Probabilistic Guarantees for Autonomous Systems

Sam Coogan
Assistant professor
Electrical and Computer Engineering
Civil and Environmental Engineering

December 6, 2019

Georgia Tech

Formal methods and Autonomous Control of Transportation Systems (FACTS) Lab

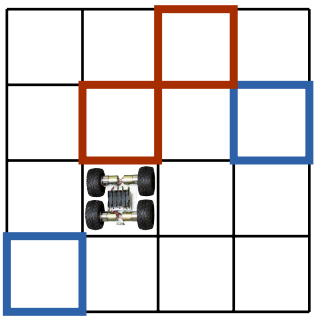
VPlayer.swf

VPlayer.swf

VPlayer.swf

VPlayer.swf

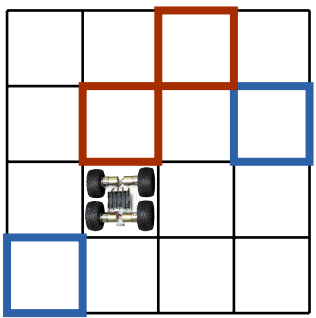
Modeling a stochastic system: Classic grid world

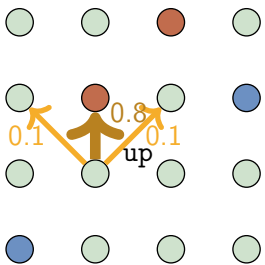


- Robot actions: $\{\texttt{up}, \texttt{down}, \texttt{left}, \texttt{right}\}$
- Actions have probabilistic consequences, *e.g.*:

Example control objective: Visit each blue square infinitely often, never visit a red square.

Modeling the system as a Markov Decision Process (MDP)

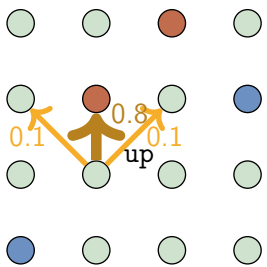




Modeling the system as a Markov Decision Process (MDP)

MDP $\mathcal{M} = (Q, Act, T, \Sigma, L)$:

- Q is set of states, *e.g.*,
  $Q = \{(1,1), (1,2), \ldots, (4,4)\}$
- Act is set of actions, *e.g.*,
  $Act = \{\texttt{up}, \texttt{down}, \texttt{left}, \texttt{right}\}$,
- $T : Q \times Act \times Q \to [0,1]$ is transition matrix, *e.g.*, $T\big((2,2), \text{up}, (3,3)\big) = 0.1$
- Σ is set of labels, *e.g.*,
  $\Sigma = \{\text{red}, \text{blue}, \text{none}\}$
- $L : Q \to \Sigma$ is labeling function, *e.g.*,
  $L\big((1,1)\big) = \text{blue}$



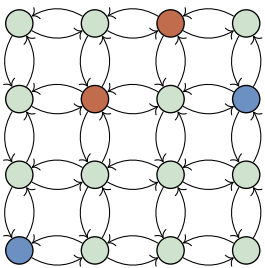
Synthesizing controllers for a canonical reach problem

Consider the goal:

Synthesize a control strategy that maximizes the probability of:
(or, given a control strategy, **verify** the probability of:)

- reaching a good (blue) state infinitely often, while
- encountering a bad (red) state finitely many times (eventually avoid)

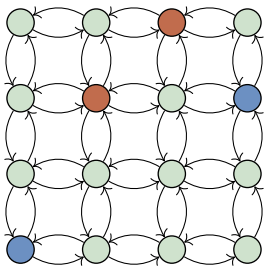
  - Fact 1: optimal controller u is memoryless, *i.e.*, $u : Q \to Act$
  - Fact 2: Maximizing probability solvable via graph-based search for "favorable" strongly connected components, then linear program (i.e., it's easy)



Complex specifications in LTL

Linear temporal logic (LTL) allows for complex specifications such as:

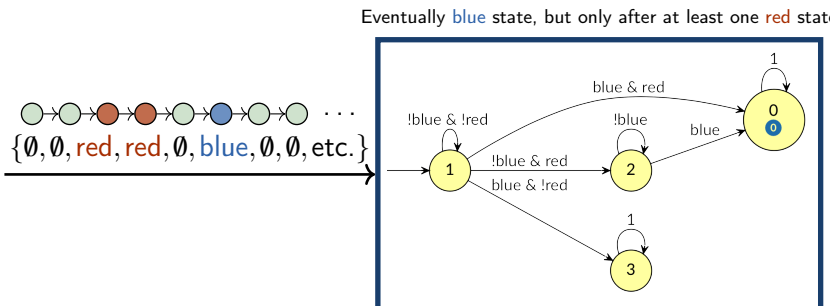
- Eventually visit a blue state, but only after visiting at least one red
- If a red state is visited, then visit a blue state within 3 time steps
- Never visit two blue states in a row



LTL synthesis/verification reduces to canonical reach problem on larger product MDP

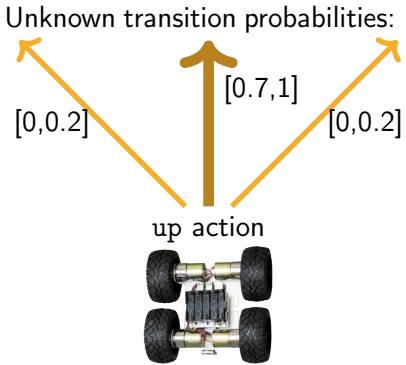
LTL Example

- LTL specs are written in a formal syntax (e.g., $\Box$ for "always", $\Diamond$ for "eventually")
- Can be converted into "memory-like" automaton with reach-avoid requirement




VPlayer.swf

Nondeterministic uncertainty and Bounded-parameter MDPs



Bounded-parameter MDP (BMDP)

$\mathcal{B} = (Q, Act, \check{T}, \hat{T}, \Sigma, L)$ same as MDP but with:

- $\check{T} : Q \times Act \times Q \to [0,1]$, lower transition bounds
- $\hat{T} : Q \times Act \times Q \to [0,1]$, upper transition bounds
- Must have $\check{T}(q,a,q') \leq \hat{T}(q,a,q')$ for all q, a, q' and

$$\sum_{q' \in Q} \check{T}(q,a,q') \leq 1 \leq \sum_{q' \in Q} \hat{T}(q,a,q')$$

In this talk

1. Dynamic systems with uncountable state-space, e.g., $\mathbb{R}^n$?
   Partition the space, approximate as BMDP abstraction
2. Solving simple reach-avoid problems in BMDPs?
   Much more complicated because intervals that include 0 affect qualitative graph structure
3. Solving against complex specifications in, *e.g.*, a temporal logic?
   Reduce to simple reachability problem, then refine state-space partition

Stochastic dynamical system

$$x[k+1] = F(x[k], w[k])$$

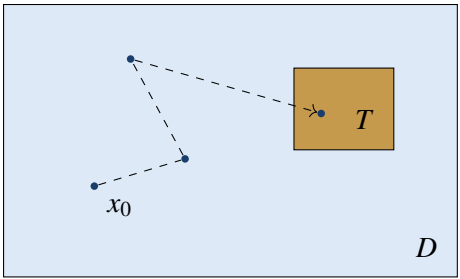
- $x[k] \in D \subset \mathbb{R}^n$ is state
- $w[k] \in W \subset \mathbb{R}^p$ is random disturbance with probability density function f_w

Note:

- Rest of talk: no control input (i.e., we are studying verification), but
- Statespace is $D \subset \mathbb{R}^n$ (Markov Chain with uncountably infinite states)

A reachability problem

A trajectory of $x[k+1] = F(x[k], w[k])$ initialized at x_0 is a random walk on D

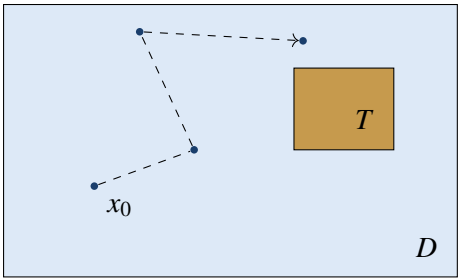


Given set $T \subset D$, and target probability threshold p_{thresh}, color:

- Green those states x_0 such that Prob($x[k]$ reaches $T|x_0) \geq p_{\text{thres}}$
- Red all other states

A reachability problem

A trajectory of $x[k+1] = F(x[k], w[k])$ initialized at x_0 is a random walk on D

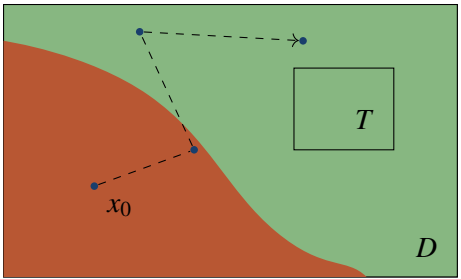


Given set $T \subset D$, and target probability threshold p_{thresh}, color:

- Green those states x_0 such that Prob($x[k]$ reaches $T | x_0$) $\geq p_{\text{thres}}$
- Red all other states

A reachability problem

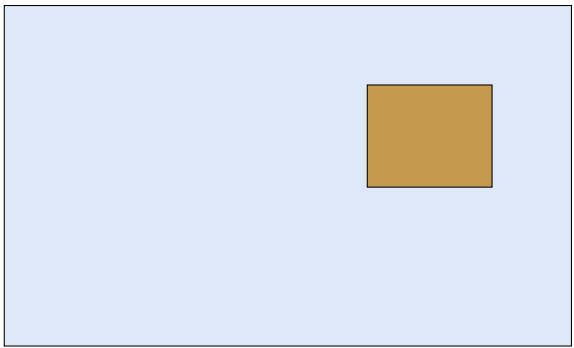
A trajectory of $x[k+1] = F(x[k], w[k])$ initialized at x_0 is a random walk on D



Given set $T \subset D$, and target probability threshold p_{thresh}, color:

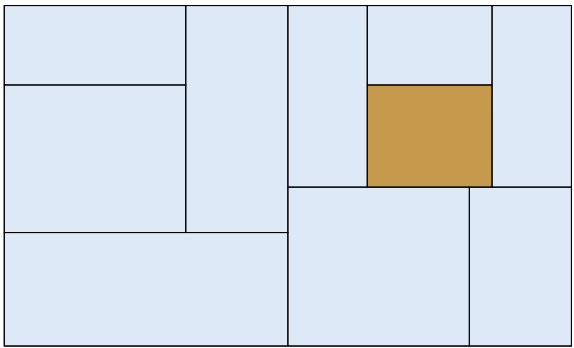
- Green those states x_0 such that Prob($x[k]$ reaches $T|x_0) \geq p_{\text{thres}}$
- Red all other states

Approximating a solution with a finite abstraction



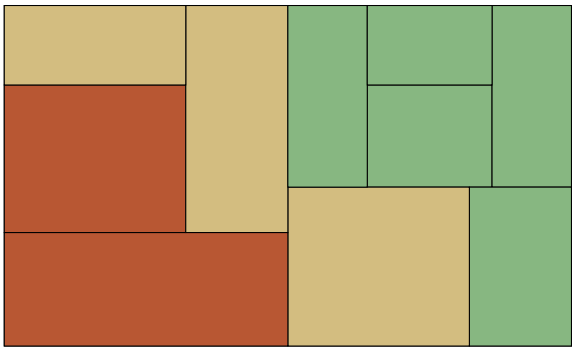
- Partition state space
- Color each partition:
  - Green,
  - Red, or
  - Yellow if inconclusive

Approximating a solution with a finite abstraction



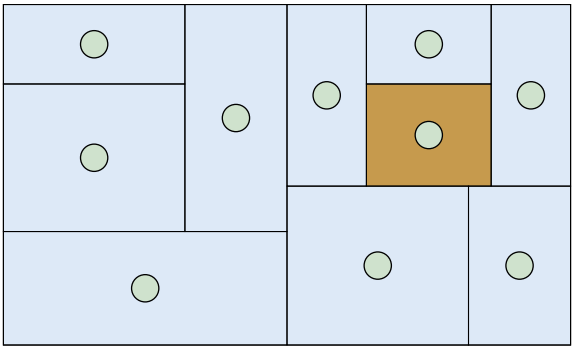
- Partition state space
- Color each partition:
  - Green,
  - Red, or
  - Yellow if inconclusive

Approximating a solution with a finite abstraction



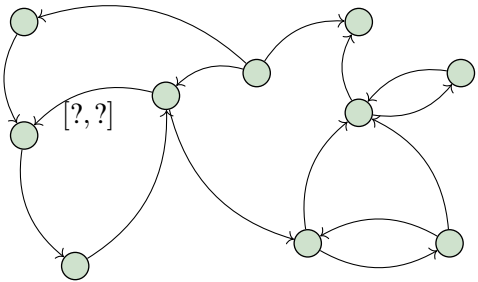
- Partition state space
- Color each partition:
  - Green,
  - Red, or
  - Yellow if inconclusive

A BMDP abstraction



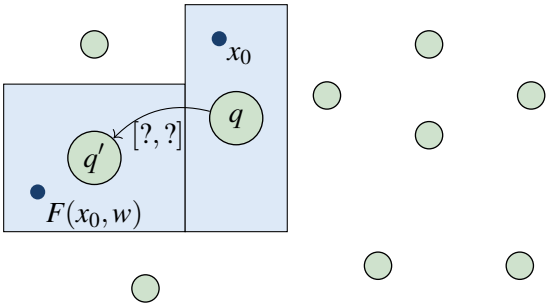
- Lower transition bound: $\min_{x_0 \in q} \text{Prob}(F(x_0, w) \in q' \mid x_0)$
- Upper transition bound: $\max_{x_0 \in q} \text{Prob}(F(x_0, w) \in q' \mid x_0)$

A BMDP abstraction



- Lower transition bound: $\min_{x_0 \in q} \text{Prob}(F(x_0, w) \in q' \mid x_0)$
- Upper transition bound: $\max_{x_0 \in q} \text{Prob}(F(x_0, w) \in q' \mid x_0)$

A BMDP abstraction



- Lower transition bound: $\min_{x_0 \in q} \text{Prob}(F(x_0, w) \in q' \mid x_0)$
- Upper transition bound: $\max_{x_0 \in q} \text{Prob}(F(x_0, w) \in q' \mid x_0)$

A BMDP abstraction

A partition Q with transition functions $\check{T} : Q \times Q \to [0,1]$ and $\hat{T} : Q \times Q \to [0,1]$ is a **BMDP abstraction** of $x[k+1] = F(x[k]) + w[k]$ if, for all $q, q' \in Q$,

$$\check{T}(q,q') \leq \min_{x_0 \in q} \text{Prob}(F(x_0,w) \in q' \mid x_0), \text{ and}$$

$$\hat{T}(q,q') \geq \max_{x_0 \in q} \text{Prob}(F(x_0,w) \in q' \mid x_0)$$

- A BMDP abstraction overapproximates the behavior of the true system
- Leads to sound (but possibly conservative) algorithms

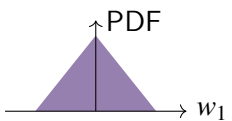
Mixed monotone stochastic systems

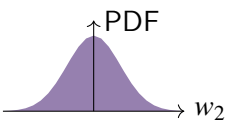
Specialize to

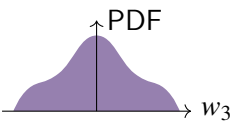
$$x[k+1] = F(x[k]) + w[k]$$

where:

1. $F : D \to D$ is *mixed monotone*
2. Each element of w is drawn from a symmetric, unimodal distribution







Mixed monotonicity [Coogan and Arcak, HSCC '15]

The system

$$x[k+1] = F(x[k])$$

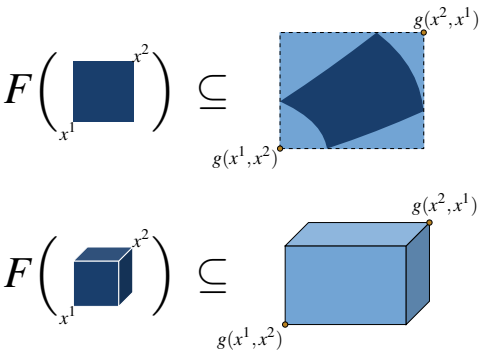
is **mixed monotone** if there exists $g : D \times D \to D$ such that:

1. for all x: $F(x) = g(x,x)$
2. for all $x^1, x^2, y : x^1 \leq x^2$ implies $g(x^1, y) \leq g(x^2, y)$
3. for all $x, y^1, y^2 : y^1 \leq y^2$ implies $g(x, y^2) \leq g(x, y^1)$

- Generalizes monotone systems, for which $g(x,y) = F(x)$

Reachable set approximation for mixed monotone systems: If $x^1 \leq x \leq x^2$, then

$$g(x^1, x^2) \leq F(x) \leq g(x^2, x^1)$$



Mixed monotonicity: A special case

$x^+ = F(x)$ is mixed monotone if

for all i,j, there exists $\delta_{ij} \in \{-1,1\}$ s.t. $\delta_{ij}\frac{\partial F_i}{\partial x_j}(x) \geq 0 \quad \forall x,$

i.e., Jacobian is **sign-stable**

Construct decomposition function:

$$g_i(x,y) = F_i(\ldots, \overset{y_j}{\not{x}_j}, \ldots) \qquad \text{if } \delta_{ij} = -1$$

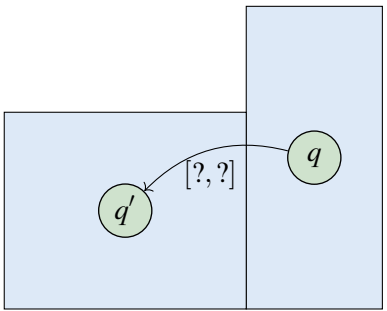
$$g(x,y) = (g_1(x,y), \ldots, g_n(x,y))$$

1. $g(x,x) = F(x)$
2. $\frac{\partial g}{\partial x}(x,y) \geq 0$
3. $\frac{\partial g}{\partial y}(x,y) \leq 0$

- Can generalize to systems with bounded Jacobian
- Mixed monotonicity is quite general, but its usefulness depends on conservatism of decomposition function

Computing transition probabilities

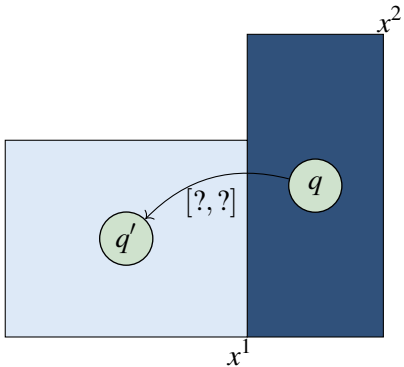
$$x[k+1] = F(x[k]) + w[k]$$



- Maximizing transition probability: Shift disturbance closest to center of q'
- Minimizing transition probability: Shift disturbance furthest from center of q'
- Integrate disturbance PDF over q'

Computing transition probabilities

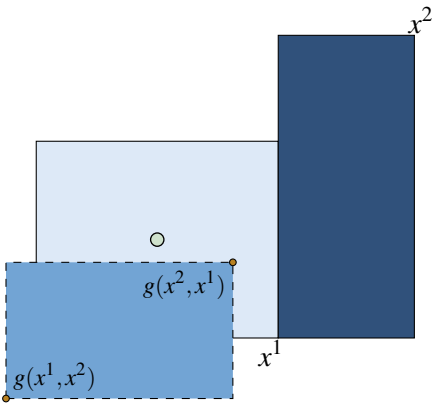
$$x[k+1] = F(x[k]) + w[k]$$



- Maximizing transition probability: Shift disturbance closest to center of q'
- Minimizing transition probability: Shift disturbance furthest from center of q'
- Integrate disturbance PDF over q'

Computing transition probabilities

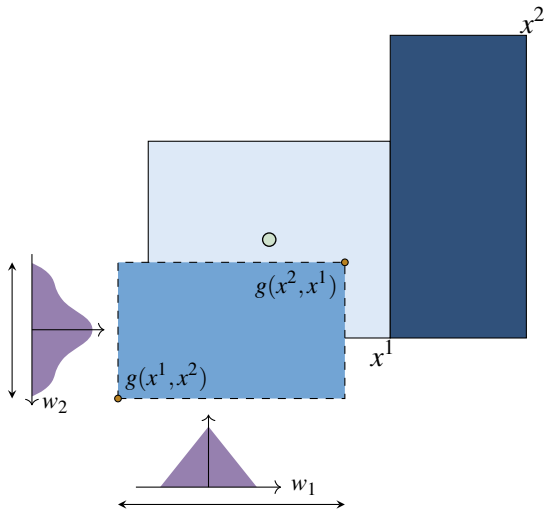
$$x[k+1] = F(x[k]) + w[k]$$



- Maximizing transition probability: Shift disturbance closest to center of q'
- Minimizing transition probability: Shift disturbance furthest from center of q'
- Integrate disturbance PDF over q'

Computing transition probabilities

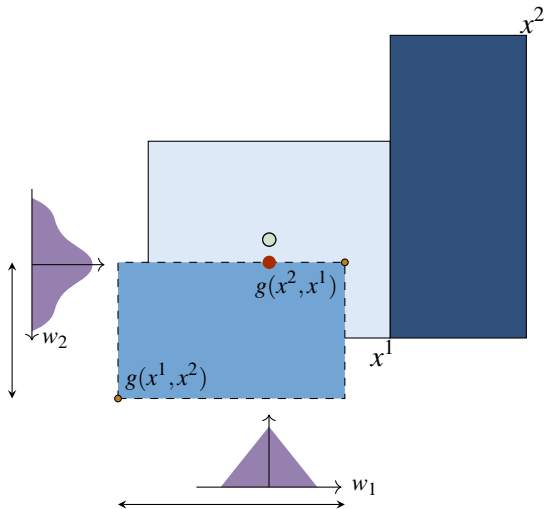
$$x[k+1] = F(x[k]) + w[k]$$



- Maximizing transition probability: Shift disturbance closest to center of q'
- Minimizing transition probability: Shift disturbance furthest from center of q'
- Integrate disturbance PDF over q'

Computing transition probabilities

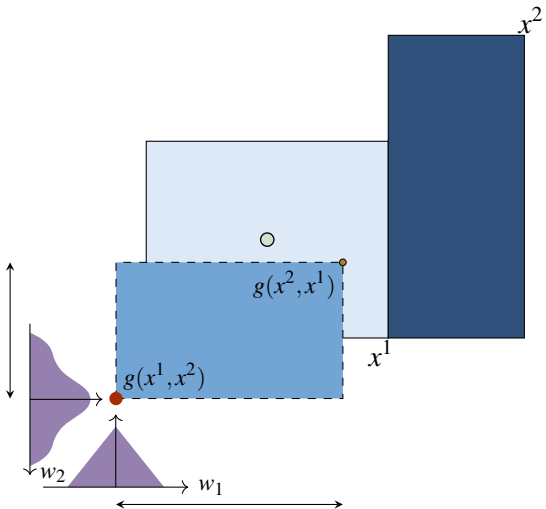
$$x[k+1] = F(x[k]) + w[k]$$



- **Maximizing transition probability: Shift disturbance closest to center of q'**
- Minimizing transition probability: Shift disturbance furthest from center of q'
- Integrate disturbance PDF over q'

Computing transition probabilities

$$x[k+1] = F(x[k]) + w[k]$$



- Maximizing transition probability: Shift disturbance closest to center of q'
- **Minimizing transition probability: Shift disturbance furthest from center of q'**
- Integrate disturbance PDF over q'

BMDP abstraction of mixed monotone systems

Theorem: Given partition $\mathcal{Q}$. Let s^q be center of partition $q \in \mathcal{Q}$, $\check{r}^q = g(x^{1,q}, x^{2,q})$ be lower corner of reach set for $q \in \mathcal{Q}$, $\hat{r}^q = g(x^{2,q}, x^{1,q})$ be upper corner of reach set for $q \in \mathcal{Q}$. Define

$$\hat{T}(q,q') = \prod_{i=1}^{n} \int_{x_i^{1,q'}}^{x_i^{2,q'}} \mathsf{PDF}_{w_i}(x_i - s_{i,max}^{q,q'}) dx_i, \qquad \check{T}(q,q') = \prod_{i=1}^{n} \int_{x_i^{1,q'}}^{x_i^{2,q'}} \mathsf{PDF}_{w_i}(x_i - s_{i,min}^{q,q'}) dx_i$$

where

$$s_{i,max}^{q,q'} = \begin{cases} s_i^q, & \text{if } s_i^q \in [\check{r}_i^q, \hat{r}_i^q] \\ \hat{r}_i^q & \text{if } s_i^q > \hat{r}_i^q \\ \check{r}_i^q & \text{if } s_i^q < \check{r}_i^q \end{cases}, \quad s_{i,min}^{q,q'} = \begin{cases} \check{r}_i^q & \text{if } s_i^q < (\hat{r}_i^q + \check{r}_i^q)/2 \\ \hat{r}_i^q & \text{otherwise.} \end{cases}$$

Then $\mathcal{Q}, \hat{T}, \check{T}$ is a BMDP abstraction.

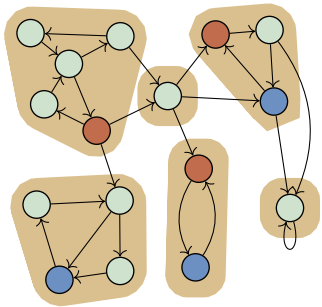
Reach-avoid verification in a Markov chain

Given Markov chain with states Q, bad states $B \subset Q$, good states $G \subset Q$, find the probability of:

- reaching a good (blue) state infinitely often, while
- encountering a bad (red) state finitely many times (eventually avoid)

Standard approach:

- Identify strongly connected components (SCC) via qualitative graph analysis
- Compute probability of reaching **accepting** bottom SCCs
- **Winning component**: states that reach accepting BSCC w.p. 1



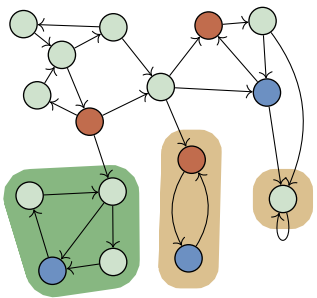
Reach-avoid verification in a Markov chain

Given Markov chain with states Q, bad states $B \subset Q$, good states $G \subset Q$, find the probability of:

- reaching a good (blue) state infinitely often, while
- encountering a bad (red) state finitely many times (eventually avoid)

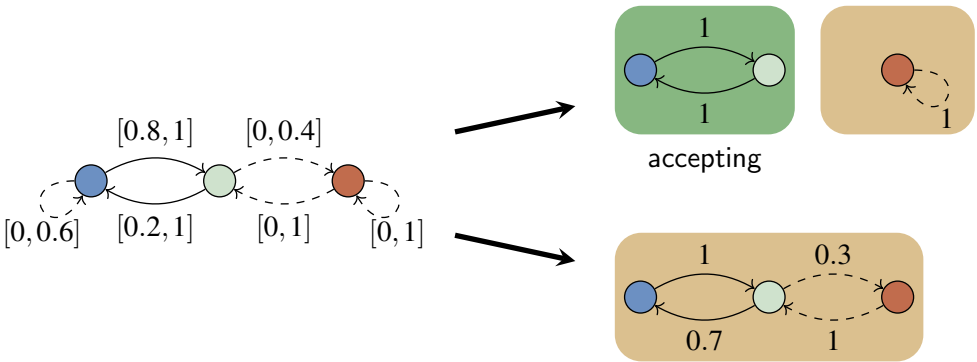
Standard approach:

- Identify strongly connected components (SCC) via qualitative graph analysis
- Compute probability of reaching **accepting** bottom SCCs
- **Winning component**: states that reach accepting BSCC w.p. 1



Reach-avoid verification in bounded-parameter Markov chains

- Edges with lower bound=zero affect qualitative graph structure

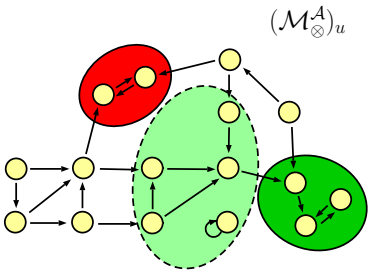


Solving BDMP verification with largest winning and losing components

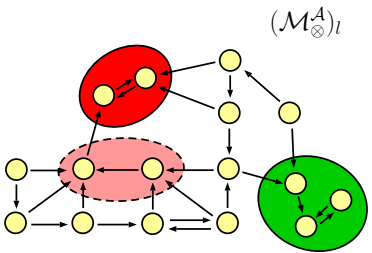
Theorem: Lower and upper bounds of reach-avoid are computed by:

- Upper bound = probability of reaching largest winning component
- Lower bound = 1−(probability of reaching largest losing component)

Best case

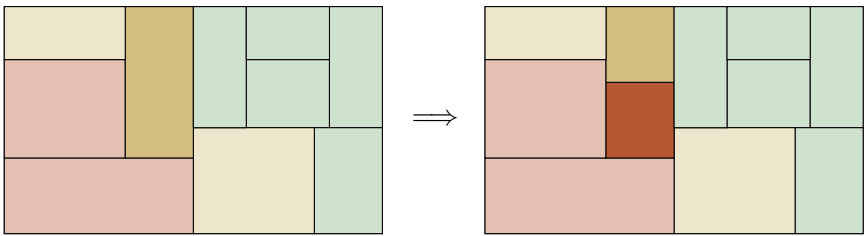


Worst case

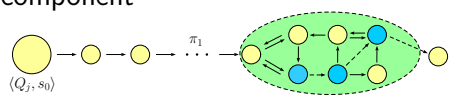


State-space refinement

Refining partition to reduce uncertain states



- **Observation:** it is important to refine intelligently
- **Heuristic algorithm** that assigns scores based on potential to make/break winning or losing component



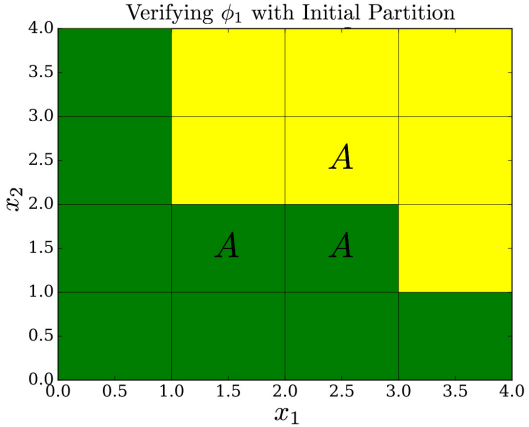
Example

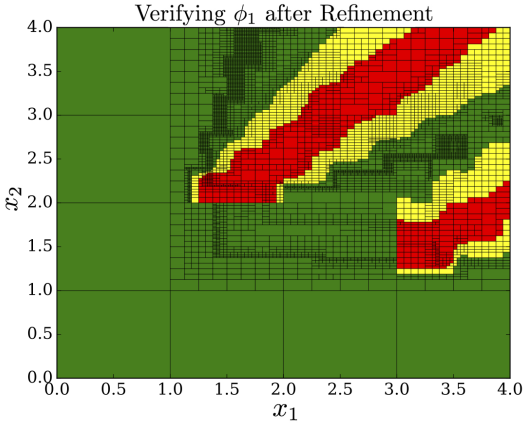
$$x_1[k+1] = x_1[k] + (-a x_1[k] + x_2[k]) \cdot \Delta T + w_1$$
$$x_2[k+1] = x_2[k] + \left(\frac{(x_1[k])^2}{(x_1[k])^2 + 1} - b x_2[k] \right) \cdot \Delta T + w_2,$$

- w_1, $w_2 \sim$ truncated Gaussians
- Two specifications:
  - At least 80% chance of: remaining in an A state for 2 time steps after entering an A state
    $\mathcal{P}_{\geq 0.80}[\Box((\neg A \wedge \bigcirc A) \rightarrow (\bigcirc\bigcirc \wedge \bigcirc\bigcirc\bigcirc A))]$
  - Less than 90% chance of: reaching a B state if it eventually always remains in A state, and always stays outside of B state if it reaches a C state
    $\mathcal{P}_{\leq 0.90}[(\Diamond\Box A \rightarrow \Diamond B) \wedge (\Diamond C \rightarrow \Box \neg B)]$

Example 1: $\mathcal{P}_{\geq 0.80}[\Box((\neg A \wedge \bigcirc A) \rightarrow (\bigcirc\bigcirc \wedge \bigcirc\bigcirc\bigcirc A))]$

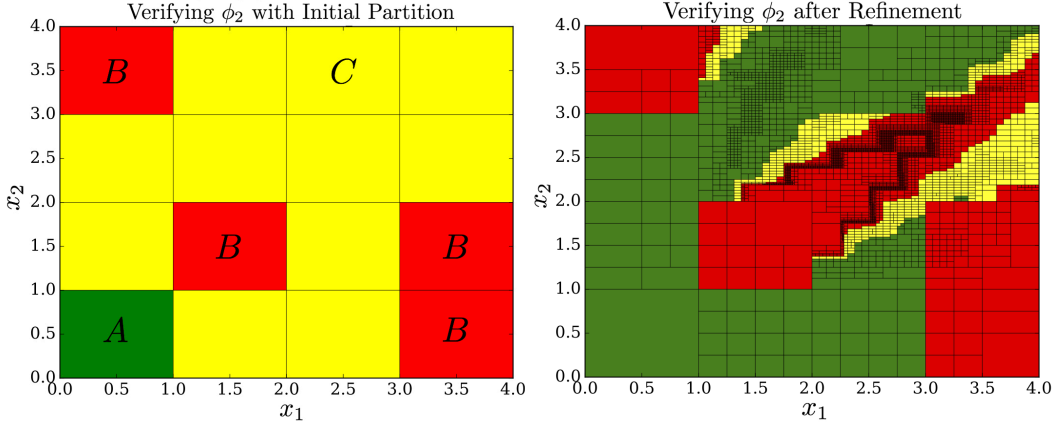
At least 80% chance of: remaining in an A state for 2 time steps after entering an A state





Example 2: $\mathcal{P}_{\leq 0.90}[(\Diamond\Box A \rightarrow \Diamond B) \wedge (\Diamond C \rightarrow \Box\neg B)]$

Less than 90% chance of: reaching a B state if it eventually always remains in A state, and always stays outside of B state if it reaches a C state



Conclusions

1. Bounded Markov Decision Processes (BMDPs) are a good model for formal methods applied to dynamical systems
2. Introduces several challenges:
   - How to compute BMDP abstractions?
   - How to reason about BMDPs?
   - How to intelligently refine BMDP abstractions?
3. Synthesis with continuous inputs (*i.e.*, $u \in \mathbb{R}^m$) introduces new and unique challenges (forthcoming paper)

- Max Dutreix, S. Coogan, "Specification-guided verification and abstraction refinement of mixed-monotone stochastic systems" arXiv, in submission.
- Max Dutreix, S. Coogan, "Efficient verification for stochastic mixed monotone systems," International Conference on Cyber-Physical Systems (ICCPS), 2018.
- S. Coogan, Murat Arcak, "Efficient finite abstraction of mixed monotone systems," International Conference on Hybrid Systems: Computation and Control (HSCC), 2015.