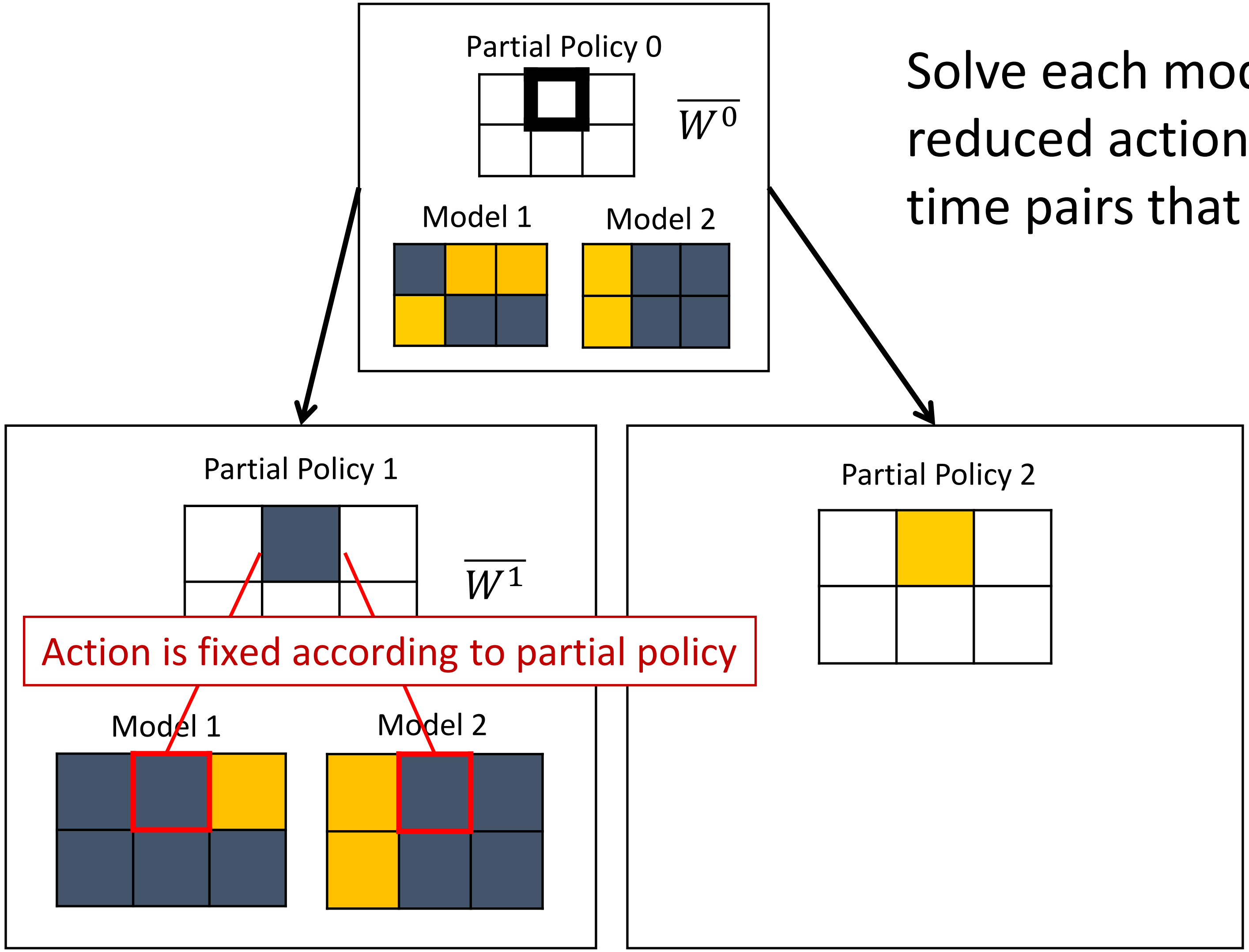
Branch & Bound proceeds by fixing a part of the policy that must match in all models



Pick a state-time pair to branch on

Fix an action to add to the **partial policy**

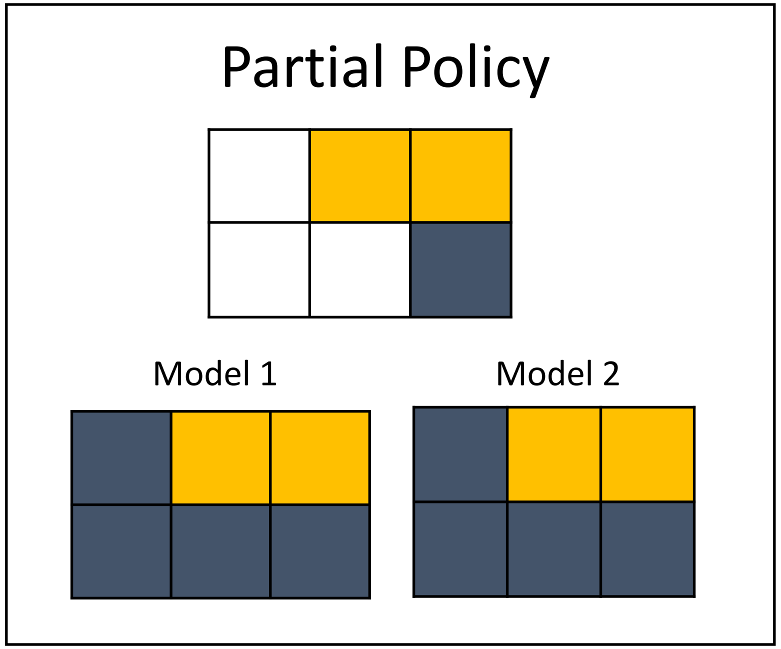
Branch & Bound solves a relaxation using backward induction to obtain upper bound



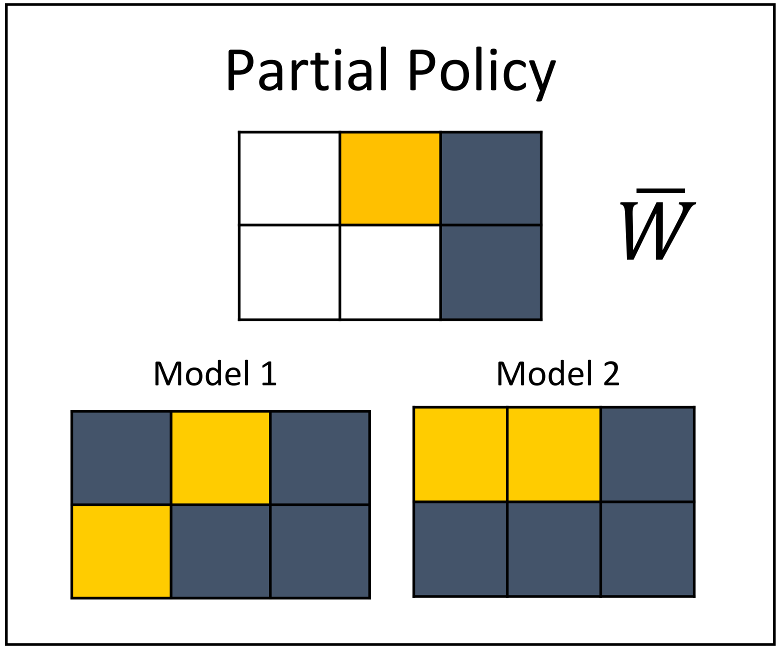
Solve each model's MDP with reduced action space for state-time pairs that are fixed

Pruning eliminates the need to explore all possible policies

<u>Prune by optimality</u>
Solving the relaxation gives an *implementable policy*



<u>Prune by bound</u>
The incumbent is better than any possible completion of the partial policy



We compared 3 exact methods on 240 instances of MMDPs

Solution Method	Implementation	% solved in 5 minutes?	Optimality Gap (avg.)
MIP Extensive Form	Gurobi		
MIP Branch-and-cut	Gurobi with Callbacks		
Branch-and-Bound	Custom code in C++		

[1] Steimle, L. N., Ahluwalia, V., Kamdar, C., and Denton B.T. (2018) “Decomposition methods for solving Multi-model Markov decision processes.” *IISE Transactions, 2022.*
[2] Gurobi Optimization, LLC (2018) “Gurobi Optimizer Reference Manual", http://www.gurobi.com

Our custom branch-and-bound approach is the fastest of the solution methods

Solution Method	Implementation	% solved in 5 minutes?	Optimality Gap (avg.)
MIP Extensive Form	Gurobi	0%	12.2%
MIP Branch-and-cut	Gurobi with Callbacks	0%	13.1%
Branch-and-Bound	Custom code in C++	**97.9%**	**1.11%**

Observations

- A custom branch-and-bound approach outperforms MIP-based solution methods
- MMDPs tend to be harder to solve when there is more variance in the models' parameters
- In many low variance cases, the mean value problem provides an optimal or near-optimal solution

The remainder of this presentation

Multi-model Markov decision processes

Branch-and-bound algorithms

Alternative ambiguity-aware formulations

So far, we have considered a risk-neutral decision-maker

Weighted value problem maximizes <u>expectation</u> of model performance

$$W^* = \max_{\pi \in \Pi} \sum_{m \in \mathcal{M}} \lambda_m v^m(\pi)$$

What if the decision-maker wants to protect against undesirable outcomes resulting from ambiguity?

Branch-and-bound algorithm is easily modified to solve other ambiguity-aware formulations

Max-min

$$\max_{\pi \in \Pi^{MD}} \min_{m \in \mathcal{M}} v^m(\pi)$$

Min-max-regret[1]

$$\min_{\pi \in \Pi^{MD}} \max_{m \in \mathcal{M}} \left\{ \max_{\bar{\pi} \in \Pi} v^m(\bar{\pi}) - v^m(\pi) \right\}$$

Percentile optimization[2]

$$\begin{aligned} \max_{z \in \mathbb{R}, \pi \in \Pi^{MD}} \quad & z \\ \text{s.t.} \quad & \mathbb{P}(v^m(\pi) \geq z) \geq 1 - \epsilon \end{aligned}$$

[1] Ahmed A, Varakantham P, Lowalekar M, Adulyasak Y, Jaillet P (2017) Sampling Based Approaches for Minimizing Regret in Uncertain Markov Decision Processes (MDPs). *Journal of Artificial Intelligence Research* 59:229–264

[2] Merakli, M. and Kucukyavuz, S. (2019) "Risk-Averse Markov Decision Processes under Parameter Uncertainty with an Application to Slow-Onset Disaster Relief." *Optimization Online*.

These problems are still NP-hard. We compared to polynomial-time alternatives

Mean Value Problem

$$\max_{\pi \in \Pi^{MD}} \left\{ \mathbb{E}^{\pi, \bar{P}} \left[\sum_{t=1}^{T} r_t(s,a) + r_{T+1}(s) \right] \right\}$$

(s,a)-rectangular finite scenario MDP*

$$\max_{a \in \mathcal{A}} \min_{p_t(s,a) \in \mathcal{P}_t(s,a)} \left\{ r_t(s,a) + \sum_{s' \in \mathcal{S}} p_t(s'|s,a) v_{t+1}(s) \right\}$$

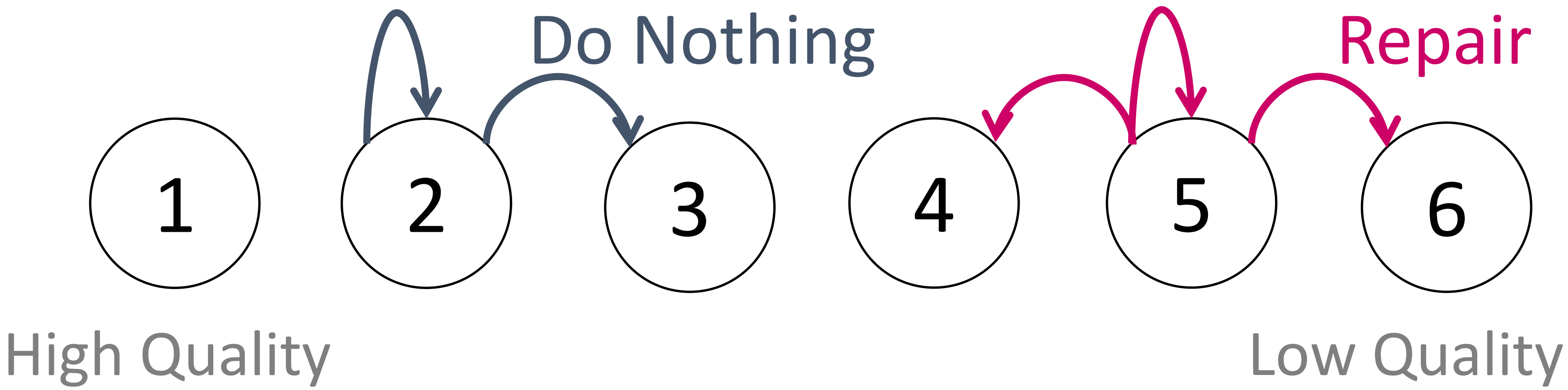
Nilim, Arnab, and Laurent El Ghaoui. "Robust control of Markov decision processes with uncertain transition matrices." *Operations Research* 53.5 (2005): 780-798.

We compared these formulations in two case studies

Machine maintenance

Cardiovascular disease management

Machine maintenance: Optimal timing of machine repairs



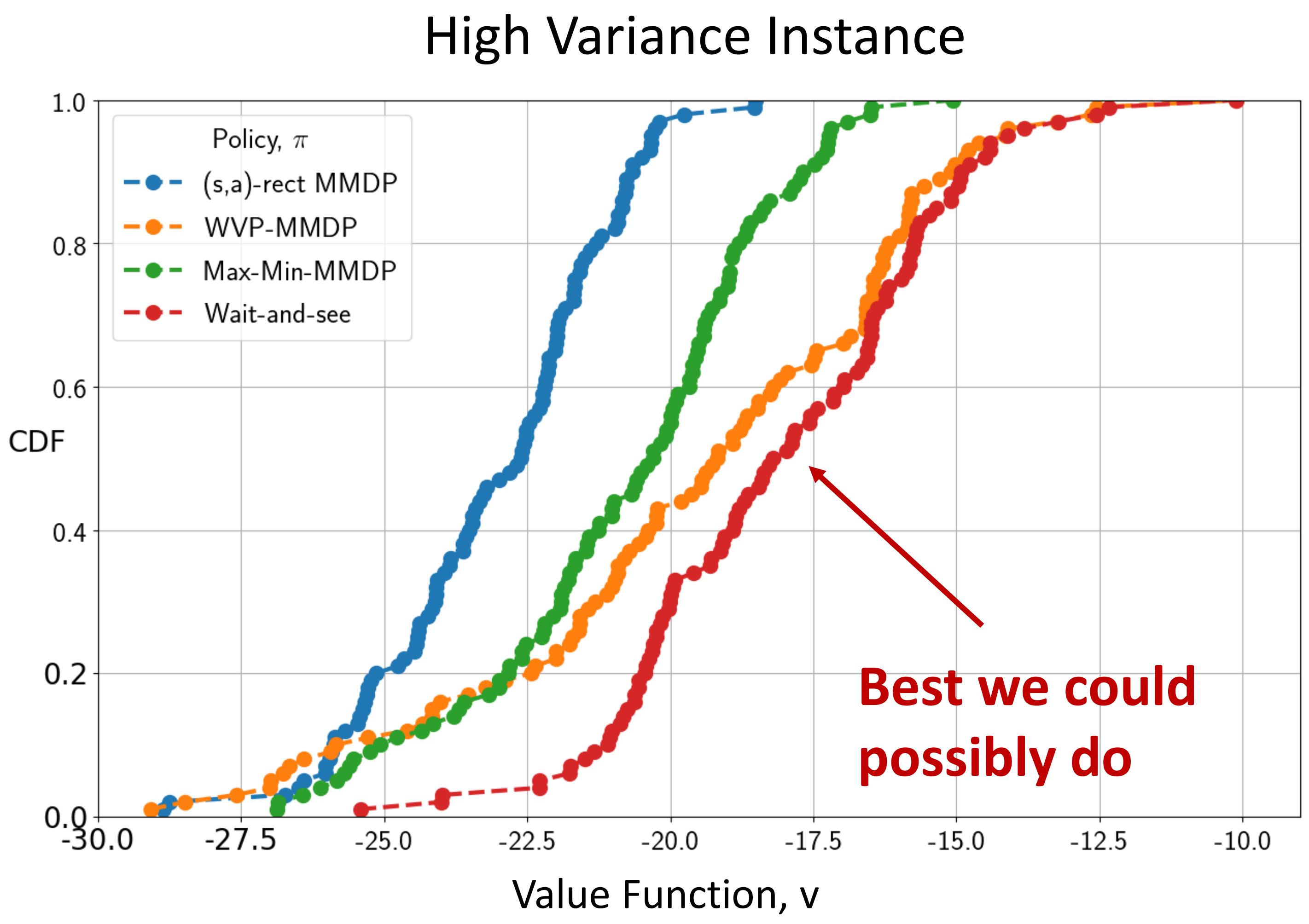
Operating costs depend on state of machine

Options:

- Do Nothing at no cost
- Minor repair at low cost
- Major repair at high cost

The distribution of the value function across models varies depending on the criteria selected

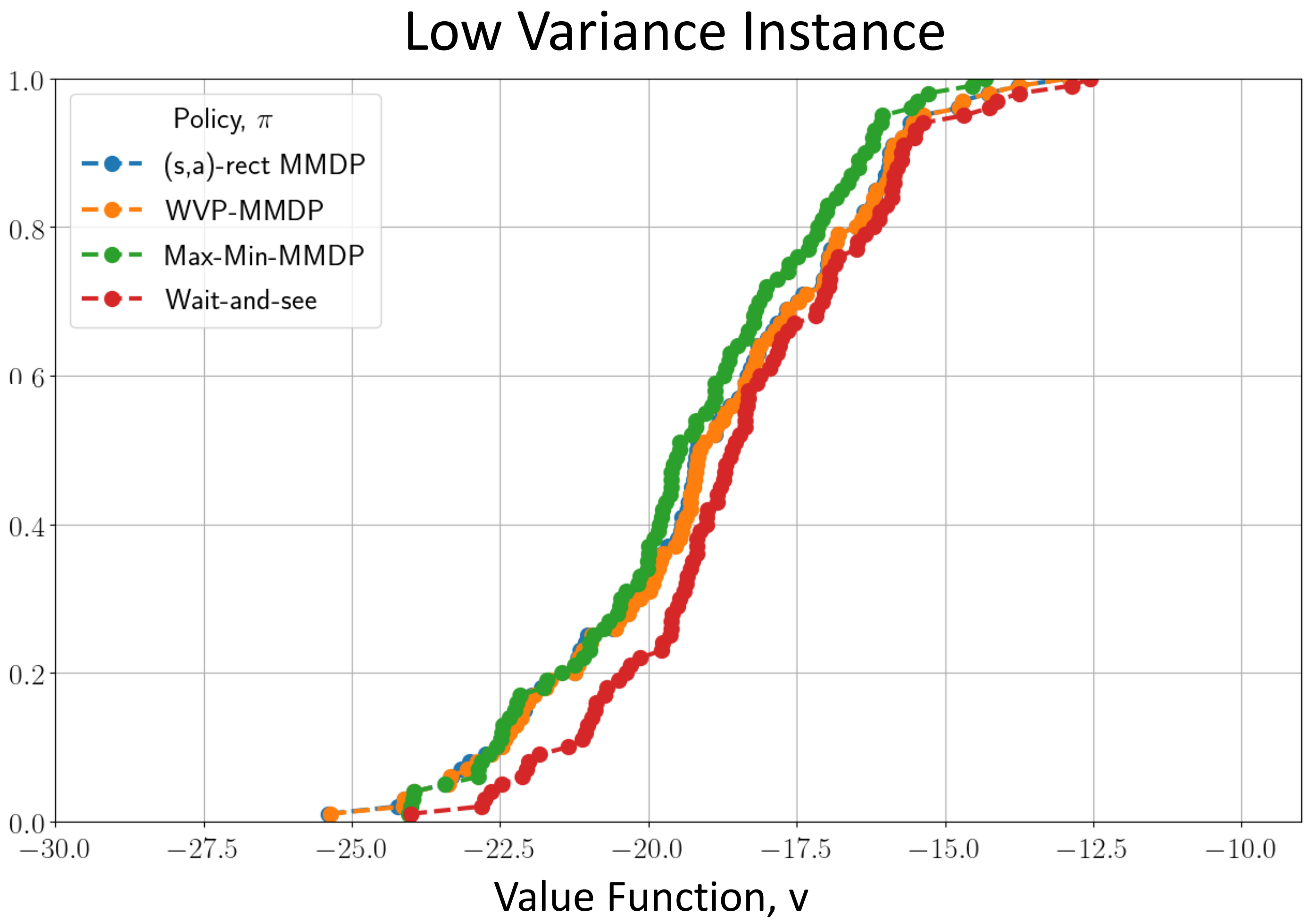
$\mathbb{P}(v^m(\pi) \leq v)$



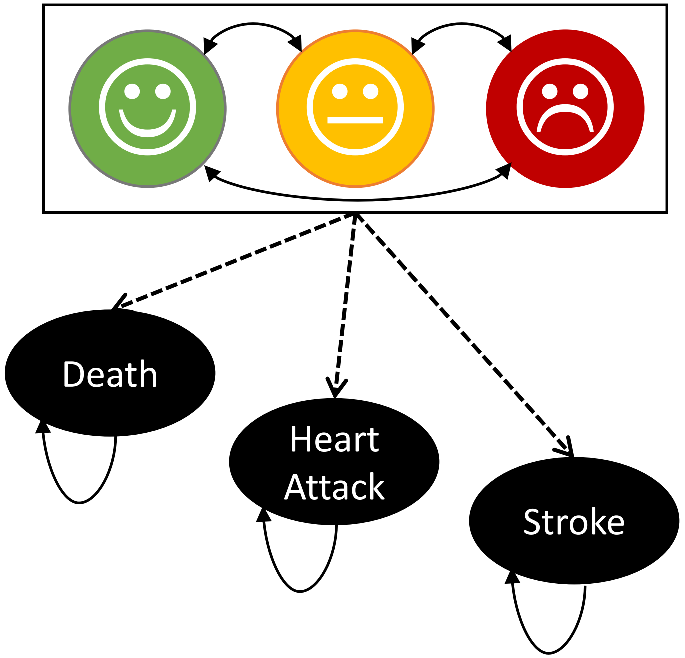
Higher cost → Lower cost

As variance in models decreases, the form of protection against ambiguity matters less

$\mathbb{P}(v^m(\pi) \leq v)$



We considered these formulations to determine the optimal time to start statins



Multi-model Markov decision process

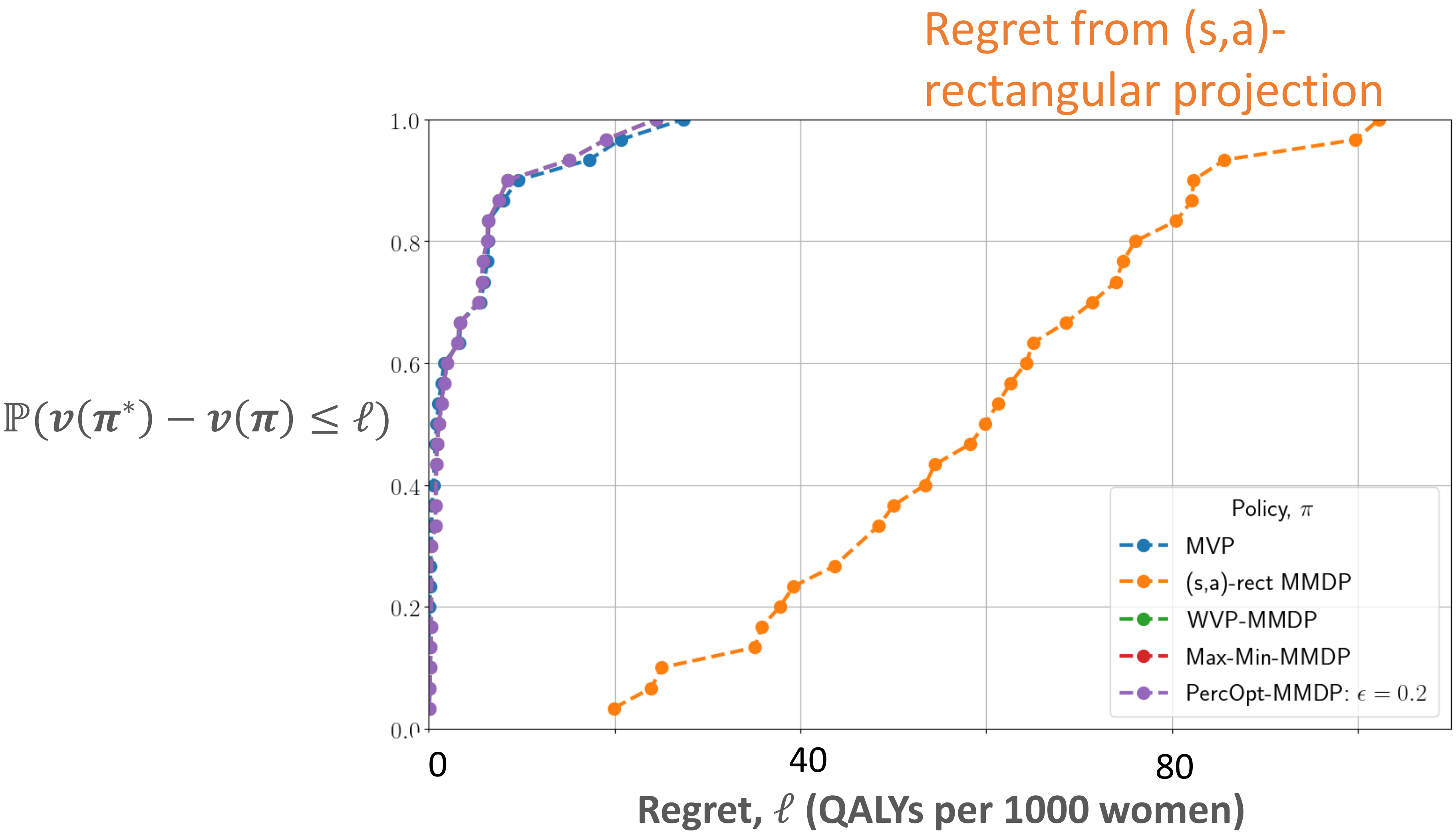
- 64 states (HDL/TC Levels)
- 3 actions (Wait, low-dose, high-dose)
- 34 decision epochs
- 30 models

Case study data

- Longitudinal data from Mayo Clinic
- ACC risk calculator
- Disutilities from medical literature

Mason, J. E., Denton, B. T., Shah, N. D., & Smith, S. A. (2014). Optimizing the simultaneous management of blood pressure and cholesterol for type 2 diabetes patients. *European Journal of Operational Research*, *233*(3), 727-738.

(s,a)-rect-MMDP can perform worse than all models



Take-away messages

- Use caution before employing the (s,a)-rectangularity property!
- MMDPs can generate superior performance in terms of expected rewards, regret, and other performance measures.
- Branch-and-bound can be customized to leverage MMDP structure and solve moderate-size problems. A fast polynomial time algorithm can scale up to very large problems.
- MMDPs are useful when there is significant variation among models. On the other hand, using an MMDP to address natural statistical variation in model parameters yields little benefit.

Acknowledgments

Michigan Engineering

Lauren, Steimle, Ph.D.

Vinayak Ahluwalia

Charmee Kamdar

Mayo Clinic

Nilay Shah, Ph.D.

UM-Dearborn School of Business

David Kaufman, Ph.D.

U.S. Department of Veterans Affairs

Rodney Hayward, MD

Jeremy Sussman, MD

This material is based upon work supported by the National Science Foundation under Grant Number CMMI- 1462060 (Denton). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Supplemental Material

Weight-Select-Update is an approximation algorithm to find a policy and model value functions

- A Markov deterministic policy
  - $\pi = \{\pi_t(s) : t \in \mathcal{T}, s \in \mathcal{S}\}$

- Value functions for each model corresponding to the policy
  - $v_{\mathrm{t}}^{m}(s), t \in \mathcal{T}, s \in \mathcal{S}, m \in \mathcal{M}$

Weight-Select-Update generates the policy and model value functions in one backward pass

Initialize value-to-go in each model:

$$\hat{v}_{T+1}^{m}(s) = r_{T+1}(s)$$

Weight-Select-Update generates the policy and model value functions in one backward pass

Initialize value-to-go in each model:

$$\hat{v}^m_{T+1}(s) = r_{T+1}(s)$$

While $t \geq 1$, for each state $s \in \mathcal{S}$:

- Conditioning on being in state s, select best action

$$\hat{\pi}_t(s) = \operatorname*{argmax}_{a \in \mathcal{A}} \underbrace{\left\{ \sum_{m=1}^{M} \lambda_m \left[r_t(s,a) + \sum_{s' \in \mathcal{S}} p^m(s'|s,a) \hat{v}^m_{t+1}(s) \right] \right\}}_{\text{Weighted value-to-go from state } s}$$

Weighted value-to-go from state *s*

Weight-Select-Update generates the policy and model value functions in one backward pass

Initialize value-to-go in each model:

$$\hat{v}^m_{T+1}(s) = r_{T+1}(s)$$

While $t \geq 1$, for each state $s \in \mathcal{S}$:

- Conditioning on being in state s, select best action

$$\hat{\pi}_t(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \underbrace{\left\{ \sum_{m=1}^{M} \lambda_m \left[r_t(s,a) + \sum_{s' \in \mathcal{S}} p^m(s'|s,a) \hat{v}^m_{t+1}(s) \right] \right\}}_{\text{Weighted value-to-go from state } s}$$

Weighted value-to-go from state *s*

Weight-Select-Update generates the policy and model value functions in one backward pass

Initialize value-to-go in each model:

$$\hat{v}^m_{T+1}(s) = r_{T+1}(s)$$

While $t \geq 1$, for each state $s \in \mathcal{S}$:

- Conditioning on being in state s, select best action

$$\hat{\pi}_t(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m \left[r_t(s,a) + \sum_{s' \in \mathcal{S}} p^m(s'|s,a) \hat{v}^m_{t+1}(s) \right] \right\}$$

- Update value-to-go in each model for policy

$$\hat{v}^m_t(s) = r_t\big(s, \hat{\pi}_t(s)\big) + \sum_{s' \in \mathcal{S}} p^m(s'|s, \hat{\pi}_t(s)) \hat{v}^m_{t+1}(s)$$

We can bound the error on the policy found via *Weight-Select-Update*

Bound on optimality gap is based on *wait-and-see*

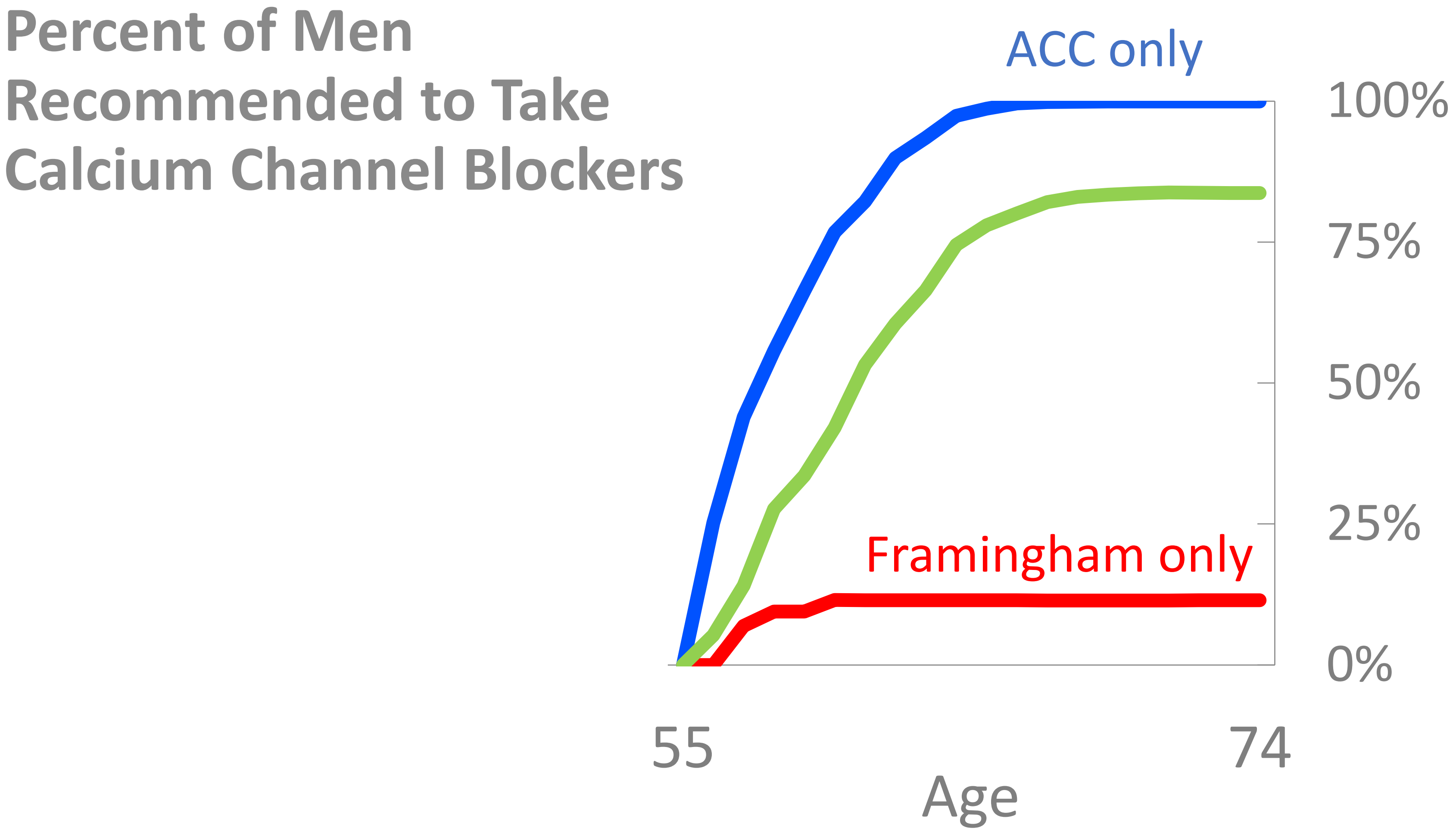
$$\sum_{m \in \mathcal{M}} \lambda_m v_m(\hat{}) \leq \max_{\pi \in \Pi^{MD}} \sum_{m \in \mathcal{M}} \lambda_m v_m(\pi)$$
$$\leq \sum_{m \in \mathcal{M}} \lambda_m \left[\max_{\pi \in \Pi^{MD}} v_m(\pi) \right]$$

Performance guarantee for 2 model MMDPs:

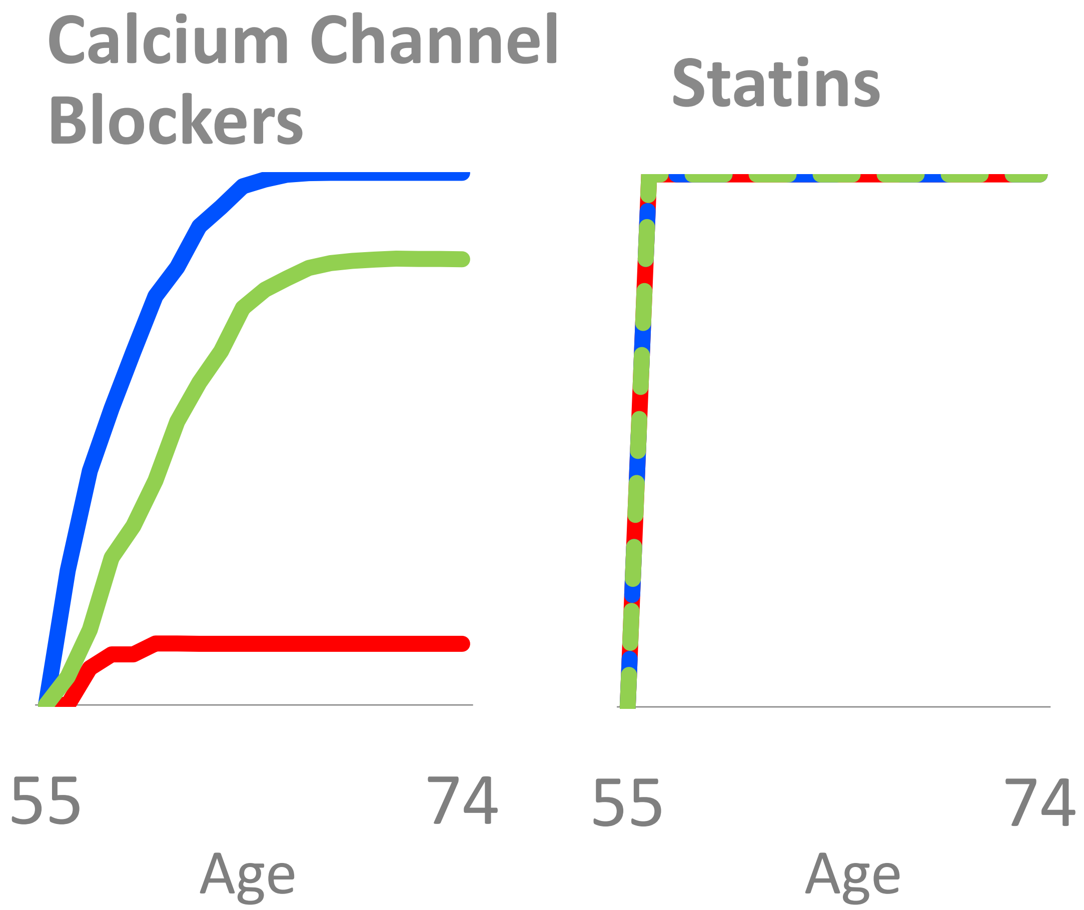
Better than choosing “wrong” model

$$\lambda_1 v^1(\pi^{*,2}) + \lambda_2 v^2(\pi^{*,1}) \leq \lambda_1 v^1(\hat{\pi}) + \lambda_2 v^2(\hat{\pi})$$

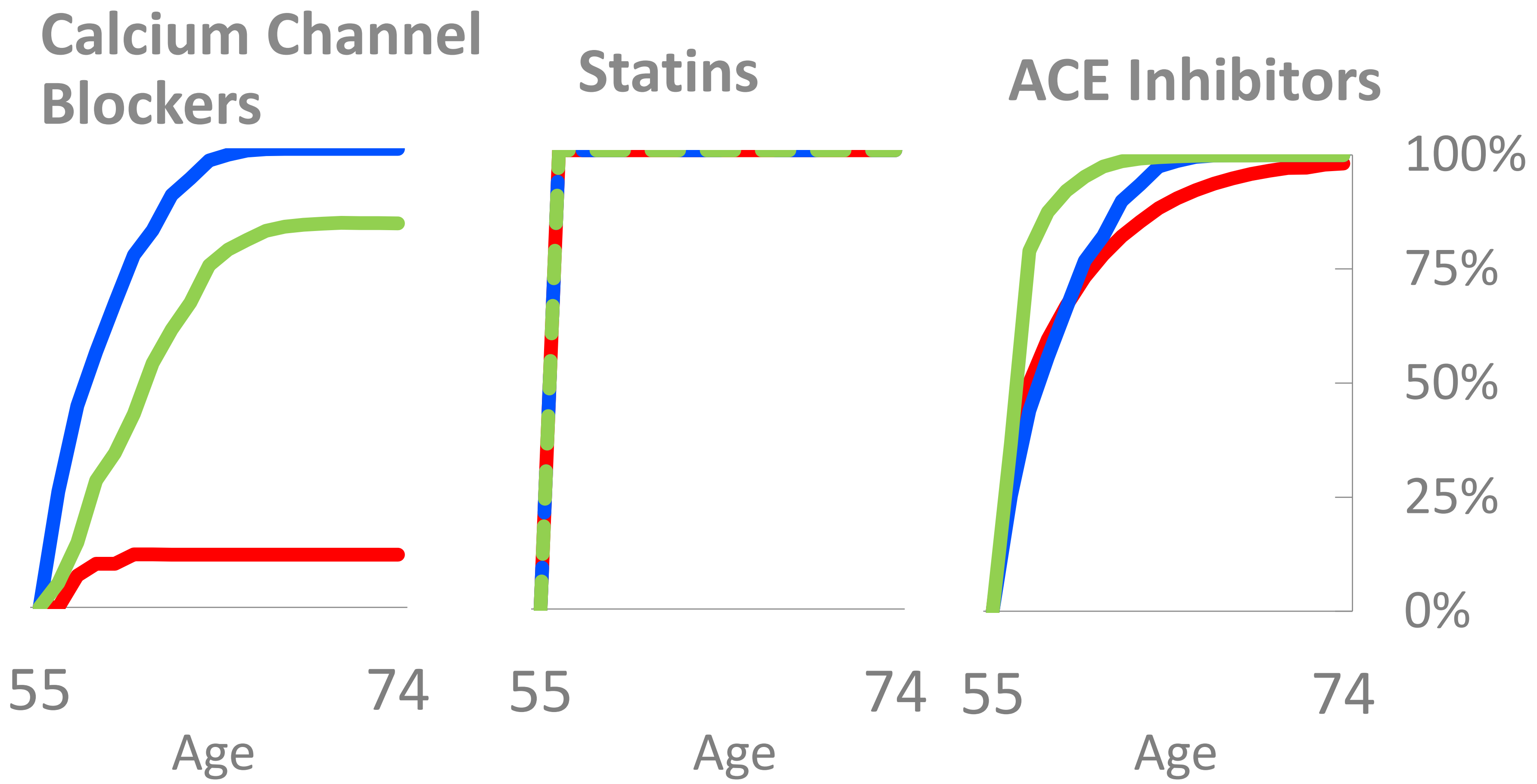
Our algorithm provides recommendations that work well in both models



Our algorithm provides recommendations that work well in both models



Our algorithm provides recommendations that work well in both models



Proposition: Solving the non-adaptive problem for an MMDP is NP-hard.

Proof Sketch: Reduction from 3-CNF-SAT which is NP-hard.

Problem instance:

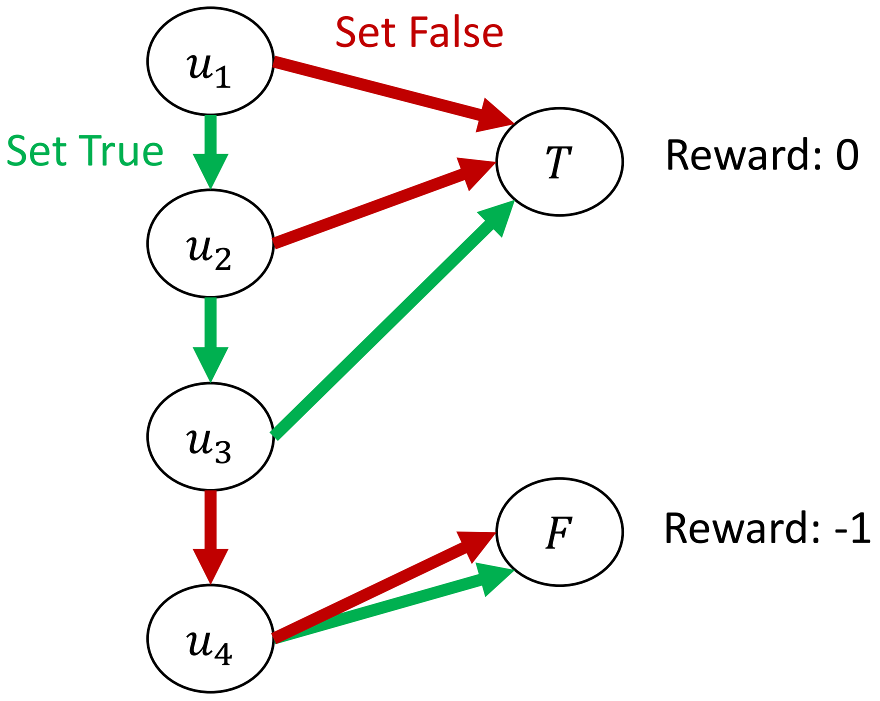
- a set of variables U = $\{u_1, u_2,…, u_n\}$
- a formula E = $C_1 \land\ C_2 \land \cdots \land C_m$ where each C_1 is CNF with 3 literals per clause

Question: Is there a truth assignment such that E is true?

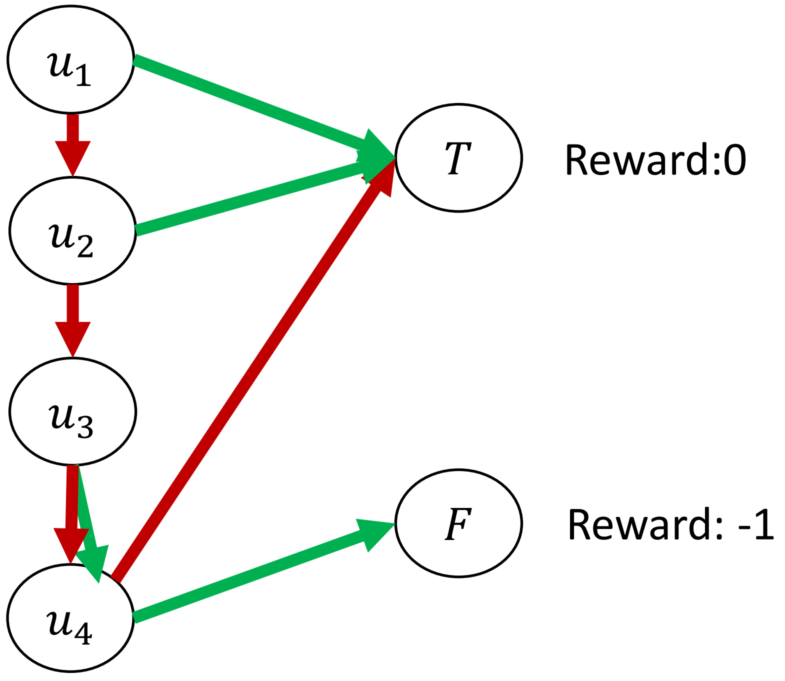
Proposition: Solving the non-adaptive problem for an MMDP is NP-hard.

Example: $E = (!u_1 \vee !u_2 \vee u_3) \wedge (u_1 \vee u_2 \vee u_4)$

E is true IFF there exists a Markov deterministic policy that achieves a weighted value > 0 in the MMDP



Model 1



Model 2

Ranges for TC, HDL, SBP states

	L	M	H	V
TC (mg/dL)	<160	[160,200)	[200, 240)	≥240
HDL (mg/dL)	<40	[40,50)	[50, 60)	≥60
SBP (mmHg)	<120	[120,140)	[140, 160)	≥160

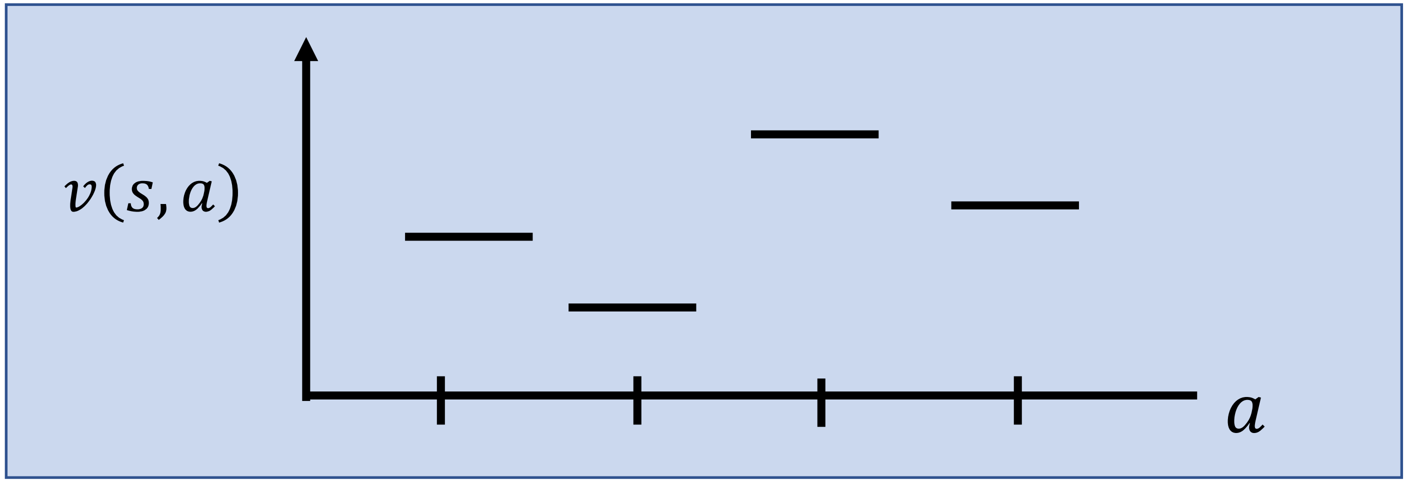
Mason, J. E., Denton, B. T., Shah, N. D., & Smith, S. A. (2014). Optimizing the simultaneous management of blood pressure and cholesterol for type 2 diabetes patients. *European Journal of Operational Research*, *233*(3), 727-738.

Linear programming can also be used to solve Markov decision processes

$v(s)$ = value-to-go from state s

$$\begin{aligned} \max_v \ & \sum_{s \in \mathcal{S}} \mu(s) v(s) \\ \text{s.t.} \quad & v(s) = \max_a \{ v(s,a) \}, \forall s \in \mathcal{S} \end{aligned}$$

$$\begin{aligned} \min_v \ & \sum_{s \in \mathcal{S}} \mu(s) v(s) \\ \text{s.t.} \quad & v(s) \geq v(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A} \end{aligned}$$



The MMDP can be solved by a MIP with Big-Ms to enforce logic constraints

Model-specific continuous value function decision variables

$v_t^m(s)$ = value to go from state *s* in epoch *t* in model *m*

Introduce binary decision variables to represent policy

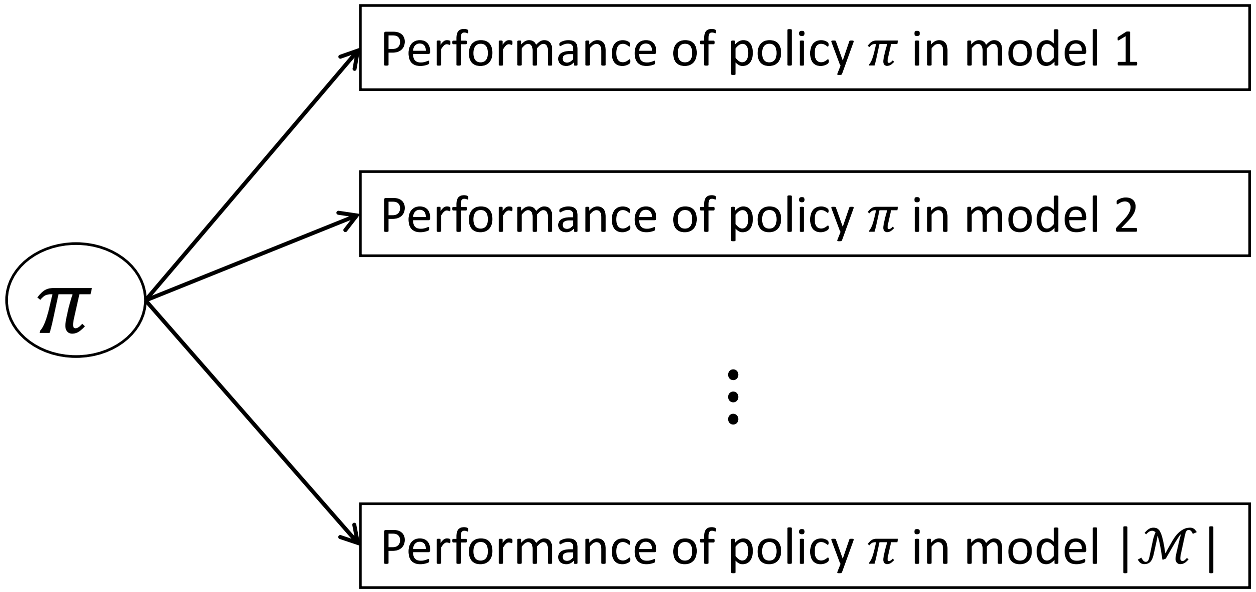
$$\pi_t(a|s) = \begin{cases} 1 & \text{if policy take action } a \text{ in state } s \text{ at epoch } t \\ 0 & \text{otherwise} \end{cases}$$

Constraints enforce value function estimates correspond to policy

$$M\pi_t(a|s) + v_t^m(s) - \sum_{s' \in \mathcal{S}} p_t^m(s'|s,a) v_{t+1}^m(s') \leq r_t(s,a) + M,$$

$$\forall s \in \mathcal{S}, a \in \mathcal{A}, t \in \mathcal{T}, m \in \mathcal{M}$$

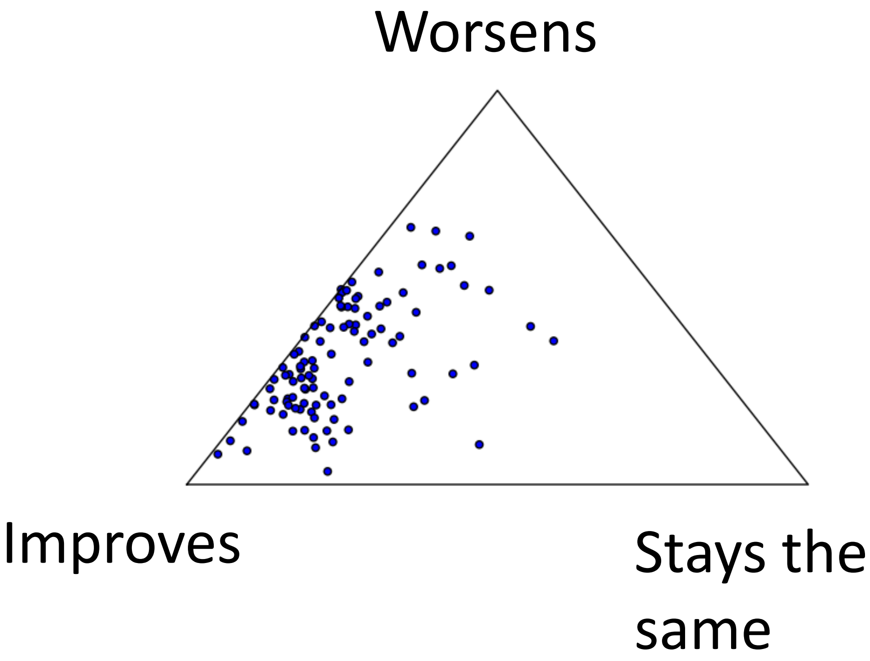
Connections to stochastic programming give insight into exact solution methods



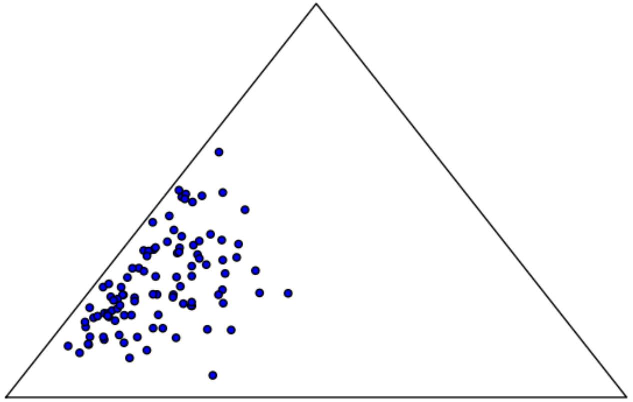
Stochastic program	MMDP
Scenarios	Model of MDP
Binary first-stage decision variables	Policy
Continuous second-stage decision variables	MDP model value functions

We used the Dirichlet distribution to control the variance among 100 models

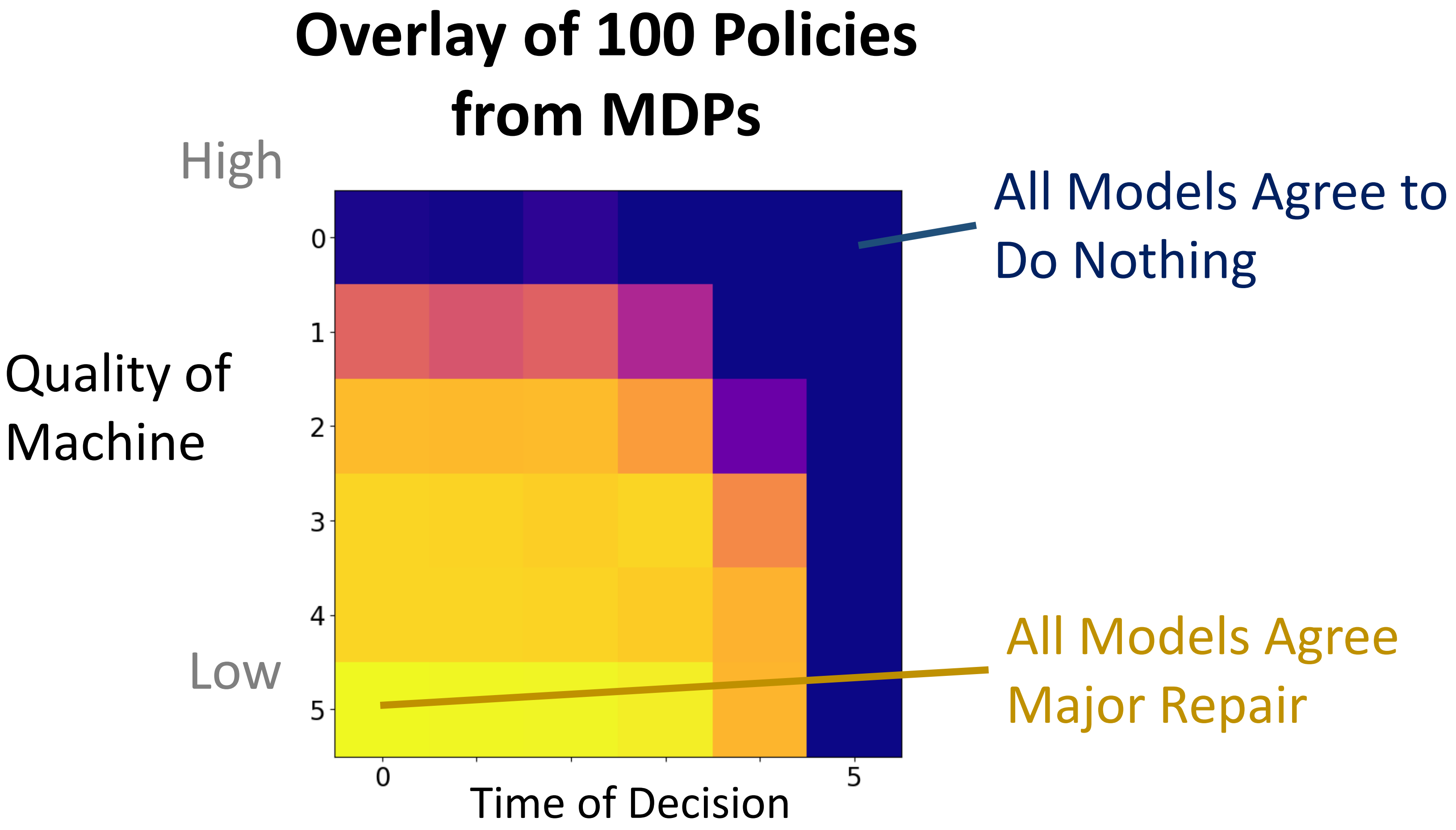
Higher variance



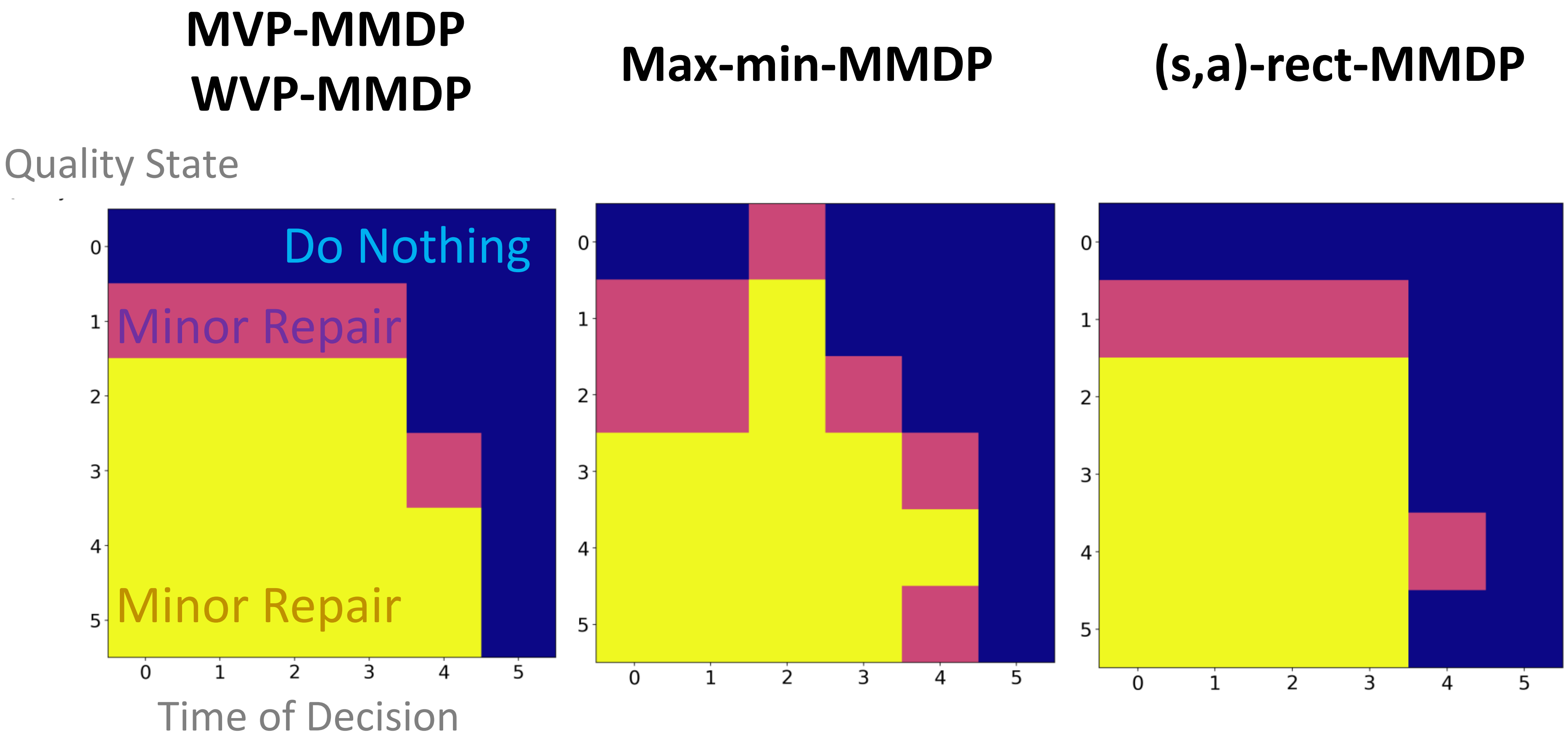
Lower variance



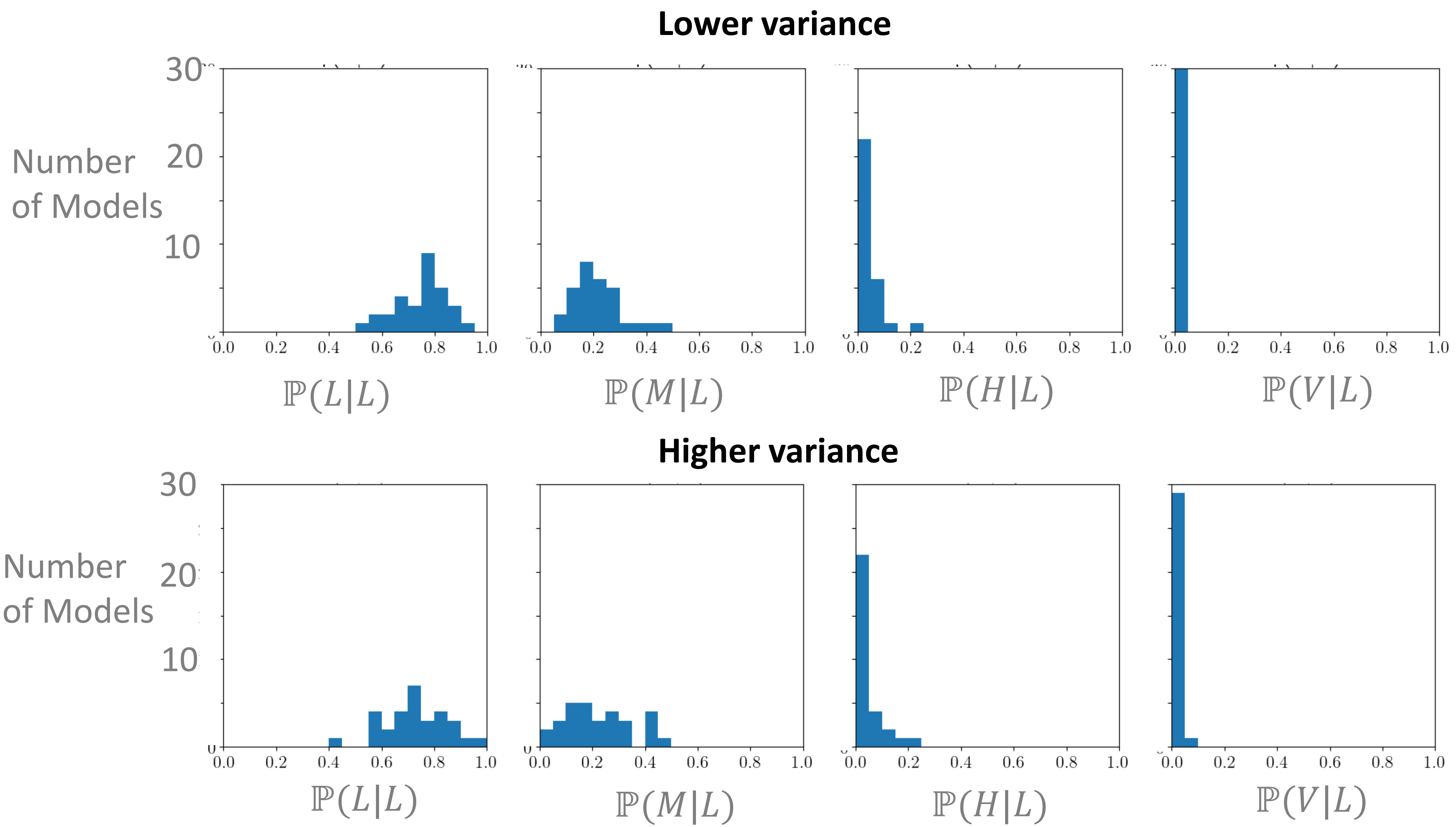
Different model suggest different maintenance recommendations



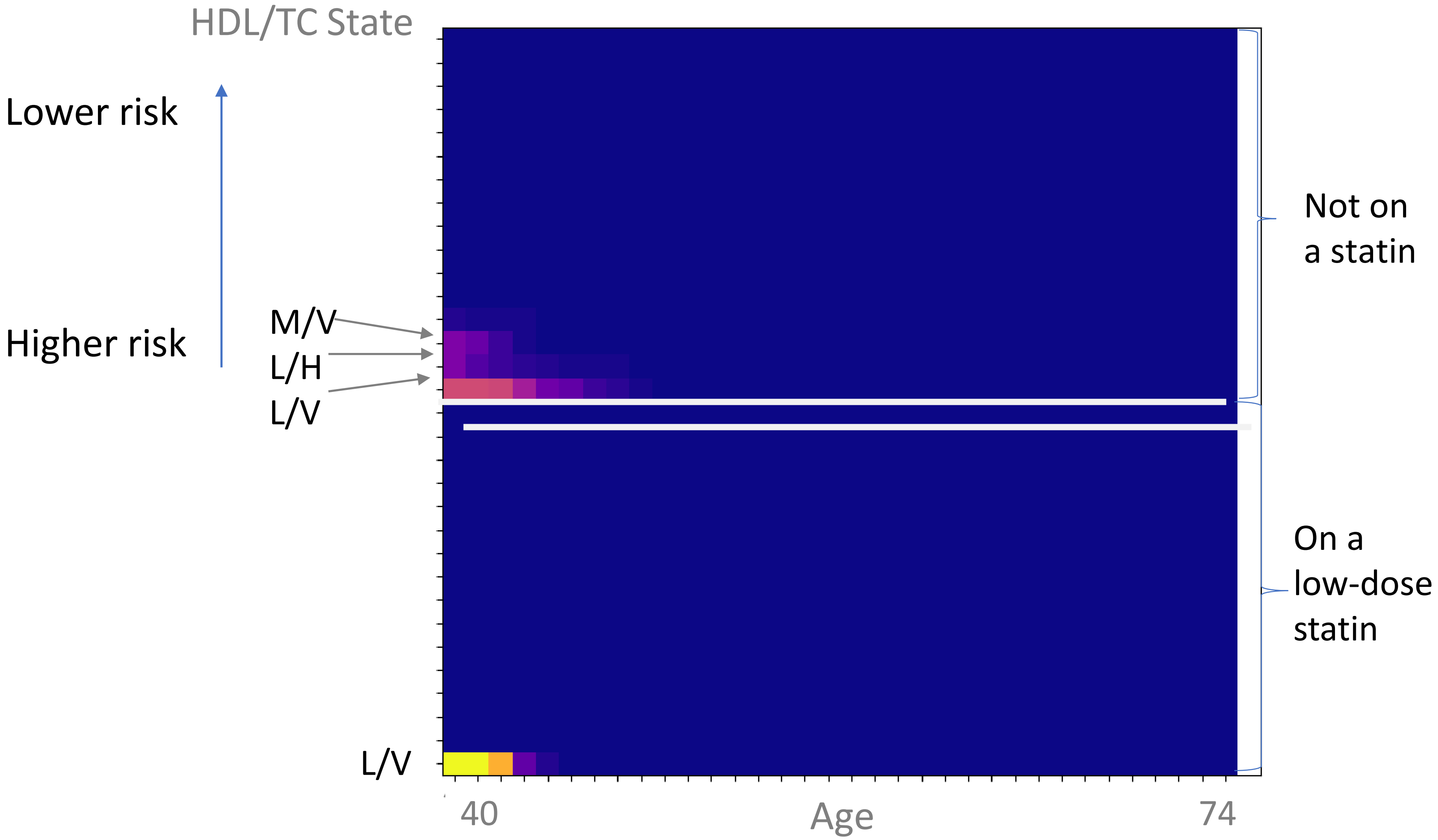
Alternate measures of protection against ambiguity may offer different policies



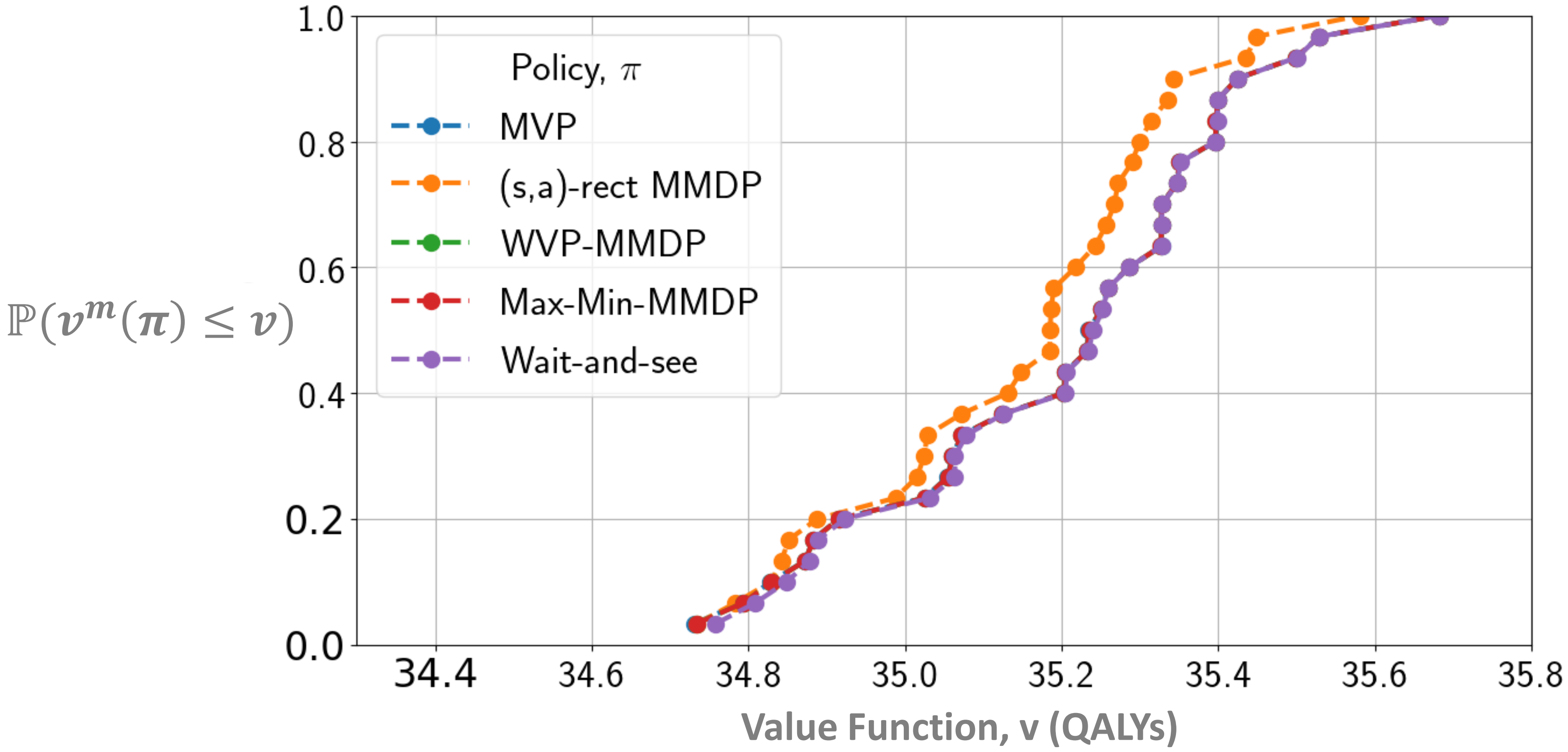
We used the Dirichlet distribution to control the variance among 30 models



Recommendations can be sensitive to which model is used



In some cases, MVP performs well on many metrics



Stochastic dynamic optimization under ambiguity

Multi-model Markov decision processes

Decomposition methods

Other ambiguity-aware formulations

There are interesting opportunities to extend this work

Infinite-horizon Markov decision processes with ambiguity

Extension: Modify relaxation in B&B

Existence of sufficient conditions for monotone policies

Extension: Sufficient conditions for monotone policy that is optimal for the MMDP

Ambiguous state-space definitions

Extension: Branching on mappings of actions in B&B