Simons Institute Bootcamp 2021

Verification and Control of Partially Observable Probabilistic Systems

Gethin Norman

University of Glasgow

Outline – Probabilistic Systems

Part 0: **Introduction to probabilistic systems and model checking**

Part 1: **Discrete-time Markov chains (DTMCs)**

- paths and probabilities for DTMCs
- probabilistic reachability and expected reachability
- extension to LTL and automata based properties

Part 2: **Markov decision processes (MDPs)**

- paths, strategies and probabilities for MDPs
- probabilistic reachability for MDPs
- extension to LTL and automata based properties

Part 3: **Partially observable probabilistic systems**

- including a recap of the background

Motivation

Guaranteeing the correctness of complex systems one needs to take into account quantitative aspects

Modelling of probabilistic phenomena

- e.g. failure rates for physical components or uncertainty arising from unreliable sensing of a continuous environment

Timing characteristics

- e.g. time-outs or delays in communication or security protocols

A further complication: such systems often require nondeterminism

- behaviour depends on inputs or instructions from some external entity such as a controller or scheduler
- and for modelling concurrency and abstraction

Motivation

Automated verification techniques have been successfully used to analyse quantitative properties of such systems

Models:

- Markov decision processes (MDPs) assuming a discrete model of time
- probabilistic timed automata (PTAs) assuming a dense model of time

Two dual problems:

- verification of some formally specified property for all possible resolutions of nondeterminism or find optimal (min/max) value
- synthesis of a controller/strategy (i.e. means to resolve nondeterminism) under which a property is guaranteed or value is optimised

Motivation

An important consideration is the extent to which the system's state is observable to the entity controlling it

Examples:

- when verifying a security protocol is secure under all potential attacks, essential to model the fact that certain data is not visible to the attacker
- a controller for a robot can only make decisions based on information that can be physically observed (i.e. through its sensors)
- when routing packets, a scheduler often cannot use channel state information as it is unavailable due to the delays and costs associated with channel probing

Motivation

Partially observable MDPs (POMDPs) are a natural way to extend MDPs in order to tackle this problem in the discrete time case

However the analysis of POMDPs is considerably more difficult

- key problems are undecidable

The use of POMDPs is common in fields such as AI and planning

- but focus is on discounted and finite horizon problems

Limited progress in the development of practical techniques for formal verification or exploration of their applicability

Overview

Techniques for the verification and control of partially observable, probabilistic systems under both discrete and dense models of time

Approximate analysis of a finite-state POMDP

- the result is a pair of lower and upper bounds on the property of interest
  - minimum/maximum reachability probability or expected reward
  - as well as the synthesis of a ‘optimal’ controller/strategy
- if the results are not precise enough, we can refine and repeat

Extend to partially observable probabilistic timed automata (POPTAs)

- extends PTAs with notion of partial observability (from POMDPs)
- semantics of a POPTA is an (uncountable) infinite-state POMDP
- develop a digital clocks discretisation for POPTAs
- reduces the analysis of a POPTA to a finite-state POMDP

Outline

MDPs

POMDPs

PTAs

POPTAs

Implementation and experimental results

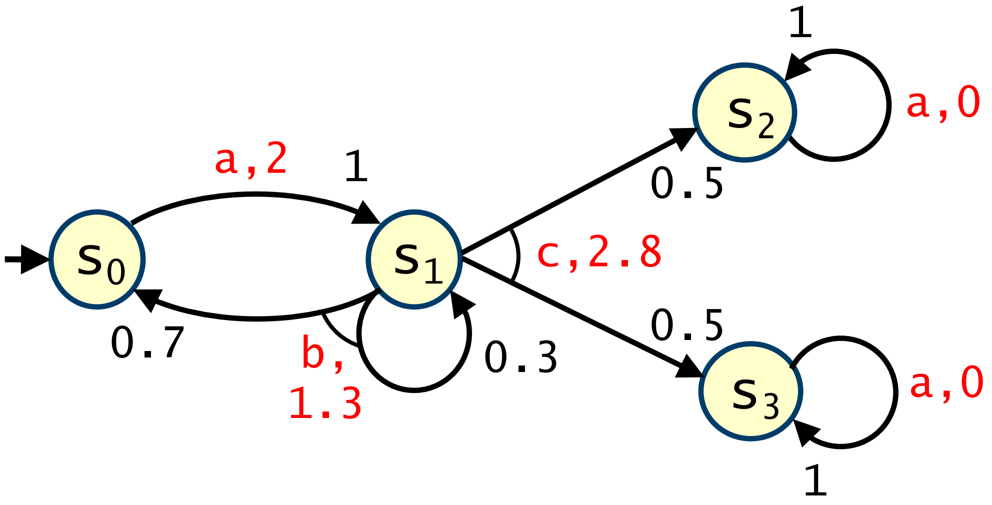
Conclusions

Markov decision processes

Model nondeterministic as well as probabilistic behaviour

An MDP is a tuple $M = (S, s_0, A, P, R)$ where

- S is a state space and s_0 an initial state
- A is an action alphabet
- $P: (S \times A) \rightarrow Dist(S)$ is a (partial) transition probability relation
  - in state s, action a is available (can be performed) if $P(s,a)$ is defined
- $R: (S \times A) \rightarrow \mathbb{R}$ is a reward function



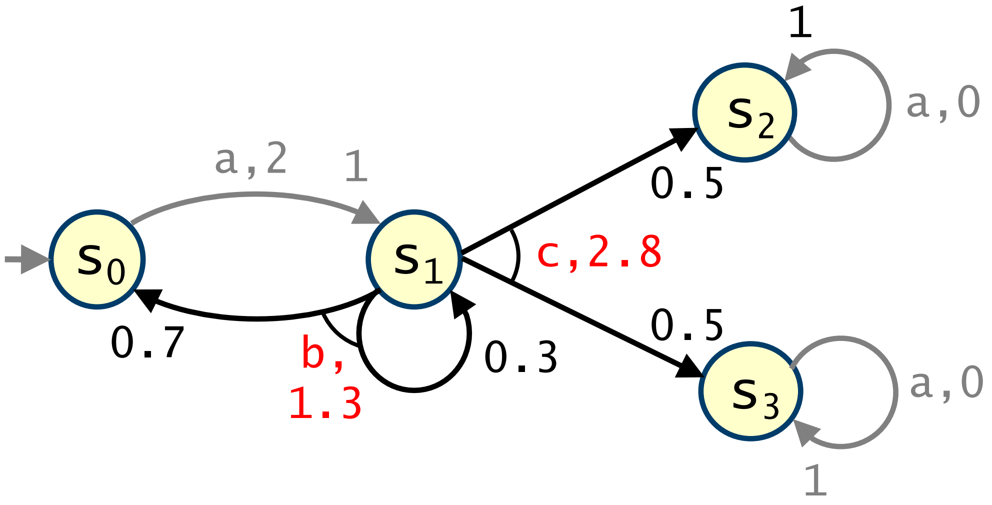
Markov decision processes

In state s nondeterministic choice over available actions

- i.e. actions a for which P(s,a) is defined

If action a is chosen, then from s in the next discrete time step

- there is a transition to state t with probability P(s,a)(t)
- reward R(s,a) is accumulate



MDPs – Paths and strategies

A path of an MDP is a sequence

$$\pi = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2}$$

- such that $P(s_i,a_i)$ is defined and $P(s_i,a_i)(s_{i+1})>0$ for all $i\geq 0$
- a path resolves both the probabilistic and nondeterministic choices
- represents an execution of the system

A strategy (aka. 'scheduler', 'controller' or 'adversary') of an MDP

- is a resolution of the nondeterminism only
- given a finite path (history), strategy chooses the next action to perform

Under a strategy the behaviour of an MDP is fully probabilistic

- i.e. is a DTMC
- can therefore reason about the probabilities of events over paths

MDPs – Optimal reachability values

Probabilistic and expected reachability are the fundamental concepts in quantitative verification

- probability of reaching target T (set of states) under a strategy σ
- expected reward accumulated before reaching target T under a strategy σ
- consider optimal (minimum or maximum) value over all strategies
- fundamental in the analysis of general temporal logic properties including
  - until, globally and more general LTL and automata-based properties
  - expected time/step bounded cumulative and instant reward properties

Efficient algorithms (and tool support) exists

- e.g. using value iteration for the computation
- computes both the optimal value and an optimal strategy
  - deterministic memoryless strategies are sufficient

Much more detail in the video lectures for parts 1 and 2

MDPs – Value iteration

Given function $V:S\rightarrow\mathbb{R}$, value iteration for expected reachability with target T, corresponds to repeatedly performing for each state s

$$V(s) = \begin{cases} 0 & \text{if } s\in T \\ \mathrm{opt}_{a\in A}\{\ R(s,a) + \Sigma_{t\in S}\ P(s,a)(t)\cdot V(t)\ \} & \text{otherwise} \end{cases}$$

where **opt** is either **min** or **max**

MDPs – Value iteration

Given function $V: S \to \mathbb{R}$, value iteration for expected reachability with target T, corresponds to repeatedly performing for each state s

$$V(s) = \begin{cases} 0 & \text{if } s \in T \\ \mathrm{opt}_{a \in A} \{ R(s,a) + \Sigma_{t \in S} P(s,a)(t) \cdot V(t) \} & \text{otherwise} \end{cases}$$

where **opt** is either **min** or **max**

have reached the target, therefore reward accumulated before reaching the target is 0

MDPs – Value iteration

Given function $V:S\rightarrow\mathbb{R}$, value iteration for expected reachability with target T, corresponds to repeatedly performing for each state s

$$V(s) = \begin{cases} 0 & \text{if } s\in T \\ \mathrm{opt}_{a\in A}\{\ R(s,a) + \Sigma_{t\in S}\ P(s,a)(t)\cdot V(t)\ \} & \text{otherwise} \end{cases}$$

where opt is either min or max

not reached target

MDPs – Value iteration

Given function $V:S\to\mathbb{R}$, value iteration for expected reachability with target T, corresponds to repeatedly performing for each state s

$$V(s) = \begin{cases} 0 & \text{if } s\in T \\ \text{opt}_{a\in A}\{\ R(s,a) + \Sigma_{t\in S}\ P(s,a)(t)\cdot V(t)\ \} & \text{otherwise} \end{cases}$$

where **opt** is either **min** or **max**

reward accumulated performing action a in state s

MDPs – Value iteration

Given function V:S→ℝ, value iteration for expected reachability with target T, corresponds to repeatedly performing for each state s

$$V(s) = \begin{cases} 0 & \text{if } s \in T \\ \mathrm{opt}_{a \in A}\{\ R(s,a) + \Sigma_{t \in S}\ P(s,a)(t) \cdot V(t)\ \} & \text{otherwise} \end{cases}$$

where **opt** is either **min** or **max**

reward accumulated performing action a in state s

the probability of making a transition to t after performing a in state s

multiplied by the expected reward of reaching the target from t

MDPs – Value iteration

Given function $V : S \to \mathbb{R}$, value iteration for expected reachability with target T, corresponds to repeatedly performing for each state s

$$V(s) = \begin{cases} 0 & \text{if } s \in T \\ \text{opt}_{a \in A}\{\, R(s,a) + \Sigma_{t \in S}\, P(s,a)(t) \cdot V(t) \,\} & \text{otherwise} \end{cases}$$

where **opt** is either **min** or **max**

reward accumulated performing action a in state s

the probability of making a transition to t after performing a in state s

multiplied by the expected reward of reaching the target from t

sum over all states t

MDPs – Value iteration

Given function V:S→ℝ, value iteration for expected reachability with target T, corresponds to repeatedly performing for each state s

$$V(s) = \begin{cases} 0 & \text{if } s \in T \\ \text{opt}_{a \in A}\{\ R(s,a) + \Sigma_{t \in S}\ P(s,a)(t) \cdot V(t)\ \} & \text{otherwise} \end{cases}$$

where **opt** is either **min** or **max**

reward accumulated performing action a in state s

the probability of making a transition to t after performing a in state s

multiplied by the expected reward of reaching the target from t

sum over all states t

optimal value over all possible action a

MDPs – Value iteration

Given function $\mathtt{V:S\to\mathbb{R}}$, value iteration for expected reachability with target **T**, corresponds to repeatedly performing for each state **s**

$$\mathtt{V(s)} = \begin{cases} 0 & \text{if } s \in T \\ \mathrm{opt}_{a\in A}\{\ R(s,a) + \Sigma_{t\in S}\ P(s,a)(t)\cdot V(t)\ \} & \text{otherwise} \end{cases}$$

where **opt** is either **min** or **max**

- under certain restrictions, repeatedly computing these values will converge to the optimal expected reachability values for all states

The case of probabilistic reachability is very similar (see the videos)

Outline

MDPs

POMDPs

PTAs

POPTAs

Implementation and experimental results

Conclusions

Partially observable Markov decision processes

POMDPs extend MDPs by restricting the extent to which their current state can be observed

- several different definitions of observability in the literature

A POMDP is a tuple $M=(S,s_0,A,P,R,O,obs)$ where:

- (S,s_0,A,P,R) is an MDP
- O is a finite set of observations
- $obs : S \rightarrow O$ is a function labelling of states with observations

Requirement: for observationally equivalent states, the same actions are available

- i.e. for states s and s' such that $obs(s)=obs(s')$
- if the available actions were different, then a strategy could differentiate the states, and therefore they would not be observationally equivalent

POMDPs

Notions of paths and strategies for MDPs transfer directly to POMDPs

However, for a POMDP we restrict to **observation-based strategies**

- must make the same choices for observationally equivalent paths

Formally for paths

$$\pi \;=\; s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \cdots \xrightarrow{a_{n-1}} s_n$$

$$\pi' = s_0' \xrightarrow{a_0'} s_1' \xrightarrow{a_1'} s_2' \xrightarrow{a_2'} \cdots \xrightarrow{a_{n-1}'} s_n'$$

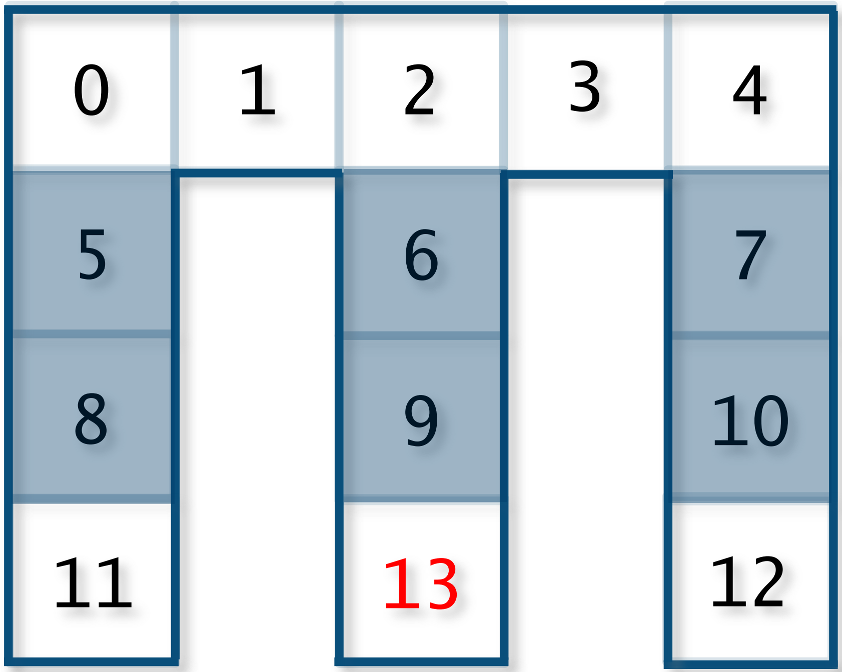
- if $obs(s_i)=obs(s_i')$ and $a_i=a_i'$ for all i, then any strategy σ of the POMDP must choose the same action after the paths π and π'

Optimal strategies for probabilistic and expected reachability are now history dependent (deterministic strategies are still sufficient)

POMDP – Maze example

Robot placed uniformly at random in a maze and tries to reach target

- based on [McCallum 1993]
- target is location 13
- four actions a robot can perform
  - north, south, east and west
- the robot cannot see its current location, only surrounding walls
  - e.g. the locations labelled 5-10 yield the same observation
- strategy choices based only on walls it can sees (and seen previously)



Optimal expected number steps to reach the target is 5.23

- for the fully observable model (i.e. MDP) the optimal expected number of steps equals 4.00

POMDPs – Optimal reachability values

For POMDPs, determining optimal reachability probabilities and expected rewards is undecidable, making exact solution intractable

A useful construction for a POMDP M is that of its belief MDP B(M)

- B(M) is a (fully observable) MDP
- has the same optimal probabilistic and expected reachability values
- cost of reducing a POMDP to an MDP: continuous (uncountable) state space
  - states are 'beliefs' (probability distributions over the state space of M)

we may not know which observationally-equivalent state we are in, however we can determine the likelihood, based on the behaviour of the POMDP, actions performed and what we have observed

POMDPs – Belief MDP

Consider belief MDP $\mathbf{B(M)}$ of POMDP $\mathbf{M=(S,s_0,A,P,R,O,obs)}$

For belief **b** (distribution over **S**), observation **o** and action **a** we have

- probability of observing o after performing action a in belief b

$$P[o|a,b] = \Sigma_{s\in S}\ b(s)\cdot\left(\ \Sigma_{t\in S\wedge obs(t)=o}\ P(s,a)(t)\ \right)$$

POMDPs – Belief MDP

Consider belief MDP $\mathtt{B(M)}$ of POMDP $\mathtt{M=(S,s_0,A,P,R,O,obs)}$

For belief **b** (distribution over S), observation **o** and action **a** we have

- probability of observing o after performing action a in belief b

$$P[o|a,b] = \Sigma_{s\in S}\ b(s)\cdot(\ \Sigma_{t\in S\wedge obs(t)=o}\ P(s,a)(t)\ )$$

probability of observing o from s when performing a

POMDPs – Belief MDP

Consider belief MDP $B(M)$ of POMDP $M=(S,s_0,A,P,R,O,obs)$

For belief **b** (distribution over S), observation **o** and action **a** we have

- probability of observing o after performing action a in belief b

$$P[o|a,b] = \Sigma_{s\in S}\ b(s)\cdot\left(\ \Sigma_{t\in S\wedge obs(t)=o}\ P(s,a)(t)\ \right)$$

probability of being in s when performing a

probability of observing o from s when performing a

POMDPs – Belief MDP

Consider belief MDP `B(M)` of POMDP $M=(S,s_0,A,P,R,O,obs)$

For belief b (distribution over S), observation o and action a we have

- probability of observing o after performing action a in belief b

$$P[o|a,b] = \sum_{s\in S} b(s)\cdot\left(\sum_{t\in S\wedge obs(t)=o} P(s,a)(t) \right)$$

sum over all states s we can be in when performing a

probability of being in s when performing a

probability of observing o from s when performing a

POMDPs – Belief MDP

Consider belief MDP $\mathtt{B(M)}$ of POMDP $\mathtt{M=(S,s_0,A,P,R,O,obs)}$

For belief **b** (distribution over S), observation **o** and action **a** we have

- probability of observing o after performing action a in belief b

$$P[o|a,b] = \Sigma_{s\in S}\ b(s)\cdot(\ \Sigma_{t\in S\wedge obs(t)=o}\ P(s,a)(t)\ )$$

sum over all states s we can be in when performing a

probability of being in s when performing a

probability of observing o from s when performing a

POMDPs – Belief MDP

Consider belief MDP $\mathtt{B(M)}$ of POMDP $\mathtt{M=(S,s_0,A,P,R,O,obs)}$

For belief b (distribution over S), observation o and action a we have

- probability of observing o after performing action a in belief b

$$P[o|a,b] = \Sigma_{s\in S}\ b(s)\cdot(\ \Sigma_{t\in S \wedge obs(t)=o}\ P(s,a)(t)\ )$$

- the belief reached from b after performing a and observing o, denoted $b^{a,o}$, is such that for any $t\in S$

$$b^{a,o}(t) = \begin{cases} \dfrac{\Sigma_{s\in S}\ P(s,a)(t)\cdot b(s)}{P[o|a,b]} & \text{if } obs(t)=o \\ 0 & \text{otherwise} \end{cases}$$

POMDPs – Belief MDP

Consider belief MDP $B(M)$ of POMDP $M=(S,s_0,A,P,R,O,obs)$

For belief b (distribution over S), observation o and action a we have

- probability of observing o after performing action a in belief b

$$P[o|a,b] = \Sigma_{s\in S}\ b(s)\cdot(\ \Sigma_{t\in S\wedge obs(t)=o}\ P(s,a)(t)\ )$$

- the belief reached from b after performing a and observing o, denoted $b^{a,o}$, is such that for any $t\in S$

$$b^{a,o}(t) = \begin{cases} \dfrac{\Sigma_{s\in S}\ P(s,a)(t)\cdot b(s)}{P[o|a,b]} & \text{if } obs(t)=o \\ 0 & \text{otherwise} \end{cases}$$

probability we believe we are in state t

POMDPs – Belief MDP

Consider belief MDP $\mathtt{B(M)}$ of POMDP $\mathtt{M=(S,s_0,A,P,R,O,obs)}$

For belief b (distribution over S), observation o and action a we have

- probability of observing o after performing action a in belief b

$$\mathtt{P[o|a,b]} = \Sigma_{s\in S}\, \mathtt{b(s)}\cdot\left(\Sigma_{t\in S\wedge obs(t)=o}\, \mathtt{P(s,a)(t)} \right)$$

- the belief reached from b after performing a and observing o, denoted $b^{a,o}$, is such that for any $t\in S$

$$\mathtt{b^{a,o}(t)} = \begin{cases} \dfrac{\Sigma_{s\in S}\, \mathtt{P(s,a)(t)\cdot b(s)}}{\mathtt{P[o|a,b]}} & \mathtt{if\ obs(t)=o} \\ 0 & \mathtt{otherwise} \end{cases}$$

observed o so must reach a state for which o is observed

POMDPs – Belief MDP

Consider belief MDP $\mathtt{B(M)}$ of POMDP $\mathtt{M=(S,s_0,A,P,R,O,obs)}$

For belief b (distribution over S), observation o and action a we have

- probability of observing o after performing action a in belief b

$$P[o|a,b] = \Sigma_{s\in S}\ b(s)\cdot(\ \Sigma_{t\in S \wedge obs(t)=o}\ P(s,a)(t)\ )$$

- the belief reached **from b** after performing a and observing o, denoted $b^{a,o}$, is such that for any $t\in S$

$$b^{a,o}(t) = \begin{cases} \dfrac{\Sigma_{s\in S}\ P(s,a)(t)\cdot b(s)}{P[o|a,b]} & \text{if } obs(t)=o \\ 0 & \text{otherwise} \end{cases}$$

probability of being in s when performing a

POMDPs – Belief MDP

Consider belief MDP $\mathtt{B(M)}$ of POMDP $\mathtt{M=(S,s_0,A,P,R,O,obs)}$

For belief b (distribution over S), observation o and action a we have

- probability of observing o after performing action a in belief b

$$P[o|a,b] = \Sigma_{s\in S}\ b(s)\cdot(\ \Sigma_{t\in S\wedge obs(t)=o}\ P(s,a)(t)\ )$$

- the belief reached from b after performing a and observing o, denoted $b^{a,o}$, is such that for any $t\in S$

$$b^{a,o}(t) = \begin{cases} \dfrac{\Sigma_{s\in S}\ P(s,a)(t)\cdot b(s)}{P[o|a,b]} & \text{if } obs(t)=o \\ 0 & \text{otherwise} \end{cases}$$

probability of reaching t from s when performing a

probability of being in s when performing a

POMDPs – Belief MDP

Consider belief MDP $B(M)$ of POMDP $M=(S,s_0,A,P,R,O,obs)$

For belief b (distribution over S), observation o and action a we have

- probability of observing o after performing action a in belief b

$$P[o|a,b] = \Sigma_{s\in S}\ b(s)\cdot(\ \Sigma_{t\in S\wedge obs(t)=o}\ P(s,a)(t)\ )$$

- the belief reached from b after performing a and observing o, denoted $b^{a,o}$, is such that for any $t \in S$

$$b^{a,o}(t) = \begin{cases} \dfrac{\Sigma_{s\in S}\ P(s,a)(t)\cdot b(s)}{P[o|a,b]} & \text{if } obs(t)=o \\ 0 & \text{otherwise} \end{cases}$$

sum over all states s we can be in when performing a

probability of reaching t from s when performing a

probability of being in s when performing a

POMDPs – Belief MDP

Consider belief MDP $\mathtt{B(M)}$ of POMDP $\mathtt{M=(S,s_0,A,P,R,O,obs)}$

For belief b (distribution over S), observation o and action a we have

- probability of observing o after performing action a in belief b

$$P[o|a,b] = \Sigma_{s\in S}\ b(s)\cdot(\ \Sigma_{t\in S\wedge obs(t)=o}\ P(s,a)(t)\ )$$

- the belief reached from b after performing a and observing o, denoted $b^{a,o}$, is such that for any $t\in S$

$$b^{a,o}(t) = \begin{cases} \dfrac{\Sigma_{s\in S}\ P(s,a)(t)\cdot b(s)}{P[o|a,b]} & \text{if } obs(t)=o \\ 0 & \text{otherwise} \end{cases}$$

probability of reaching t from belief b when performing a

POMDPs – Belief MDP

Consider belief MDP $\mathtt{B(M)}$ of POMDP $\mathtt{M=(S,s_0,A,P,R,O,obs)}$

For belief **b** (distribution over S), observation **o** and action **a** we have

- probability of observing o after performing action a in belief b

$$\mathtt{P[o|a,b]} = \Sigma_{s\in S}\ b(s)\cdot\left(\ \Sigma_{t\in S\wedge obs(t)=o}\ P(s,a)(t)\ \right)$$

- the belief reached from b after performing a and observing o, denoted $b^{a,o}$, is such that for any $t\in S$

$$b^{a,o}(t) = \begin{cases} \dfrac{\Sigma_{s\in S}\ P(s,a)(t)\cdot b(s)}{P[o|a,b]} & \text{if } obs(t)=o \\ 0 & \text{otherwise} \end{cases}$$

probability of reaching t from belief b when performing a

probability of observing o from belief b when performing a

POMDPs – Belief MDP

Consider belief MDP $\mathtt{B(M)}$ of POMDP $\mathtt{M=(S,s_0,A,P,R,O,obs)}$

For belief b (distribution over S), observation o and action a we have

- probability of observing o after performing action a in belief b

$$P[o|a,b] = \Sigma_{s\in S}\ b(s)\cdot(\ \Sigma_{t\in S\wedge obs(t)=o}\ P(s,a)(t)\ )$$

- the belief reached from b after performing a and observing o, denoted $b^{a,o}$, is such that for any $t\in S$

$$b^{a,o}(t) = \begin{cases} \dfrac{\Sigma_{s\in S}\ P(s,a)(t)\cdot b(s)}{P[o|a,b]} & \text{if } obs(t)=o \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\text{probability of reaching } t \text{ from belief } b \text{ when performing } a \text{ and observing } o}{\text{probability of observing } o \text{ from belief } b \text{ when performing } a}$$

POMDPs – Belief MDP

Consider belief MDP $\mathtt{B(M)}$ of POMDP $\mathtt{M=(S,s_0,A,P,R,O,obs)}$

For belief b (distribution over S), observation o and action a we have

- probability of observing o after performing action a in belief b

$$\mathtt{P[o|a,b]} = \Sigma_{s\in S}\ \mathtt{b(s)}\cdot(\ \Sigma_{t\in S \wedge obs(t)=o}\ \mathtt{P(s,a)(t)}\ )$$

- the belief reached from b after performing a and observing o, denoted $b^{a,o}$, is such that for any $t\in S$

$$b^{a,o}(t) = \begin{cases} \dfrac{\Sigma_{s\in S}\ P(s,a)(t)\cdot b(s)}{P[o|a,b]} & \text{if } obs(t)=o \\ 0 & \text{otherwise} \end{cases}$$

probability of reaching t from belief b when performing a, conditioned on observing o

POMDPs – Belief MDP

Consider belief MDP $\mathtt{B(M)}$ of POMDP $\mathtt{M=(S,s_0,A,P,R,O,obs)}$

For belief **b** (distribution over S), observation **o** and action **a** we have

- probability of observing o after performing action a in belief b

$$\mathtt{P[o|a,b]} = \Sigma_{s\in S}\ \mathtt{b(s)}\cdot(\ \Sigma_{t\in S\wedge obs(t)=o}\ \mathtt{P(s,a)(t)}\ )$$

- the belief reached from b after performing a and observing o, denoted $b^{a,o}$, is such that for any $t\in S$

$$b^{a,o}(t) = \begin{cases} \dfrac{\Sigma_{s\in S}\ P(s,a)(t)\cdot b(s)}{P[o|a,b]} & \text{if } obs(t)=o \\ 0 & \text{otherwise} \end{cases}$$

- reward accumulated after performing action a in belief b

$$\mathtt{R(b,a)} = \Sigma_{s\in S}\ \mathtt{R(s,a)\cdot b(s)}$$

POMDPs – Belief MDP

Consider belief MDP $\mathtt{B(M)}$ of POMDP $\mathtt{M=(S,s_0,A,P,R,O,obs)}$

For belief b (distribution over S), observation o and action a we have

- probability of observing o after performing action a in belief b

$$P[o|a,b] = \Sigma_{s\in S}\ b(s)\cdot(\ \Sigma_{t\in S\wedge obs(t)=o}\ P(s,a)(t)\ )$$

- the belief reached from b after performing a and observing o, denoted $b^{a,p}$, is such that for any $t\in S$

$$b^{a,o}(t) = \begin{cases} \dfrac{\Sigma_{s\in S}\ P(s,a)(t)\cdot b(s)}{P[o|a,b]} & \text{if } obs(t)=o \\ 0 & \text{otherwise} \end{cases}$$

probability of being in s

- reward accumulated after performing action a in belief b

$$R(b,a) = \Sigma_{s\in S}\ R(s,a)\cdot b(s)$$

POMDPs – Belief MDP

Consider belief MDP $B(M)$ of POMDP $M=(S,s_0,A,P,R,O,obs)$

For belief b (distribution over S), observation o and action a we have

- probability of observing o after performing action a in belief b

$$P[o|a,b] = \Sigma_{s\in S}\ b(s)\cdot(\ \Sigma_{t\in S \wedge obs(t)=o}\ P(s,a)(t)\ )$$

- the belief reached from b after performing a and observing o, denoted $b^{a,o}$, is such that for any $t \in S$

$$b^{a,o}(t) = \begin{cases} \dfrac{\Sigma_{s\in S}\ P(s,a)(t)\cdot b(s)}{P[o|a,b]} & \text{if } obs(t)=o \\ 0 & \text{otherwise} \end{cases}$$

reward accumulated when performing action a in state s

probability of being in s

- reward accumulated after performing action a in belief b

$$R(b,a) = \Sigma_{s\in S}\ R(s,a)\cdot b(s)$$

POMDPs – Belief MDP

Consider belief MDP $\mathtt{B(M)}$ of POMDP $\mathtt{M=(S,s_0,A,P,R,O,obs)}$

For belief b (distribution over S), observation o and action a we have

- probability of observing o after performing action a in belief b

$$\mathtt{P[o|a,b]} = \Sigma_{s\in S}\ \mathtt{b(s)}\cdot\big(\ \Sigma_{t\in S\wedge obs(t)=o}\ \mathtt{P(s,a)(t)}\ \big)$$

- the belief reached from b after performing a and observing o, denoted $b^{a,o}$, is such that for any $t\in S$

$$\begin{cases} \dfrac{\Sigma_{s\in S}\ \mathtt{P(s,a)(t)\cdot b(s)}}{\mathtt{P[o|a,b]}} & \mathtt{if\ obs(t)=o} \\ 0 & \mathtt{otherwise} \end{cases}$$

sum over all states s we can be in when performing a

reward accumulated when performing action a in state s

probability of being in s

- reward accumulated after performing action a in belief b

$$\mathtt{R(b,a)} = \Sigma_{s\in S}\ \mathtt{R(s,a)\cdot b(s)}$$

POMDPs – Belief MDP

Consider belief MDP $B(M)$ of POMDP $M=(S,s_0,A,P,R,O,obs)$

For belief b (distribution over S), observation o and action a we have

- probability of observing o after performing action a in belief b

  $$P[o|a,b] = \Sigma_{s\in S}\ b(s)\cdot(\ \Sigma_{t\in S \wedge obs(t)=o}\ P(s,a)(t)\ )$$

- the belief reached from b after performing a and observing o, denoted $b^{a,o}$, is such that for any $t\in S$

  $$\begin{cases} \dfrac{\Sigma_{s\in S}\ P(s,a)(t)\cdot b(s)}{P[o|a,b]} & \text{if } obs(t)=o \\ 0 & \text{otherwise} \end{cases}$$

sum over all states s we can be in when performing a

reward accumulated when performing action a in state s

probability of being in s

- reward accumulated after performing action a in belief b

  $$R(b,a) = \Sigma_{s\in S}\ R(s,a)\cdot b(s)$$

POMDPs – Belief MDP

Recall for MDP and state **s** value iteration corresponds to performing

$$V(s) = \mathrm{opt}_{a \in A} \{ R(s,a) + \Sigma_{t \in S} P(s,a)(t) \cdot V(t) \}$$

For a belief **b** value iteration on **B(M)** corresponds to performing

$$V(b) = \mathrm{opt}_{a \in A} \{ R(b,a) + \Sigma_{o \in O} P[o|a,b] \cdot V(b^{a,o}) \}$$

POMDPs – Belief MDP

Recall for MDP and state **s** value iteration corresponds to performing

$$V(s) = \mathrm{opt}_{a\in A} \{ R(s,a) + \Sigma_{t\in S} P(s,a)(t)\cdot V(t) \}$$

For a belief **b** value iteration on **B(M)** corresponds to performing

$$V(b) = \mathrm{opt}_{a\in A} \{ R(b,a) + \Sigma_{o\in O} P[o|a,b]\cdot V(b^{a,o}) \}$$

– now update based on what we observe as opposed to the successor states using:

POMDPs – Belief MDP

Recall for MDP and state s value iteration corresponds to performing

$$V(s) = \mathrm{opt}_{a \in A} \{ R(s,a) + \Sigma_{t \in S} P(s,a)(t) \cdot V(t) \}$$

For a belief b value iteration on B(M) corresponds to performing

$$V(b) = \mathrm{opt}_{a \in A} \{ R(b,a) + \Sigma_{o \in O} P[o|a,b] \cdot V(b^{a,o}) \}$$

- now update based on what we observe as opposed to the successor states using:
  - $R(b,a)$ reward accumulated after performing action a in belief b

POMDPs – Belief MDP

Recall for MDP and state s value iteration corresponds to performing

$$V(s) = \mathrm{opt}_{a \in A} \{ R(s,a) + \Sigma_{t \in S} P(s,a)(t) \cdot V(t) \}$$

For a belief **b** value iteration on **B(M)** corresponds to performing

$$V(b) = \mathrm{opt}_{a \in A} \{ R(b,a) + \Sigma_{o \in O} P[o|a,b] \cdot V(b^{a,o}) \}$$

- now update based on what we observe as opposed to the successor states using:
  - R(b,a) reward accumulated after performing action a in belief b
  - P[o|a,b] probability of observing o after performing action a in belief b

POMDPs – Belief MDP

Recall for MDP and state s value iteration corresponds to performing

$$V(s) = opt_{a \in A} \{ R(s,a) + \Sigma_{t \in S} P(s,a)(t) \cdot V(t) \}$$

For a belief **b** value iteration on **B(M)** corresponds to performing

$$V(b) = opt_{a \in A} \{ R(b,a) + \Sigma_{o \in O} P[o|a,b] \cdot V(b^{a,o}) \}$$

- now update based on what we observe as opposed to the successor states using:
  - $R(b,a)$ reward accumulated after performing action a in belief b
  - $P[o|a,b]$ probability of observing o after performing action a in belief b
  - $b^{a,o}$ the belief reached from b after performing a and observing o

POMDPs – Grid based methods

The set of beliefs, the set of distributions **Dist(S)**, is the state space

- therefore the state space is uncountable

Need to resort to an approximate solution of the belief MDP

- grid based methods: computes (approximates) values for a finite set of representative beliefs $G = \{g_1,\dots,g_N\}$ whose convex hull is Dist(S)

Requires interpolation

- given arbitrary belief (distribution) b find real constants $\gamma_1,\dots,\gamma_N$ such that

$$b = \gamma_1 \cdot g_1 + \cdots + \gamma_N \cdot g_N$$

  - constants exist as we require the convex hull of G to be Dist(S)

Will discuss a concrete implementation later

- i.e. how to come up with representative beliefs

POMDPs – Grid based methods

Approximate solution: value iteration over the grid $\mathbf{G = \{g_1,\ldots,g_N\}}$

For a grid point $\mathbf{g_i}$ value iteration on $\mathbf{B(M)}$ corresponds to performing

$$V(g_i) = \mathrm{opt}_{a\in A} \{ R(g_i,a) + \Sigma_{o\in O} \mathbf{P}[o|a,g]\cdot V(g_i^{a,o}) \}$$

- as we have seen, we can compute $R(g_i,a)$, $P[o|a,g_i]$ and $g_i^{a,o}$ for each o and a and there are only finitely many observations and actions
- however, no reason for $g_i^{a,o}$ to be a grid point so value $V(g_i^{a,o})$ is unknown
  - $g_i^{a,o}$ the belief reached from g_i after performing a and observing o
  - unless o is a target observation in which case $V(g^{a,o})=0$

POMDPs – Grid based methods

Approximate solution: value iteration over the grid $\mathbf{G = \{g_1,\ldots,g_N\}}$

For a grid point $\mathbf{g_i}$ value iteration on $\mathbf{B(M)}$ corresponds to performing

$$V(g_i) = \text{opt}_{a\in A} \{ R(g_i,a) + \Sigma_{o\in O} \mathbf{P}[o|a,g]\cdot V(g_i^{a,o}) \}$$

- as we have seen, we can compute $R(g_i,a)$, $P[o|a,g_i]$ and $g_i^{a,o}$ for each o and a and there are only finitely many observations and actions
- however, no reason for $g_i^{a,o}$ to be a grid point so value $V(g_i^{a,o})$ is unknown
- by construction of the grid points there exists constants $\gamma_1,\ldots,\gamma_N$ such that $g_i^{a,o} = \gamma_1\cdot g_1 +\cdots + \gamma_N\cdot g_N$
- we therefore instead approximate $V(g_i^{a,o})$ with $\gamma_1\cdot V(g_1) + \cdots + \gamma_N\cdot V(g_N)$

POMDPs – Grid based methods

After performing value iteration which has been proved with converge

- i.e. finding values $V(g_1),\dots,V(g_N)$

We can then synthesise an ‘optimal’ finite-memory strategy σ_{opt}

- the choices of this strategy are built by stepping through the belief MDP
- for the current belief, choosing action that achieves the ‘optimal’ value
- for any belief b the ‘optimal’ action is given by

$$\sigma_{opt}(b) = \mathrm{argopt}_{a\in A}\ \{\ R(b,a) + \Sigma_{o\in O}P[o|a,b]\cdot\underbrace{(\gamma_1\cdot V(g_1)+\cdots+\gamma_N\cdot V(g_N))}_{\boxed{\begin{array}{c}\text{approximation of } V(b^{a,o})\\ \text{where } b^{a,o} = \gamma_1\cdot g_1 + \cdots + \gamma_N\cdot g_N\end{array}}}\ \}$$

Can then build and solve a DTMC using the strategy’s choices

Slight detour

In practice, not usually interested in knowing that all reachability values are between the minimum and maximum values

Instead just interested in one of the optimal reachability values or optimal strategies

When considering worst case behaviour

- maximum probability of an error or expected time/cost
- minimum probability message arrives or expected profit

For controller synthesis the best case is of interest

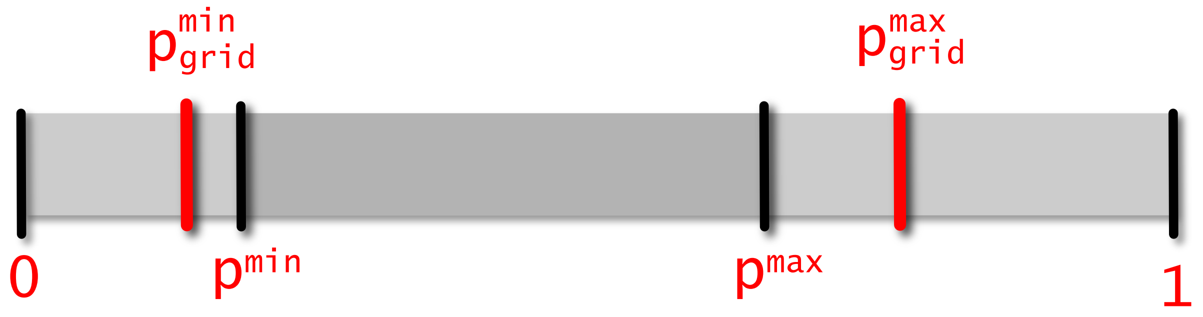
- controller that minimizes probability of an error or expected time/cost
- controller that maximizes probability message arrives or expected profit

POMDPs – Grid based methods

Consider the case of probabilistic reachability

Value iteration on grid yields an over approximation of optimal values

- based on results from [Yu & Bertsekas 2004]

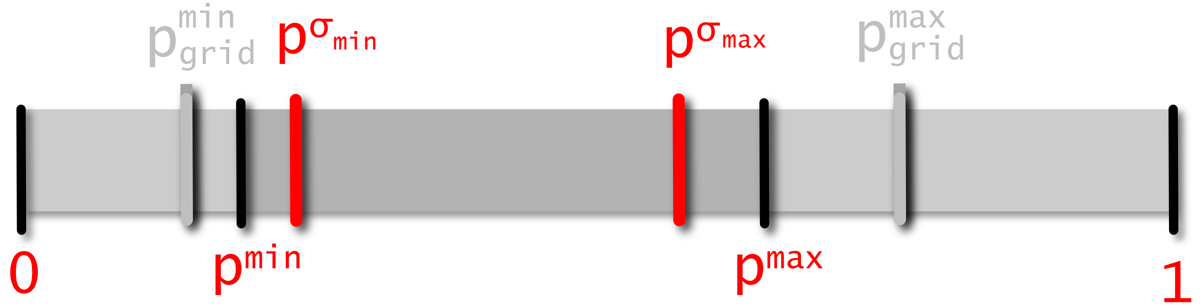


POMDPs – Grid based methods

Consider the case of probabilistic reachability

Value iteration on grid yields an over approximation of optimal values

Synthesize a finite memory strategy σ_{opt} using the obtained results yields under approximation of optimal values



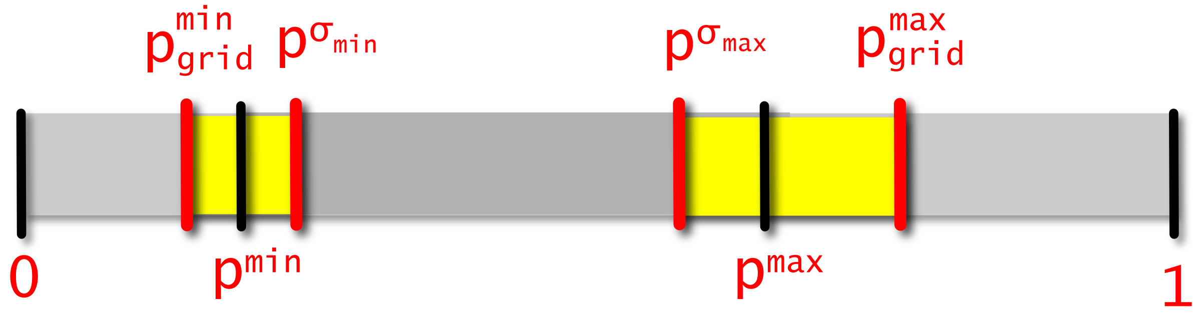
POMDPs – Grid based methods

Consider the case of probabilistic reachability

Value iteration on grid yields an **over approximation** of optimal values

Synthesize a finite memory strategy σ_{opt} using the obtained results yields **under approximation** of optimal values

Gives us two sided bounds for each optimal value and an indication of how 'optimal' is the synthesized strategy σ_{opt}



POMDPs – Grid based methods

Consider the case of probabilistic reachability

Value iteration on grid yields an over approximation of optimal values

Synthesize a finite memory strategy σ_{opt} using the obtained results yields under approximation of optimal values

Gives us two sided bounds for each optimal value and an indication of how ‘optimal’ is the synthesized strategy σ_{opt}

If the bounds/strategy are too coarse can refine the grid and repeat

- no guarantee this will yield tighter bounds (problem is undecidable)
- all we have is asymptotic convergence (converges as we go to infinity)

Interlude – The dining cryptographers problem

Problem

- N cryptographers share a meal
- the meal is either paid for by the master or by one of the cryptographers
- the master decides who pays
- each cryptographer is informed by the master whether or not they pay

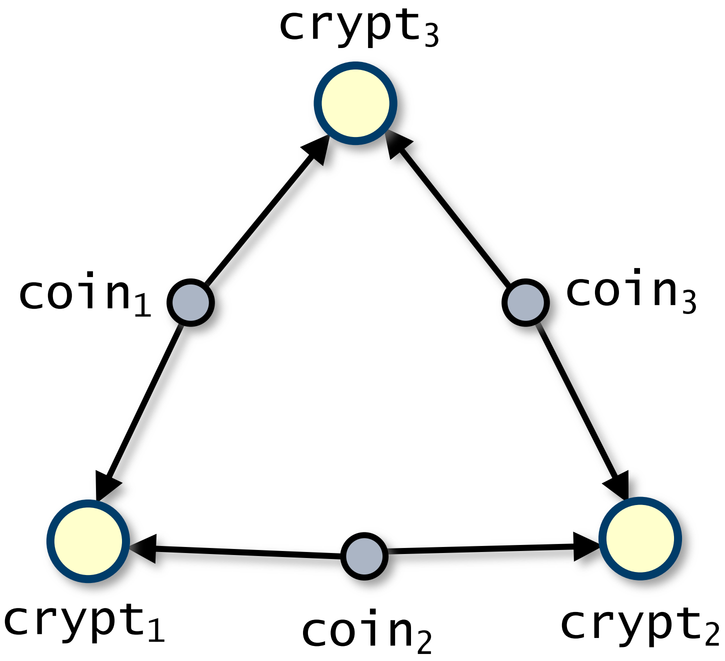
Goal

- the cryptographers would like to know whether the meal is being paid for by the master or one of themselves
- without knowing who is paying and without involving the master

Interlude – The dining cryptographers problem

Chaum's solution [Chaum, 1988]

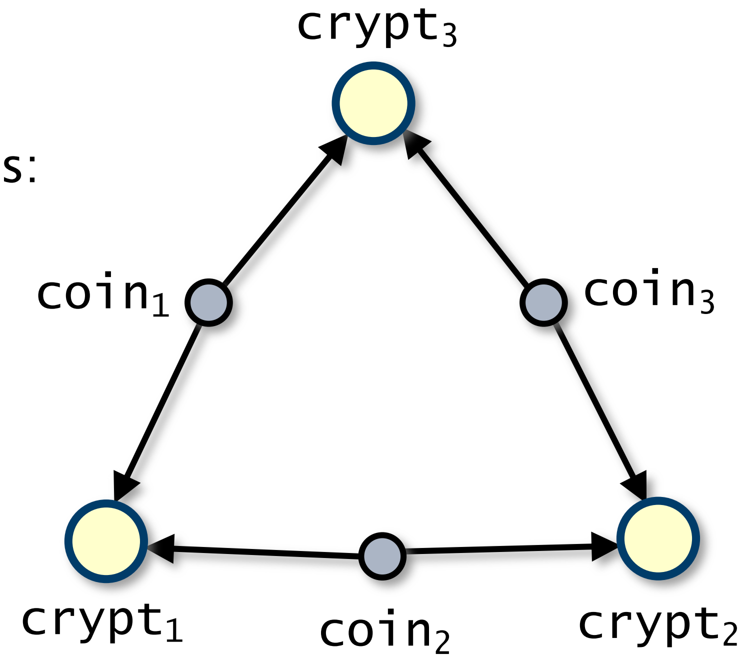
- each cryptographer flips a coin
- tells only their **left** neighbour the value of the coin
- each cryptographer looks at the two coins they can see
  - their own coin and their **right** neighbour's coin
- if the cryptographer is not paying announces:
  - agree if the coins agree
  - disagree otherwise
- if the cryptographer is paying announces:
  - disagree if the coins agree
  - agree otherwise



Interlude – The dining cryptographers problem

Chaum's solution [Chaum, 1988]

- if the cryptographer is not paying announces:
  - agree if the coins agree
  - disagree otherwise
- if the cryptographer is paying announces:
  - agree if the coins agree
  - disagree otherwise



Correctness: an **odd** number of **agree**'s indicates that the master paid while an **even** number indicates that a cryptographer paid

Anonymity: in the latter case, neither the non-paying cryptographers nor any external observer will be able to deduce who is paying

Interlude – The dining cryptographers problem

Without POMDPs can verify in the probabilistic model checker PRISM

- correctness is straightforward
- checking anonymity requires 2^N properties to be verified and is a hack
  - I did the hacking

POMDP model in PRISM

- model is very simple and only one property to check
- one cryptographer guesses who paid after the announcements assuming which of the other $N-1$ cryptographer pays was random chosen
- not a hack

Quick demo...

Outline

MDPs

POMDPs

PTAs

POPTAs

Implementation and experimental results

Conclusions

Time, clocks and Zones

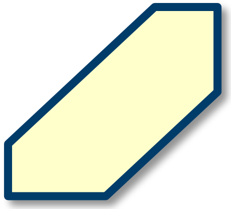
Dense time domain: non-negative reals $\mathbb{R}$

Finite set of **clocks** denoted **X**

- clock $x \in X$ is a variables taking values from time domain $\mathbb{R}$
- clocks increase at the same rate as real time and can be reset to 0
- a clock valuation over the clocks X is a vector $v \in \mathbb{R}^X$
  - for $t \in \mathbb{R}$, $v+t$ is the clock valuation where $(v+t)(x)=v(x)+t$ for all $x \in X$

Zones over clocks **X**, denoted **Zones(X)**, are given by the syntax:

- $\zeta ::= x \le d \mid c \le x \mid x+c \le y+d \mid \neg\zeta \mid \zeta \wedge \zeta$
  - where $x,y \in X$ and $c,d \in \mathbb{N}$
- can be considered as a subclass of polyhedra



A clock valuation **v** satisfies a zone **ζ** if

- ζ resolves to `true` after substituting each clock $x \in X$ with $v(x)$

Probabilistic timed automata (PTAs)

Probabilistic timed automata (PTAs)

- Markov decision processes (MDPs) + real-valued clocks
- or timed automata + discrete probabilistic choice
- model probabilistic, nondeterministic and timed behaviour

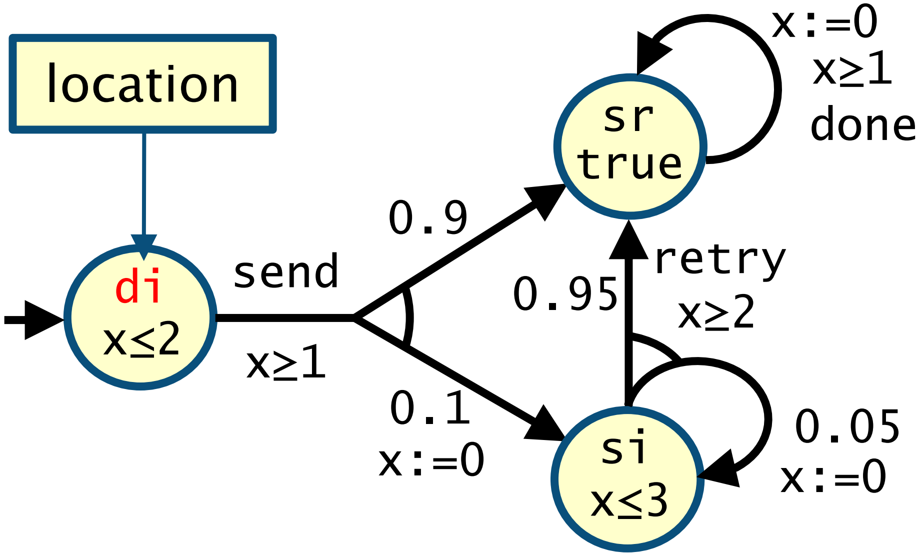
PTA is a tuple $(L, l_0, Act, X, inv, enab, prob, r)$

- L is a finite set of locations with an initial location $l_0 \in L$
- Act is a finite set of actions
- X is a finite set of clocks
- $inv : L \rightarrow Zones(X)$ is an invariant condition
- $enab : (L \times Act) \rightarrow Zones(X)$ is an enabling condition
- $prob : (L \times Act) \rightarrow Dist(2^X \times L)$ is a probabilistic transition function
- $r = (r_L, r_A)$ is a reward structure where $r_L : L \rightarrow \mathbb{R}$ and $r_A : (L \times Act) \rightarrow \mathbb{R}$

PTAs – Example

Models a simple probabilistic communication protocol

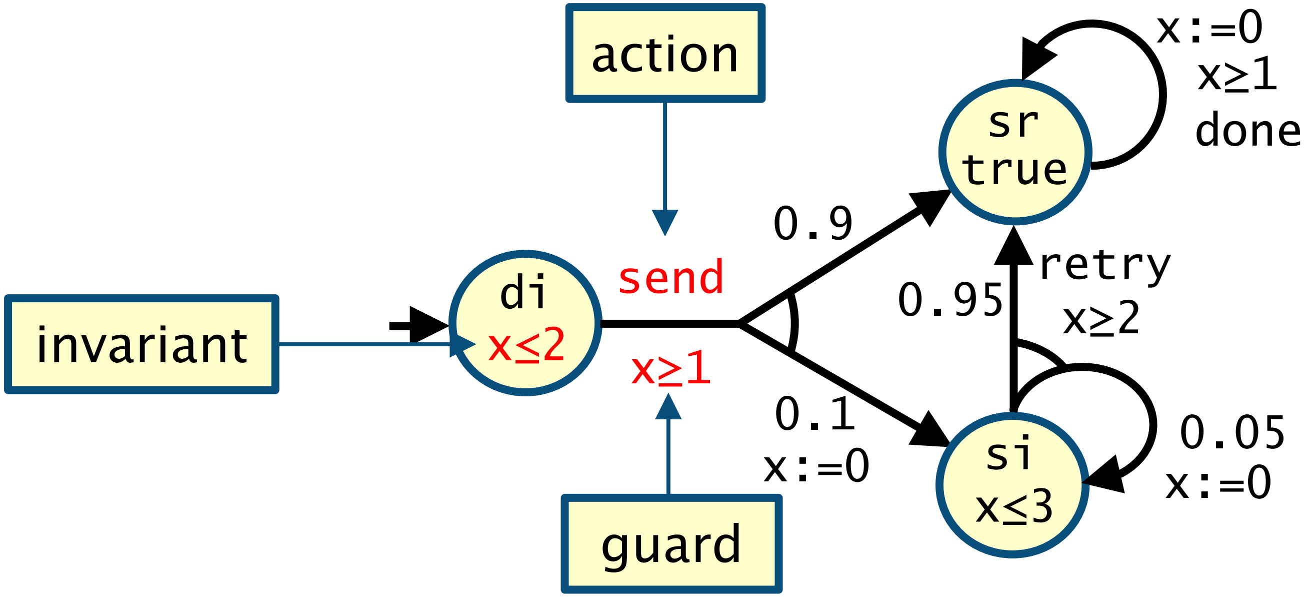
- starts in location di;



PTAs – Example

Models a simple probabilistic communication protocol

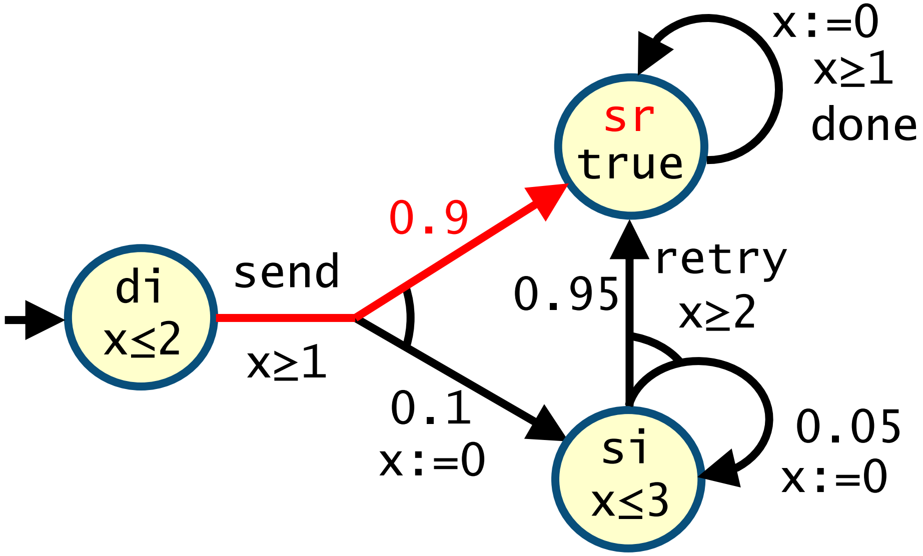
- starts in location di; after between 1 and 2 time units, the protocol attempts to send the data



PTAs – Example

Models a simple probabilistic communication protocol

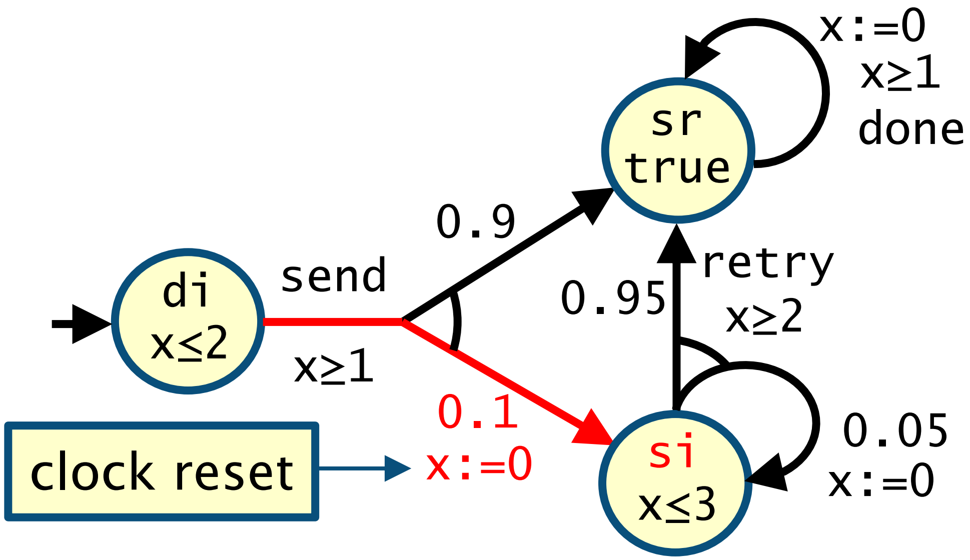
- starts in location di; after between 1 and 2 time units, the protocol attempts to send the data
  - with probability 0.9 data is sent correctly, move to location sr



PTAs – Example

Models a simple probabilistic communication protocol

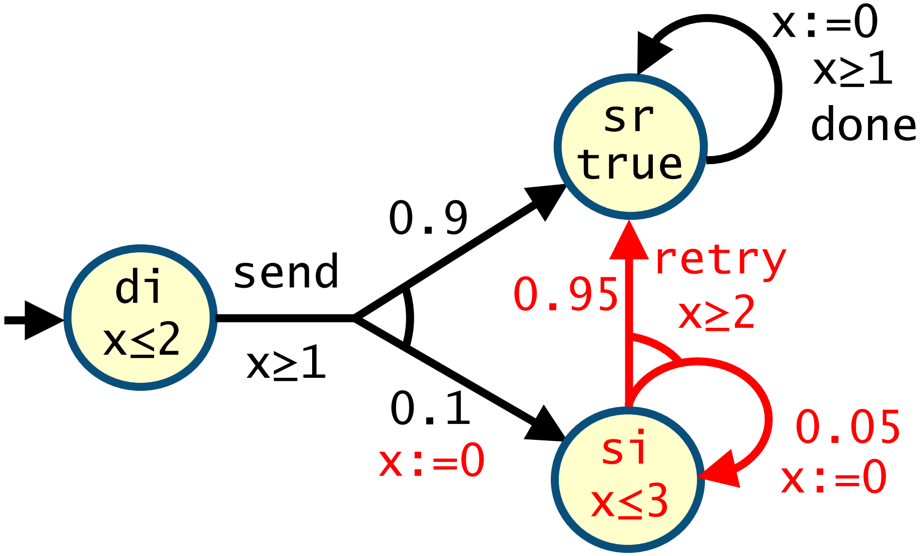
- starts in location di; after between 1 and 2 time units, the protocol attempts to send the data
  - with probability 0.9 data is sent correctly, move to location sr
  - with probability 0.1 data is lost, move to location si



PTAs – Example

Models a simple probabilistic communication protocol

- starts in location di; after between 1 and 2 time units, the protocol attempts to send the data
  - with probability 0.9 data is sent correctly, move to location sr
  - with probability 0.1 data is lost, move to location si
- in location si, after 2 to 3 time units, attempts to retry
  - correctly sent with probability 0.95 and lost with probability 0.05

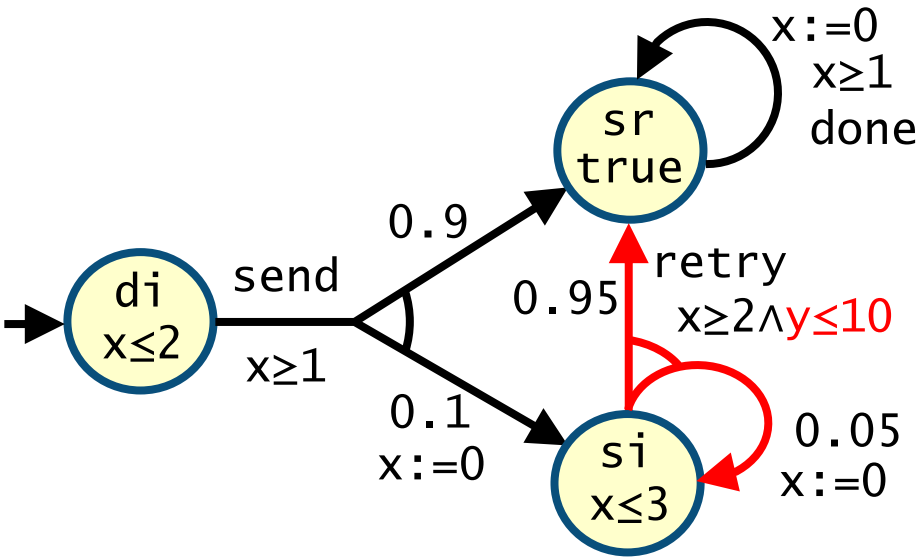


PTAs – Example

Models a simple probabilistic communication protocol

- starts in location di; after between 1 and 2 time units, the protocol attempts to send the data
  - with probability 0.9 data is sent correctly, move to location sr
  - with probability 0.1 data is lost, move to location si
- in location si, after 2 to 3 time units, attempts to retry
  - correctly sent with probability 0.95 and lost with probability 0.05

can now only retry if it is also the case that less than 10 times units have elapsed since initialization



PTAs – Semantics

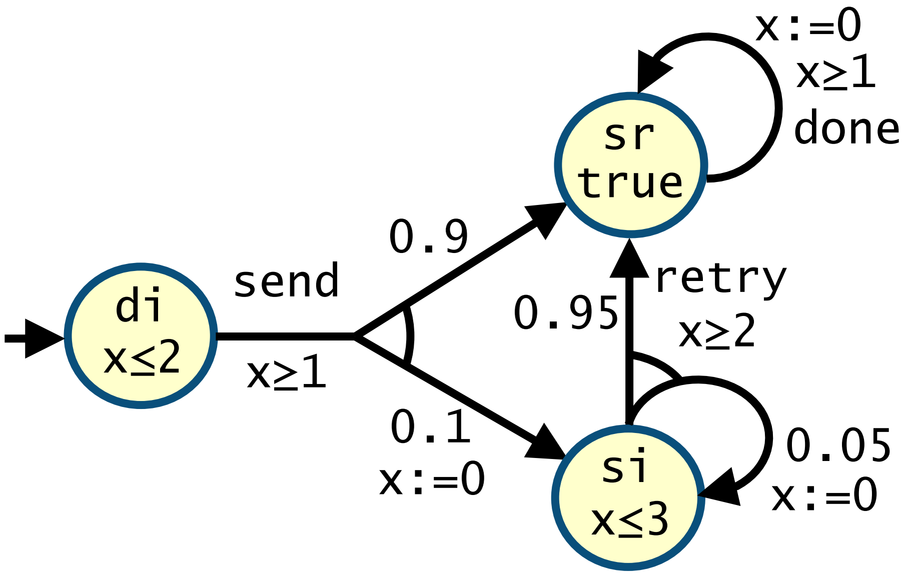
Semantics of a PTA is an infinite state MDP

- states of the MDP are location-clock valuation pairs $(l,v) \in L \times \mathbb{R}^X$ such that invariant $inv(l)$ is satisfied by the clock valuation v
- initial state is the initial location with all clocks initialized to zero
- actions of the MDP are time-PTA action pairs $(t,a) \in \mathbb{R} \times Act$
  - corresponds to letting t time units pass and then performing action a

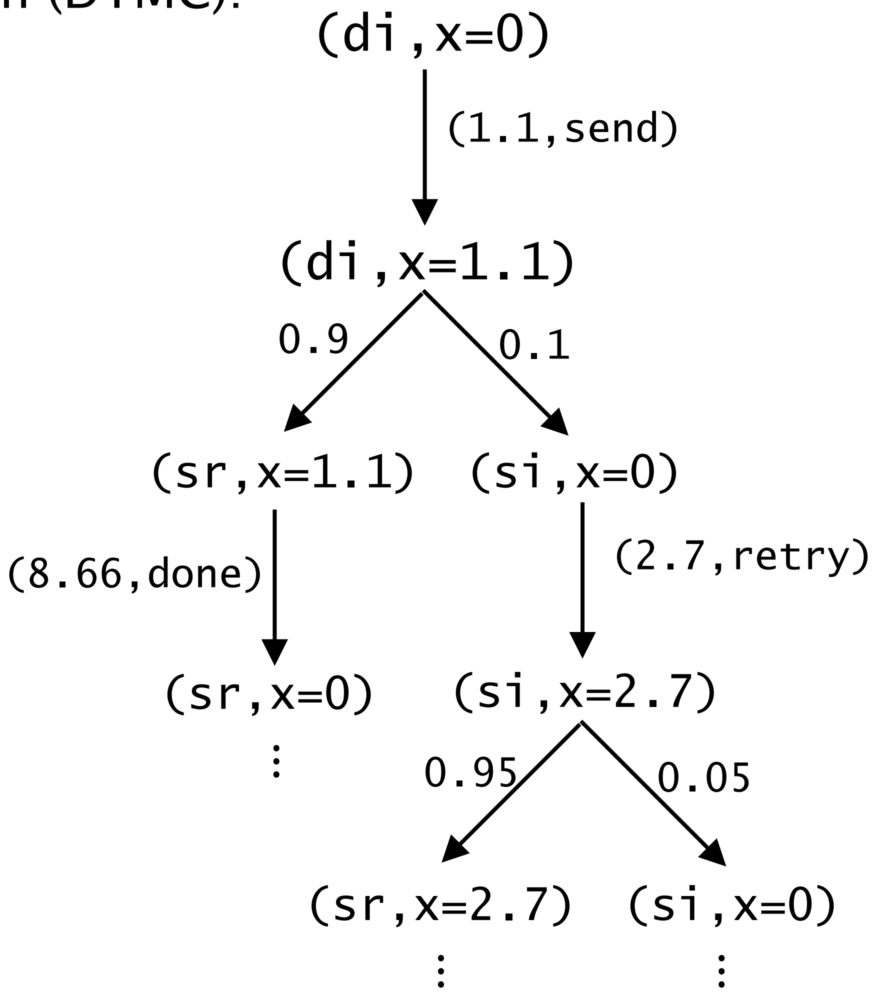
In state (l,v) a **nondeterministic choice** over the time t that elapses and action a performed under the requirement that from (l,v):

- the invariant $inv(l)$ is continuously satisfied during time t
  - i.e. $v+t'$ satisfies $inv(l)$ for all $0 \leq t' \leq t$
- a is enabled after t time units have elapsed
  - i.e. $enab(l,a)$ is satisfied by $v+t$
- if a is chosen, then probability of moving to location l' and resetting the set of clocks Y equals $prob(l,a)(l',Y)$

PTAs – Semantics



Example execution (DTMC):



PTAs – Semantics

PTAs have two kinds of rewards:

- location rewards accumulated at rate $r_L(l)$ in location l as time passes
- action rewards accumulated instantaneous with value $r_A(l,a)$ when taking action a in l

PTAs equipped with such reward structures are a probabilistic extension of linearly-priced timed automata

- also called weighted timed automata

PTAs – Digital clocks

Clocks can only take integer (digital) values

- i.e. time domain is a subset of $\mathbb{N}$ as opposed to $\mathbb{R}$

Digital clocks semantics yields a finite-state MDP which preserves optimal probabilistic and expected reachability values

- under the requirement that zones are closed (no strict inequalities) and diagonal-free (no comparison of clock values)
- clocks bounded by $k_{max}+1$, where k_{max} is the maximum constant in PTA

Automated analysis exists (e.g. using PRISM)

- translation from PTA to a finite state MDP can also be done manually
- many case studies despite restrictions
- can lead to large MDPs (partially alleviated by symbolic model checking)

PTAs – Alternative approaches for analysis

The region graph useful for proving decidability, but exponential size

Zone-based approaches based on “building” a finite state MDP by traversing the PTA where the states are location-zone pairs

- forwards reachability
  - can only compute upper bounds on maximum reachability probabilities
  - extended to computation of optimal reachability probabilities using game-based abstraction refinement but introduces non-convex zones
    - more complex in practice to perform operations on such zones
- backwards reachability
  - can compute optimal reachability probabilities and rewards
  - requires non-convex zones for minimum probabilities and rewards
  - rational constants in zones also required for rewards

Outline

MDPs

POMDPs

PTAs

POPTAs

Implementation and experimental results

Conclusions

Partially observable PTAs (POPTAs)

POPTA is a tuple $(L,l_0,Act,X,inv,enab,prob,r,O_L,obs_L)$

- $(L,l_0,Act,X,inv,enab,prob,r)$ is a PTA
- O_L is a finite set of observations
- $obs_L:L \rightarrow O_L$ is a labelling of locations with observations

Require that for any locations l and l' :

- if $obs_L(l)=obs_L(l')$, then $inv(l)=inv(l')$ and $enab(l,a)=enab(l',a)$ for all actions a
- i.e. for any observationally equivalent locations the invariant and enabling conditions are the same
- similar to the requirement on POMDPs
- otherwise the strategy could differentiate the locations, and therefore they would not be observationally equivalent

Partially observable PTAs (POPTAs)

POPTA is a tuple **$(L, l_0, Act, X, inv, enab, prob, r, O_L, obs_L)$**

- $(L, l_0, Act, X, inv, enab, prob, r)$ is a PTA
- O_L is a finite set of observations
- $obs_L : L \rightarrow O_L$ is a labelling of locations with observations

The semantics is an (uncountable) infinite state POMDP

- states as for PTAs: location-clock valuation pairs (l, v)
- probabilistic transition and reward functions as for PTAs
- observations of the POMDP given by $obs((l, v)) = (obs_L(l), v)$

Note: all clocks are **visible**

- things gets even more complex if (some) clocks are hidden – underlying semantics will need to be a partially observable (two player) game

POPTAs and digital clocks

Theorem. Under the following restrictions the digital clocks semantics preserves probabilistic and expected reachability values of POPTAs

1. zones must be closed and diagonal free
   - as for PTAs and TAs is required when using digital clocks semantics
2. resets can only be applied to clocks that are non-zero
   - clock resets can be used to differentiate between locations
     - when there are probabilistic edges going to observationally equivalent locations where only one of the edges resets a (visible) clock
   - without this with real-valued clocks, a strategy can do this at no "cost"
     - can choose an arbitrarily small amount of time to elapse before taking the corresponding transition
   - while with digital clocks must let at least 1 time unit pass

Both restrictions can be easily checked syntactically

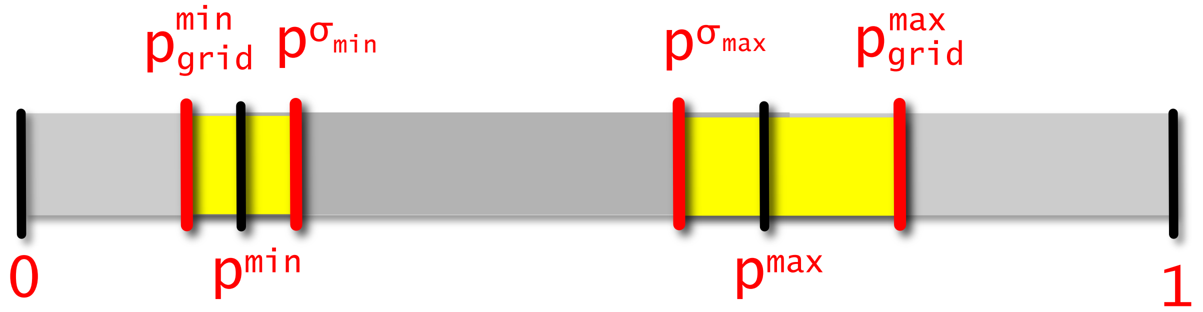
POPTAs and digital clocks

Semantics of a POPTA is an infinite state POMDP

Using digital clocks semantics we can construct an ‘equivalent’ finite state POMDP

Can then analyse the finite POMDP using grid-based techniques

- again produce lower and upper bounds on the property of interest and ‘optimal’ strategy σ_{opt}
- and if the results are not precise enough, we can refine and repeat



Outline

MDPs

POMDPs

PTAs

POPTAs

Implementation and experimental results

Conclusions

Implementation – For POMPs and POPTAs

Recently integrated into the main branch of PRISM

- extends existing modelling language for MDPs and PTAs
- allows model variables to be specified as either observable or hidden
  - or just the truth values only predicates over variables to be visible
- computes a pair of bounds for a given property
- and synthesizes a corresponding strategy

Case studies available

- through the PRISM website

Google
Summer of Code

Implementation – For POMPs and POPTAs

Uses a fixed grid from the literature [Lovejoy 1991]

- requires a `grid resolution constant` $M \in \mathbb{N}$, grid is then given by
- $G = \{ (1/M) \cdot v \mid v \in \mathbb{N}^S \wedge \Sigma_i v_i = M \} \subseteq Dist(S)$

A benefit is that interpolation is very efficient

- i.e. given arbitrary belief (distribution) b finding constants $\gamma_1, \dots, \gamma_N$ such that $b = \gamma_1 \cdot g_1 + \cdots + \gamma_N \cdot g_N$
- this using a process called triangulation

A downside is that the grid size is exponential in **M**

- efficiency might be improved with more complex grids that vary using value iteration
    - e.g. grid updated based on the beliefs found during each iteration

Case studies

Wireless downlink scheduling of traffic to number of users/channels

- uses hard deadlines (packets not sent by their deadline are dropped)
- packets have priorities (`low`, `medium` or `high`)
- status of channels is not available (unobservable)
    - since probing channels requires non-negligible network resources
- generate optimal scheduling of packets for time bounded properties
    - minimum number of dropped packets
    - maximize expected cumulative reward
        - rewards accumulated when packets sent and based on priorities

Analysis demonstrates that **`idling`** is sometimes the optimal choice

- `idling`: not sending a packet when there are packets to send
- confirms results obtained from handwritten proofs
- `idling` allows the scheduler to learn more about the status of channels which have `high` priority packets scheduled

Case studies

NRL (Naval Research Laboratory) Pump

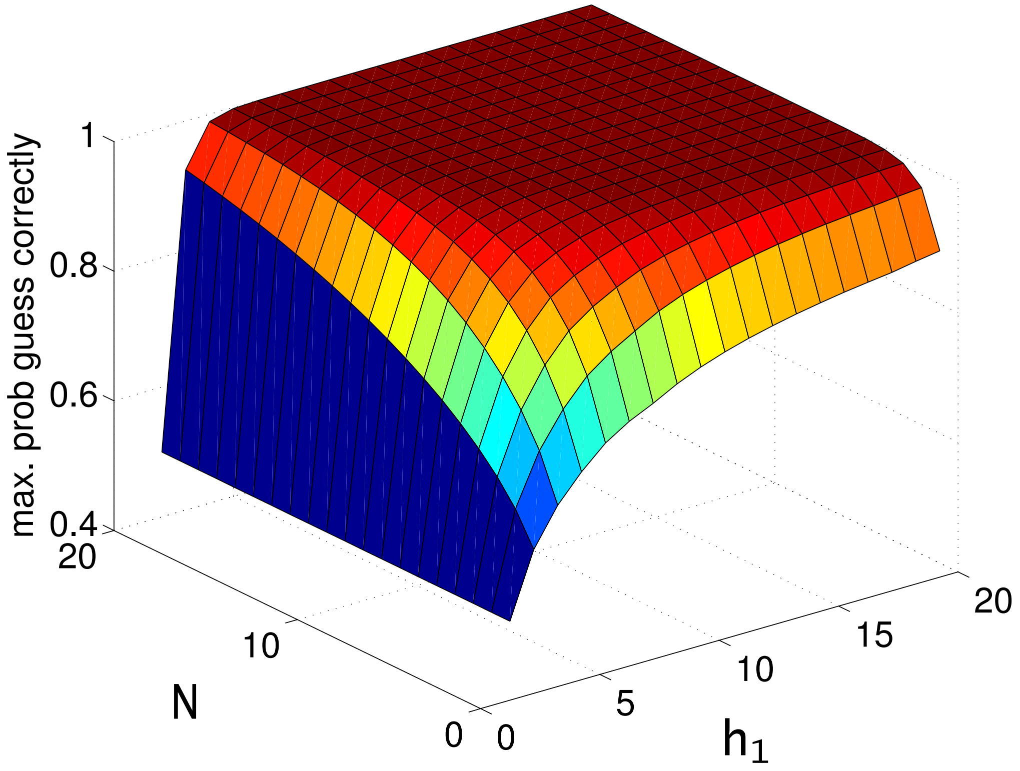
- aims to prevent a covert channel leaking information from 'high' to 'low' through the timing of messages and acknowledgements
- buffers communication adding probabilistic delays to acks from 'high' to minimize information leakage while maintaining network performance

Modelled the pump as a POPTA using a hidden variable for a secret value $z \in \{0,1\}$ which '**high**' tries to covertly communicate to '**low**'

- 'high' adds a delay of either h_0 or h_1, depending on the value of z, whenever sending an acknowledgement to 'low'

Maximum probability that '**low**' can correctly guess the secret value after **N** messages sent (and acknowledgements received) by '**low**'

Case studies – NRL Pump

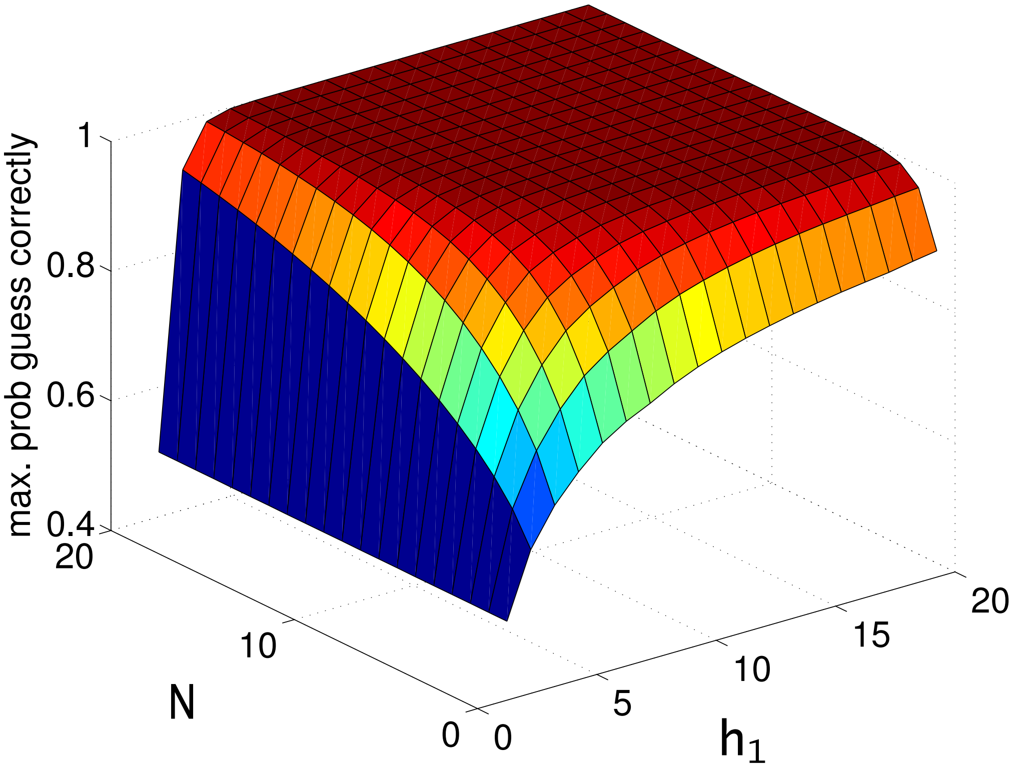


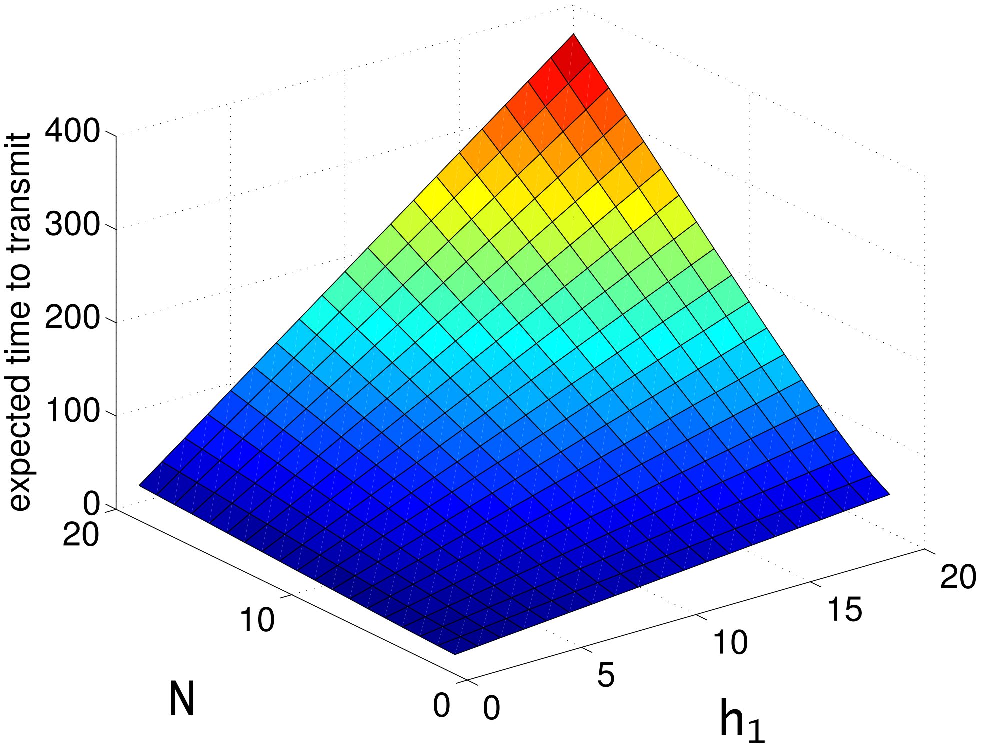
N – number of messages 'low' sends (and acks 'high' returns)

h_i – delay added by high if secret value equals i (h_0=2 in all plots)

increasing the difference between h_0 and h_1 or N improve the chances of 'low' correctly guessing the secret

Case studies – NRL Pump





N – number of messages 'low' sends (and acks 'high' returns)

h_i – delay added by high if secret value equals i (h_0=2 in all plots)

increasing the difference between h_0 and h_1 or N improve the chances of 'low' correctly guessing the secret, at the cost of a decrease in network performance

Case studies

Task scheduling

- processors have different speeds and energy consumption
- scheduler cannot observe if a process is `sleeping` or `idling`
- synthesize optimal schedulers
  - minimising expected execution time or energy usage

Non-repudiation protocol

- designed to allow an `originator` to send information to a `recipient`
- guarantees non-repudiation
  - neither party can deny having participated in the information transfer
- recipient cannot observe the number of messages the originator will send
- maximum probability that `recipient` gains an unfair advantage
  - gains information from `originator` while able to deny participation

Analysis of the results

Analysed POPTAs where integer semantics yields POMDPs of up to 60,000 states

Verification/synthesis for POMDPs and POPTAs usually taking just a few seconds and, at worst, 20 minutes

In general, the lower and upper bounds generated are very close (or even equal, in which case we obtain precise results)

- although in some case bounds do not converge given memory limitations
- grid approximation can have millions states

Outline

MDPs

POMDPs

PTAs

POPTAs

Implementation and experimental results

Conclusions

Conclusions

Model non-determinism, real-time, probability & partial observability

- all are required for many real-world case studies

Future work and improvements include

- only presented a simple solution method for POMDPs
- many more advanced techniques in AI and planning
  - e.g. grid points are not not fixed
- symbolic techniques (currently only have an explicit implementation)
- use zone based approach (most successful approach for PTAs and TAs)
- extend POPTAs with unobservable clocks
  - need a second player (environment) to make choices based on the values of the hidden clocks some work on this for (non-probabilistic) TAs
  - however partially observable stochastic games are even harder to solve

Questions?